

# LilyPond

---

El tipografiador de música

## Referencia de la notación

### El equipo de desarrolladores de LilyPond

Este manual ofrece una referencia para toda la notación musical que se puede producir con LilyPond versión 2.23.3. Da por sentado que el lector está familiarizado con el material que hay en Sección “Manual de aprendizaje” en *Manual de Aprendizaje*.

Para mayor información sobre la forma en que este manual se relaciona con el resto de la documentación, o para leer este manual en otros formatos, consulte Sección “Manuales” en *Información general*.

Si le falta algún manual, encontrará toda la documentación en <http://lilypond.org/>.

Copyright © 1998–2021 por los autores. *La traducción de la siguiente nota de copyright se ofrece como cortesía para las personas de habla no inglesa, pero únicamente la nota en inglés tiene validez legal.*

*The translation of the following copyright notice is provided for courtesy to non-English speakers, but only the notice in English legally counts.*

Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU, versión 1.1 o cualquier versión posterior publicada por la Free Software Foundation; sin ninguna de las secciones invariantes. Se incluye una copia de esta licencia dentro de la sección titulada “Licencia de Documentación Libre de GNU”.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Para la versión de LilyPond 2.23.3

---

# Índice General

<b>1</b>	<b>Notación musical</b>	<b>1</b>
1.1	Alturas	1
1.1.1	Escritura de notas	1
	Escritura de octava absoluta	1
	Escritura de octava relativa	2
	Alteraciones accidentales	6
	Nombres de las notas en otros idiomas	8
1.1.2	Modificación de varias notas a la vez	10
	Comprobación de octava	10
	Transposición	11
	Inversión	14
	Retrogradación	15
	Transformaciones modales	16
1.1.3	Imprimir las alturas	18
	Clave	18
	Armadura de la tonalidad	23
	Corchetes de octava	25
	Transposición de los instrumentos	28
	Alteraciones accidentales automáticas	30
	Tesitura	38
1.1.4	Cabeza de las notas	40
	Cabezas de nota especiales	41
	Cabezas de notas de Notación Fácil	42
	Cabezas de notas con formas diversas	44
	Improvisación	47
1.2	Duraciones	48
1.2.1	Escritura de las duraciones (valores rítmicos)	49
	Duración de las notas	49
	Grupos especiales	51
	Escarlar las duraciones	56
	Ligaduras de unión	58
1.2.2	Escritura de silencios	62
	Silencios	62
	Silencios invisibles	64
	Silencios de compás completo	65
1.2.3	Impresión de las duraciones	70
	Indicación de compás	70
	Indicaciones metronómicas	74
	Anacrusas	78
	Música sin compasear	79
	Notación polimétrica	81
	División automática de las notas	84
	Mostrar los ritmos de la melodía	86
1.2.4	Barras	88
	Barras automáticas	88
	Establecer el comportamiento de las barras automáticas	91
	Barras manuales	100
	Barras progresivas	103

1.2.5	Compases	104
	Barras de compás	104
	Numeración de compases	112
	Comprobación de compás y de número de compás	117
	Llamadas de ensayo	118
1.2.6	Asuntos rítmicos especiales	120
	Notas de adorno	120
	Alinear con una cadenza	125
	Gestión del tiempo	126
1.3	Expresiones	127
1.3.1	Expresiones adosadas a las notas	128
	Articulaciones y ornamentos	128
	Matices dinámicos	131
	Indicaciones dinámicas nuevas	137
1.3.2	Expresiones como curvas	139
	Ligaduras de expresión	140
	Ligaduras de fraseo	143
	Marcas de respiración	144
	Caídas y elevaciones	146
1.3.3	Expresiones como líneas	147
	Glissando	147
	Arpeggio	152
	Trinos	155
1.4	Repeticiones	158
1.4.1	Repeticiones largas	159
	Repeticiones normales	159
	Marcas de repetición manual	166
	Repeticiones explícitas	168
1.4.2	Repeticiones cortas	170
	Repeticiones de compás o parte de ellos	170
	Repeticiones de trémolo	173
1.5	Notas simultáneas	175
1.5.1	Una voz única	176
	Notas en acorde	176
	Repetición de acordes	178
	Expresiones simultáneas	179
	Racimos (clusters)	181
1.5.2	Varias voces	181
	Polifonía en un solo pentagrama	181
	Estilos de voz	185
	Resolución de las colisiones	186
	Combinación de silencios	191
	Combinación automática de las partes	192
	Escribir música en paralelo	197
1.6	Notación de los pentagramas	200
1.6.1	Impresión de los pentagramas	200
	Crear instancias de pentagramas nuevos	201
	Agrupar pentagramas	202
	Grupos de pentagramas anidados	205
	Separación de sistemas	207
1.6.2	Modificación de pentagramas sueltos	208
	El símbolo del pentagrama	208
	Pentagramas de Ossia	212
	Ocultar pentagramas	215

1.6.3	Escritura de las particellas .....	220
	Nombres de instrumentos .....	220
	Citar otras voces .....	224
	Formateo de las notas guía .....	227
	Compresión de los compases vacíos .....	233
1.7	Anotaciones editoriales .....	235
1.7.1	Dentro del pentagrama .....	235
	Seleccionar el tamaño de la tipografía para la notación .....	235
	Indicaciones de digitación .....	240
	Notas ocultas .....	242
	Colorear objetos .....	243
	Paréntesis .....	245
	Plicas .....	245
1.7.2	Fuera del pentagrama .....	247
	Nombre de las notas .....	247
	Globos de ayuda .....	248
	Líneas de rejilla .....	249
	Corchetes de análisis .....	251
1.8	Texto .....	254
1.8.1	Escritura del texto .....	254
	Panorámica de los objetos de texto .....	254
	Guiones de texto .....	257
	Extensiones de texto .....	258
	Indicaciones de texto .....	260
	Texto separado .....	262
1.8.2	Formatear el texto .....	264
	Introducción al marcado de texto .....	264
	Seleccionar la tipografía y su tamaño .....	266
	Alineación de texto .....	269
	Notación gráfica dentro de elementos de marcado .....	274
	Notación musical dentro de elementos de marcado .....	276
	Elementos de marcado de varias páginas .....	279
1.8.3	Tipografías .....	279
	Explicación de las fuentes tipográficas .....	280
	Fuentes de un solo elemento .....	283
	Fuentes tipográficas del documento completo .....	284
<b>2</b>	<b>Notación especializada .....</b>	<b>286</b>
2.1	Música vocal .....	286
2.1.1	Notación común para música vocal .....	286
	Referencias para música vocal .....	286
	Introducir la letra .....	287
	Alineación de la letra a una melodía .....	288
	Duración automática de las sílabas .....	290
	Duración manual de las sílabas .....	293
	Varias sílabas sobre una nota .....	294
	Varias notas sobre una sílaba .....	295
	Líneas de extensión y guiones .....	298
	Cambios de vocal graduales .....	299
2.1.2	Técnicas específicas para la letra .....	299
	Trabajar con letra y variables .....	299
	Posicionamiento vertical de la letra .....	301
	Colocación horizontal de las sílabas .....	306
	Letra y repeticiones .....	307

Letras en divisi .....	315
Polifonía con letras compartidas .....	317
2.1.3 Versos .....	319
Añadir números de verso .....	319
Añadir expresiones dinámicas a los versos .....	320
Añadir el nombre de los cantantes a los versos .....	320
Versos con ritmos distintos .....	321
Imprimir los versos al final .....	324
Imprimir los versos al final en varias columnas .....	325
2.1.4 Canciones .....	326
Referencias para canciones .....	326
Hojas guía de acordes .....	327
2.1.5 Música coral .....	328
Referencias para música coral .....	328
Disposiciones de la partitura para música coral .....	329
2.1.6 Ópera y musicales .....	331
Referencias para ópera y musicales .....	331
Nombres de los personajes .....	332
Guías musicales .....	334
Música hablada .....	338
Diálogos encima de la música .....	338
2.1.7 Cánticos salmos e himnos .....	339
Referencias para cánticos y salmos .....	339
Preparar un cántico .....	340
Puntuación de un salmo .....	346
Compases parciales en melodías de himno .....	349
2.1.8 Música vocal antigua .....	352
2.2 Teclados y otros instrumentos de varios pentagramas .....	352
2.2.1 Notación común para instrumentos de teclado .....	353
Referencias para teclados .....	353
Cambiar de pentagrama manualmente .....	354
Cambiar de pentagrama automáticamente .....	356
Líneas de cambio de pentagrama .....	358
2.2.2 Piano .....	361
Pedales de piano .....	361
2.2.3 Acordeón .....	362
Símbolos de Discanto .....	362
2.2.4 Arpa .....	363
Referencias para notación de arpa .....	364
Pedales de arpa .....	364
2.3 Instrumentos de cuerda sin trastes .....	365
2.3.1 Notación común para cuerdas sin trastes .....	365
Referencias para cuerdas sin trastes .....	365
Indicaciones de arco .....	366
Armónicos .....	366
Snap (Bartók) pizzicato .....	367
2.4 Instrumentos de cuerda con trastes .....	368
2.4.1 Notación común para cuerdas con trastes .....	369
Referencias para cuerdas con trastes .....	369
Indicación de los números de cuerda .....	369
Tablaturas predeterminadas .....	371
Tablaturas personalizadas .....	386
Marcas de diagramas de trastes .....	390
Diagramas predefinidos de trastes .....	400

Diagramas de traste automáticos.....	411
Digitaciones de la mano derecha .....	414
2.4.2 Guitarra.....	415
Indicar la posición y la cejilla.....	415
Indicar armónicos y notas tapadas.....	416
Indicación de acordes de potencia o de quinta vacía.....	417
2.4.3 Banjo.....	418
Tablaturas de banjo.....	418
2.4.4 Laúd.....	419
Tablaturas de laúd.....	419
2.5 Percusión.....	420
2.5.1 Notación común para percusión.....	420
Referencias para percusión.....	420
Notación básica de percusión .....	420
Redobles .....	421
Percusión afinada.....	422
Pautas de percusión .....	422
Pautas de percusión personalizadas .....	425
Notas fantasma.....	427
2.6 Instrumentos de viento .....	428
2.6.1 Notación común para instrumentos de viento.....	428
Referencias para instrumentos de viento.....	428
Digitaciones .....	429
2.6.2 Gaita.....	431
Definiciones para la gaita .....	431
Ejemplo de música de gaita.....	432
2.6.3 Instrumentos de viento madera.....	433
2.6.3.1 Diagramas de posiciones para viento madera .....	433
2.7 Notación de acordes .....	442
2.7.1 Modo de acordes.....	442
Panorámica del modo de acordes.....	442
Acordes más usuales.....	443
Acordes extendidos y alterados .....	444
Inversiones de acorde y realizaciones especiales .....	447
2.7.2 Imprimir los acordes .....	448
Impresión de los nombres de acorde.....	448
Nombres de acorde personalizados .....	450
2.7.3 Bajo cifrado .....	456
Introducción al bajo cifrado.....	456
Introducir el bajo cifrado .....	457
Imprimir el bajo cifrado .....	460
2.8 Música contemporánea.....	462
2.8.1 Altura y armonía en la música contemporánea.....	462
Referencias para alturas y armonía en música contemporánea .....	462
Notación microtonal.....	462
Armonía y armaduras de tonalidad contemporáneas.....	463
2.8.2 Enfoques contemporáneos del ritmo.....	463
Referencias para enfoque contemporáneos del ritmo .....	463
Grupos de valoración especial en música contemporánea.....	463
Compases contemporáneos.....	463
Notación polimétrica extendida.....	463
Barrado en música contemporánea.....	463
Líneas divisorias en música contemporánea.....	463
2.8.3 Notación gráfica .....	463

2.8.4	Técnicas de partitura contemporáneas.....	463
2.8.5	Nuevas técnicas instrumentales.....	463
2.8.6	Lecturas adicionales y partituras de interés.....	463
	Libros y artículos sobre notación musical contemporánea.....	463
	Partituras y ejemplos musicales.....	463
2.9	Notación antigua.....	463
2.9.1	Panorámica de los estilos contemplados.....	465
2.9.2	Notación antigua - funcionalidades comunes.....	465
	Contextos predefinidos.....	465
	Ligaduras.....	466
	Custos.....	466
2.9.3	Tipografiar música mensural.....	468
	Contextos de la música mensural.....	468
	Claves de la música mensural.....	469
	Indicaciones de compás de la música mensural.....	470
	Cabezas de nota de la música mensural.....	471
	Corchetes de la música mensural.....	471
	Silencios de la música mensural.....	472
	Alteraciones y armaduras de la música mensural.....	473
	Alteraciones de anotación ( <i>musica ficta</i> ).....	473
	Ligaduras mensurales blancas.....	474
2.9.4	Tipografiado del canto gregoriano.....	475
	Contextos del canto gregoriano.....	476
	Claves de canto gregoriano.....	476
	Alteraciones y armaduras de canto gregoriano.....	477
	Divisiones.....	478
	Articulaciones del canto gregoriano.....	478
	Puntos de aumentación ( <i>morae</i> ).....	479
	Ligaduras de neumas cuadrados gregorianos.....	480
2.9.5	Tipografiado del canto kievano en notación cuadrada.....	486
	Contextos del canto kievano.....	486
	Claves del canto kievano.....	487
	Notas del canto kievano.....	488
	Alteraciones accidentales del canto kievano.....	488
	Líneas divisorias del canto kievano.....	489
	Melismas del canto kievano.....	489
2.9.6	Trabajar con música antigua - escenarios y soluciones.....	490
	Incipits.....	490
	Disposición tipo «Mensurstriche».....	491
	Transcripción de canto gregoriano.....	492
	Antigua y moderna desde la misma fuente.....	495
	Marcado editorial.....	496
2.10	Músicas del mundo.....	497
2.10.1	Notación común para músicas no occidentales.....	497
	Extensión de la notación y los sistemas de afinación.....	497
2.10.2	Música árabe.....	497
	Referencias para música árabe.....	497
	Nombres de nota árabes.....	498
	Armaduras de tonalidad árabes.....	499
	Indicaciones de compás árabes.....	501
	Ejemplo de música árabe.....	502
	Lecturas adicionales sobre música árabe.....	503
2.10.3	Música clásica de Turquía.....	503
	Referencias para música clásica de Turquía.....	503

Nombres de las notas en turco .....	503
Armaduras de tonalidad de la música turca .....	504
Lecturas adicionales sobre música turca .....	505
<b>3 Entrada y salida generales .....</b>	<b>506</b>
3.1 Estructura del código de entrada .....	506
3.1.1 Estructura de una partitura .....	506
3.1.2 Varias partituras en un libro .....	507
3.1.3 Varios archivos de salida a partir de uno de entrada .....	508
3.1.4 Nombres de los archivos de salida .....	509
3.1.5 Estructura del archivo .....	510
3.2 Títulos y encabezamientos .....	512
3.2.1 Crear títulos, encabezamientos y pies de página .....	512
Explicación de los títulos .....	512
Presentación predeterminada de los títulos de partes de libro y partitura .....	515
Disposición predeterminada de las cabeceras y pies de página .....	519
3.2.2 Títulos, encabezamientos y pies de página personalizados .....	520
Formateo personalizado del texto de los bloques de título .....	520
Personalización de los títulos .....	520
Disposición personalizada de cabeceras y pies de página .....	523
3.2.3 Crear metadatos en los archivos de salida .....	525
3.2.4 Crear notas al pie .....	526
Notas al pie dentro de expresiones musicales .....	526
Notas al pie en texto independiente .....	531
3.2.5 Referencia a números de página .....	534
3.2.6 Índice general .....	535
3.3 Trabajar sobre los archivos de entrada .....	538
3.3.1 Inclusión de archivos de LilyPond .....	538
3.3.2 Distintas ediciones a partir de una misma fuente .....	540
Uso de las variables .....	540
Uso de etiquetas .....	541
Uso de ajustes globales .....	545
3.3.3 Caracteres especiales .....	546
Codificación del texto .....	546
Unicode .....	546
Alias de ASCII .....	547
3.4 Controlar la salida .....	548
3.4.1 Extracción de fragmentos de música .....	548
3.4.2 Saltar la música corregida .....	549
3.4.3 Formatos de salida alternativos .....	550
Salida de SVG .....	550
3.4.4 Sustituir la tipografía de la notación .....	550
3.5 Creación de salida MIDI .....	552
3.5.1 Notación contemplada por el MIDI .....	553
3.5.2 Notación no contemplada en el MIDI .....	553
3.5.3 El bloque MIDI .....	554
3.5.4 Control de las dinámicas del MIDI .....	555
Indicaciones de matiz dinámico en el MIDI .....	555
Establecimiento del volumen MIDI .....	556
Establecimiento de las propiedades del bloque MIDI .....	558
3.5.5 Uso de los instrumentos MIDI .....	559
3.5.6 Uso de las repeticiones con el MIDI .....	560
3.5.7 Asignación de canales MIDI .....	561
3.5.8 Propiedades de contexto para efectos MIDI .....	563



3.5.9	Enriquecimiento de la salida MIDI .....	564
3.5.10	El script Articulate .....	564
3.6	Extraer información musical .....	565
3.6.1	Displaying LilyPond notation .....	565
3.6.2	Impresión de las expresiones musicales de Scheme .....	565
3.6.3	Guardar los eventos musicales en un archivo .....	565
<b>4</b>	<b>Problemas de espaciado .....</b>	<b>567</b>
4.1	Disposición de la página .....	567
4.1.1	El bloque <code>\paper</code> .....	567
4.1.2	Tamaño del papel y escalado automático .....	568
	Fijar el tamaño del papel .....	568
	Escalado automático al tamaño del papel .....	569
4.1.3	Variables de espaciado de <code>\paper</code> verticales fijas .....	570
4.1.4	Variables de espaciado de <code>\paper</code> verticales flexibles .....	571
	Estructura de las listas-A de espaciado vertical flexible .....	571
	Lista de variables de espaciado de <code>\paper</code> verticales flexibles .....	572
4.1.5	Variables de espaciado de <code>\paper</code> horizontales .....	573
	Variables de <code>\paper</code> para la anchura y los márgenes .....	573
	Variables de <code>\paper</code> para el modo de doble cara .....	574
	Variables de <code>\paper</code> para desplazamientos y sangrados .....	575
4.1.6	Otras variables de <code>\paper</code> .....	575
	Variables de <code>\paper</code> para los saltos de línea .....	575
	Variables de <code>\paper</code> para los saltos de página .....	576
	Variables de <code>\paper</code> para la numeración de las páginas .....	577
	Variables de <code>\paper</code> diversas .....	578
4.2	Disposición de la partitura .....	579
4.2.1	El bloque <code>\layout</code> .....	579
4.2.2	Establecer el tamaño del pentagrama .....	581
4.3	Salto .....	582
4.3.1	Salto de línea .....	582
4.3.2	Salto de página .....	586
	Salto de página manuales .....	586
	Salto de página óptimos .....	587
	Salto de página mínimos .....	587
	Salto de página del tipo página única .....	588
	Salto de página del tipo una-línea .....	588
	Salto de página del tipo una-línea-altura-automática .....	588
	Paso de página óptimo .....	588
4.4	Espaciado vertical .....	589
4.4.1	Espaciado vertical flexible dentro de los sistemas .....	590
	Propiedades de espaciado dentro de los sistemas .....	590
	Espaciado de pautas no agrupadas .....	593
	Espaciado de pautas agrupadas .....	594
	Espaciado de las líneas que no son pautas .....	595
4.4.2	Posicionamiento explícito de los pentagramas y los sistemas .....	597
4.4.3	Evitar las colisiones verticales .....	605
4.5	Espaciado horizontal .....	606
4.5.1	Panorámica del espaciado horizontal .....	606
4.5.2	Sección de espaciado nueva .....	608
4.5.3	Cambiar el espaciado horizontal .....	609
	Estiramiento uniforme de los grupos especiales .....	610
	Espaciado estricto de las notas .....	611
4.5.4	Longitud de la línea .....	611

4.5.5	Notación proporcional .....	611
4.6	Encajar la música en menos páginas .....	618
4.6.1	Mostrar el espaciado .....	618
4.6.2	Cambiar el espaciado .....	619
<b>5</b>	<b>Cambiar los valores por omisión .....</b>	<b>622</b>
5.1	Contextos de interpretación .....	622
5.1.1	Explicación de los contextos .....	622
	Definiciones de salida - estructura de los contextos .....	622
	Score. El contexto maestro .....	623
	Contextos del nivel superior: contenedores de pentagramas .....	623
	Contextos de nivel intermedio. Pentagramas .....	623
	Contextos del nivel más bajo. Voces .....	624
5.1.2	Crear y referenciar contextos .....	624
5.1.3	Mantener vivos los contextos .....	628
5.1.4	Modificar los complementos (plug-ins) de contexto .....	631
5.1.5	Cambiar los valores por omisión de los contextos .....	633
	Cambiar todos los contextos del mismo tipo .....	633
	Cambiar solamente un contexto determinado .....	635
	Orden de precedencia .....	637
5.1.6	Definir contextos nuevos .....	638
5.1.7	Orden de disposición de los contextos .....	640
5.2	Explicación del Manual de referencia de funcionamiento interno .....	642
5.2.1	Navegar por la referencia del programa .....	642
5.2.2	Interfaces de la presentación .....	643
5.2.3	Determinar la propiedad del grob .....	644
5.2.4	Convenciones de nombres .....	645
5.3	Modificar las propiedades .....	646
5.3.1	Panorámica de la modificación de las propiedades .....	646
5.3.2	La instrucción <code>\set</code> .....	646
5.3.3	La instrucción <code>\override</code> .....	648
5.3.4	La instrucción <code>\tweak</code> .....	650
5.3.5	<code>\set</code> frente a <code>\override</code> .....	652
5.3.6	La instrucción <code>\offset</code> .....	653
5.3.7	Modificación de las listas-A .....	658
5.4	Conceptos y propiedades útiles .....	660
5.4.1	Modos de entrada .....	660
5.4.2	Dirección y posición .....	661
	Indicadores de dirección de las articulaciones .....	662
	La propiedad de dirección .....	662
5.4.3	Distancias y medidas .....	663
5.4.4	Dimensiones .....	664
5.4.5	Propiedades del símbolo del pentagrama .....	664
5.4.6	Objetos de extensión .....	664
	Uso del <code>spanner-interface</code> .....	665
	Uso del <code>line-spanner-interface</code> .....	667
5.4.7	Visibilidad de los objetos .....	670
	Eliminar el sello .....	670
	Hacer transparentes los objetos .....	670
	Pintar los objetos de blanco .....	671
	Uso de <code>break-visibility</code> (visibilidad en el salto) .....	671
	Consideraciones especiales .....	673
5.4.8	Estilos de línea .....	676
5.4.9	Rotación de objetos .....	677

Rotación de objetos de presentación.....	677
Rotación de elementos de marcado.....	677
5.5 Trucos avanzados.....	678
5.5.1 Alineación de objetos.....	678
Establecer directamente X-offset y Y-offset.....	679
Uso del <code>side-position-interface</code> .....	679
Uso del <code>self-alignment-interface</code> .....	679
Uso del <code>break-alignable-interface</code> .....	681
5.5.2 Agrupación vertical de objetos gráficos.....	683
5.5.3 Modificación de los sellos.....	683
5.5.4 Modificación de las formas.....	684
Modificación de ligaduras de unión y de expresión.....	684
5.5.5 Modificación de objetos de extensión divididos.....	688
Uso de <code>\alterBroken</code> .....	688
5.5.6 Contenedores unpure-pure.....	690
5.6 Uso de las funciones musicales.....	691
5.6.1 Sintaxis de las funciones de sustitución.....	692
5.6.2 Ejemplos de funciones de sustitución.....	692

## Apéndice A Tablas del manual sobre notación..... 695

A.1 Cuadro de nombres de acordes.....	695
A.2 Modificadores de acorde más usuales.....	696
A.3 Afinaciones de cuerdas predefinidas.....	699
A.4 Diagramas predefinidos de posiciones.....	700
Diagramas para guitarra.....	700
Diagramas para ukelele.....	702
Diagramas para mandolina.....	704
A.5 Tamaños de página predefinidos.....	706
A.6 Instrumentos MIDI.....	709
A.7 Lista de colores.....	710
A.8 La tipografía Emmentaler.....	713
Glifos de clave.....	713
Glifos de indicación de compás.....	713
Glifos de cifras.....	714
Glifos de alteraciones.....	714
Glifos de las cabezas de nota predeterminadas.....	715
Glifos de las cabezas de nota especiales.....	715
Glifos de las cabezas de nota con formas.....	716
Glifos de silencios.....	720
Glifos de corchetes.....	720
Glifos de puntillos.....	721
Glifos de matices dinámicos.....	721
Glifos de inscripciones.....	722
Glifos de flechas.....	724
Glifos de puntas de corchete.....	724
Glifos de pedal.....	725
Glifos de acordeón.....	725
Glifos de ligadura.....	725
Glifos de vaticana.....	725
Glifos de medicea.....	726
Glifos de Hufnagel.....	727
Glifos de mensural.....	728
Glifos de neomensural.....	731
Glifos de Petrucci.....	732

Glifos de Solesmes .....	733
Glifos de la notación del canto kievano .....	733
A.9 Estilos de cabezas de nota .....	734
A.10 Estilos de clave .....	734
Claves estándar .....	735
Clave de pentagramas de percusión .....	736
Claves de pautas de tablatura .....	736
Claves de la música antigua .....	736
A.11 Instrucciones de marcado de texto .....	739
A.11.1 Font .....	740
A.11.2 Align .....	750
A.11.3 Graphic .....	765
A.11.4 Music .....	774
A.11.5 Instrument Specific Markup .....	780
A.11.6 Accordion Registers .....	784
A.11.7 Other .....	789
A.12 Instrucciones de listas de marcado de texto .....	796
A.13 Lista de caracteres especiales .....	798
A.14 Lista de articulaciones .....	800
Indicaciones de articulación .....	800
Indicaciones de adornos .....	800
Indicaciones de calderón .....	801
Indicaciones específicas de ciertos instrumentos .....	801
Indicaciones de repetición .....	801
Indicaciones antiguas .....	802
A.15 Notas de percusión .....	802
A.16 Glosario técnico .....	805
alist (lista-A) .....	805
callback .....	805
closure (cerradura) .....	805
glifo .....	805
grob (objeto gráfico) .....	805
immutable .....	806
interfaz .....	806
lexer (analizador léxico) .....	806
mutable .....	806
output-def (definición de salida) .....	807
parser (analizador sintáctico) .....	807
variable del analizador sintáctico .....	807
prob (objeto de propiedades) .....	807
smob (objeto de Scheme) .....	807
stencil (sello) .....	808
A.17 Todas las propiedades de contexto .....	808
A.18 Propiedades de disposición .....	822
A.19 Funciones musicales disponibles .....	844
A.20 Identificadores de modificación de contextos .....	856
A.21 Predicados de tipo predefinidos .....	856
R5RS primary predicates .....	856
R5RS secondary predicates .....	856
Guile predicates .....	857
LilyPond scheme predicates .....	857
LilyPond exported predicates .....	858
A.22 Funciones de Scheme .....	859

Apéndice B	Hoja de referencia rápida .....	905
Apéndice C	GNU Free Documentation License .....	909
Apéndice D	Índice de instrucciones de LilyPond .....	916
Apéndice E	Índice de LilyPond .....	926

# 1 Notación musical

Este capítulo explica cómo crear notación musical.

## 1.1 Alturas

The image displays two systems of musical notation for piano. The first system consists of two staves (treble and bass clef) in common time (C). The music is in D major, indicated by two sharps (F# and C#). The first staff begins with a piano (*p*) dynamic and a *dolce e molto legato* marking. It features a series of chords and single notes, with a crescendo (*cresc.*) leading to a fortissimo (*sf*) section. The second system starts at measure 38 and continues the piece, maintaining the piano (*p*) dynamic. Both systems include articulation marks such as 'Red.' and '\*' below the notes, and various musical symbols like slurs and ties.

En esta sección se discute cómo especificar la altura de las notas. Este proceso se compone de tres fases: entrada, modificación y salida.

### 1.1.1 Escritura de notas

En esta sección se describe la manera de introducir la altura de las notas. Existen dos formas distintas de colocar las notas en su octava correspondiente: el modo absoluto y el relativo. En casi todas las ocasiones, será más práctico el modo relativo.

#### Escritura de octava absoluta

El nombre de una nota se especifica usando las letras minúsculas de la *a* a la *g*. Las notas cuyos nombres van desde *c* hasta *b* se imprimen en la octava inferior al Do central.

```
{
  \clef bass
  c4 d e f
  g4 a b c
  d4 e f g
}
```

The image shows a musical staff with a bass clef and common time signature. The notes are written as follows: c4 (C2), d (D2), e (E2), f (F2), g4 (G3), a (A3), b (B3), c (C4), d4 (D4), e (E4), f (F4), and g (G4). This sequence demonstrates the absolute octave notation where the octave number is explicitly written for each note.

Se pueden especificar otras octavas mediante una comilla simplee (') o una coma (,). Cada ' eleva la altura en una octava; cada , baja la altura una octava.

```
{
```

```

\clef treble
c'4 e' g' c''
c'4 g b c'
\clef bass
c,4 e, g, c
c,4 g,, b,, c,
}

```



Las marcas de octava comunes se pueden escribir una sola vez sobre una nota de referencia si se emplea `\fixed` antes de la música. Las notas dentro de `\fixed` solo necesitan las marcas de apóstrofo ' o de coma , cuando están por encima o por debajo de la octava de la nota de referencia.

```

{
  \fixed c' {
    \clef treble
    c4 e g c'
    c4 g, b, c
  }
  \clef bass
  \fixed c, {
    c4 e g c'
    c4 g, b, c
  }
}

```



La altura de las notas de la expresión musical que sigue a `\fixed` no resultan afectadas por un `\relative` circundante, que se estudia a continuación.

## Véase también

Glosario musical: Sección “Pitch names” en *Glosario Musical*.

Fragmentos de código: Sección “Pitches” en *Fragmentos de código*.

## Escritura de octava relativa

La entrada de octava absoluta requiere que se especifique la octava para todas y cada una de las notas. En contraste con ello, el modo de entrada de octava relativa especifica cada octava en relación a la nota anterior: si se cambia la octava de una nota ello afectará a todas las notas siguientes.

El modo relativo de notas se debe introducir de forma explícita usando la instrucción `\relative`:

```

\relative altura_inicial expresión_musical

```

En el modo relativo, se supone que cada nota se encuentra lo más cerca posible de la nota anterior. Esto significa que la octava de una nota que está dentro de *expresión\_musical* se calcula como sigue:

- Si no se usa ninguna marca de cambio de octava en una nota, su octava se calcula de forma que el intervalo que forme con la nota anterior sea menor de una quinta. Este intervalo se determina sin considerar las alteraciones.
- Se puede añadir una marca de cambio de octava ' o , para elevar o bajar la altura, respectivamente, en una octava más en relación con la altura calculada sin esta marca.
- Se pueden usar varias marcas de cambio de octava. Por ejemplo, '' y ,, alteran la altura en dos octavas.
- La altura de la primera nota es relativa a *altura\_inicial*. *altura\_inicial* se especifica en modo de octava absoluta. ¿Qué opciones tienen sentido?

c (Do), en cualquier octava

La identificación del Do central con c' es algo bastante básico, por lo que tiende a ser fácil encontrar octavas de c. Si nuestra música comienza con gis (un Sol sostenido) por encima de c'', tendríamos que escribir algo como `\relative { gis'' ... }`

una nota que está una o más octavas de la primera nota de dentro

Escribir `\relative { gis'' ... }` hace que sea fácil determinar la altura absoluta de la primera nota de dentro.

ninguna altura de inicio explícita

La forma `\relative { gis'' ... }` sirve como una versión más compacta de la opción anterior: la primera nota de dentro se escribe ella misma en altura absoluta (esto resulta ser equivalente a elegir f (Fa) como la altura de referencia).

La documentación suele utilizar esta última opción.

Aquí podemos ver el modo relativo en acción:

```
\relative {
  \clef bass
  c d e f
  g a b c
  d e f g
}
```



Las marcas de cambio de octava se utilizan para intervalos mayores de la cuarta:

```
\relative {
  c' g c f,
  c' a, e'' c
}
```





Una serie de notas sin ninguna marca de octava puede, a pesar de todo, abarcar intervalos muy grandes:

```
\relative {
  c f b e
  a d g c
}
```



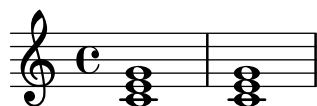
Cuando hay unos bloques `\relative` anidados dentro de otros, el bloque `\relative` más interno comienza con su propia nota de referencia independientemente del `\relative` exterior.

```
\relative {
  c' d e f
  \relative {
    c'' d e f
  }
}
```



`\relative` no tiene efecto sobre los bloques `\chordmode`.

```
\new Staff {
  \relative c''' {
    \chordmode { c1 }
  }
  \chordmode { c1 }
}
```



`\relative` no se permite dentro de los bloques `\chordmode`.

La música que está dentro de un bloque `\transpose` es absoluta, a no ser que se incluya una instrucción `\relative`.

```
\relative {
  d' e
  \transpose f g {
    d e
    \relative {
      d' e
    }
  }
}
```

}



Si el elemento anterior es un acorde, la primera nota del acorde se utiliza para determinar la primera nota del siguiente acorde. Dentro de los acordes, la siguiente nota siempre está en relación a la anterior.

```
\relative {
  c'
  <c e g>
  <c' e g'>
  <c, e, g' '>
}
```



Como se explicó más arriba, la octava de las notas se calcula solamente a partir de sus nombres, sin tener en cuenta alteración alguna. Por tanto, un Mi doble sostenido después de un Si se escribirá más agudo, mientras que un Fa doble sostenido se escribirá más grave. En otras palabras, se considera a la cuarta doble aumentada un intervalo menor que la quinta doble disminuida, independientemente del número de semitonos de cada uno de ellos.

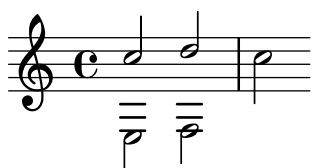
```
\relative {
  c''2 fis
  c2 ges
  b2 eisis
  b2 fes
}
```



En situaciones complejas puede ser útil recuperar una determinada altura, independientemente de lo que hubiera ocurrido con anterioridad. Esto se puede hacer utilizando `\resetRelativeOctave`:

```
\relative {
  <<
  { c''2 d }
  \\\
  { e,,2 f }
  >>
  \resetRelativeOctave c''
  c2
}
```

}



## Véase también

Glosario musical: Sección “fifth” en *Glosario Musical*, Sección “interval” en *Glosario Musical*, Sección “Pitch names” en *Glosario Musical*.

Referencia de la notación: [Comprobación de octava], página 10.

Fragmentos de código: Sección “Pitches” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “RelativeOctaveMusic” en *Referencia de Funcionamiento Interno*.

## Alteraciones accidentales

**Nota:** Las alteraciones accidentales y las armaduras son una frecuente causa de confusión para los nuevos usuarios. En LilyPond, los nombres de las notas especifican las alturas; la armadura y la clave determinan de qué forma se presentan estas alturas. Una nota sin alteración como `c` significa ‘Do natural’, si que le afecten la armadura ni la clave. Para ver más información, consulte Sección “Alturas y armaduras” en *Manual de Aprendizaje*.

Se escribe una nota *sostenida* añadiendo `is` al nombre de la nota, y un *bemol* añadiendo `es`. Como es de esperar, un *dobles sostenido* y un *doble bemol* se obtiene añadiendo `isis` o `eses`. Esta sintaxis deriva de los nombres de las notas en holandés. Para utilizar otros nombres para las alteraciones, consulte [Nombres de las notas en otros idiomas], página 8.

```
\relative c'' { ais1 aes aisis aeses }
```



Una nota natural se introduce como el nombre de la nota, sin más; no se necesita ningún sufijo. Se imprime un símbolo de becuadro si es necesario para cancelar el efecto de una alteración o armadura anterior.

```
\relative c'' { a4 aes a2 }
```



Se pueden escribir medios bemoles y los medios sostenidos; a continuación presentamos una serie de DOs cada vez más agudos:

```
\relative c'' { ceseh1 ces ceh c cih cis cisih }
```



Normalmente las alteraciones accidentales se imprimen automáticamente, pero también puede imprimirlas manualmente. Un alteración recordatoria se puede forzar añadiendo un signo de admiración ! después de la altura de la nota. Se puede obtener una alteración de precaución (o sea, una alteración entre paréntesis) añadiendo el signo de interrogación ? después del nombre de la nota.

```
\relative c'' { cis cis cis! cis? c c c! c? }
```



Las alteraciones sobre notas unidas por ligadura sólo se imprimen al comienzo de un sistema:

```
\relative c'' {
  cis1~ 1~
  \break
  cis
}
```



## Fragmentos de código seleccionados

### *Ocultar las alteraciones sobre notas ligadas al principio de un sistema nuevo*

Aquí se muestra la manera de ocultar las alteraciones de las notas ligadas al comienzo de un sistema nuevo.

```
\relative c'' {
  \override Accidental.hide-tied-accidental-after-break = ##t
  cis1~ cis~
  \break
  cis
}
```



### Evitar que se añadan becuadros adicionales automáticamente

Según las reglas tradicionales de composición tipográfica, se imprime un becuadro antes de un sostenido o un bemol cuando se tiene que cancelar un doble sostenido o un doble bemol anterior en la misma nota. Para modificar este comportamiento a la práctica actual, establezca el valor de la propiedad `extraNatural` a `##f` (falso) dentro del contexto de `Staff`.

```
\relative c'' {
  aeses4 aes ais a
  \set Staff.extraNatural = ##f
  aeses4 aes ais a
}
```



Véase también

Glosario musical: Sección “sharp” en *Glosario Musical*, Sección “flat” en *Glosario Musical*, Sección “double sharp” en *Glosario Musical*, Sección “double flat” en *Glosario Musical*, Sección “Pitch names” en *Glosario Musical*, Sección “quarter tone” en *Glosario Musical*.

Manual de aprendizaje: Sección “Alturas y armaduras” en *Manual de Aprendizaje*.

Referencia de la notación: [Alteraciones accidentales automáticas], página 30, [Alteraciones de anotación (*musica ficta*)], página 473, [Nombres de las notas en otros idiomas], página 8.

Fragmentos de código: Sección “Pitches” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Accidental\_engraver” en *Referencia de Funcionamiento Interno*, Sección “Accidental” en *Referencia de Funcionamiento Interno*, Sección “AccidentalCautionary” en *Referencia de Funcionamiento Interno*, Sección “accidental-interface” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

No existen estándares reconocidos ampliamente para denotar los bemoles de tres cuartos, de manera que los símbolos de LilyPond no se ajustan a ningún estándar.

## Nombres de las notas en otros idiomas

Existen conjuntos predefinidos de nombres de notas y sus alteraciones para algunos idiomas aparte del inglés. La selección del idioma de los nombres de las notas se suele hacer al principio del archivo; el ejemplo siguiente está escrito utilizando los nombres italianos de las notas:

```
\language "italiano"

\relative {
    do' re mi sib
}
```



Los idiomas disponibles y los nombres de las notas que definen son los siguientes:

Idioma	Nombre de las notas
nederlands	c d e f g a bes b

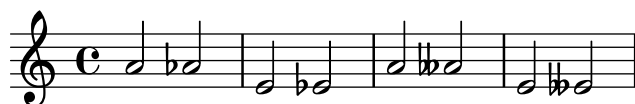
català o	do re mi fa sol la sib si
catalan	
deutsch	c d e f g a b h
english	c d e f g a bf/b-flat b
español o	do re mi fa sol la sib si
espanol	
français	do ré/re mi fa sol la sib si
italiano	do re mi fa sol la sib si
norsk	c d e f g a b h
português o	do re mi fa sol la sib si
portugues	
suomi	c d e f g a b h
svenska	c d e f g a b h
vlaams	do re mi fa sol la sib si

Además de los nombres de las notas, los sufijos de las alteraciones pueden también variar dependiendo del idioma:

Idioma	sostenido	bemol	doble sostenido	doble bemol
nederlands	is	es	isis	eses
català o	d/s	b	dd/ss	bb
catalan				
deutsch	is	es	isis	eses
english	s/-sharp	f/-flat	ss/x/-sharpsharp	ff/-flatflat
español o	s	b	ss/x	bb
espanol				
français	d	b	dd/x	bb
italiano	d	b	dd	bb
norsk	iss/is	ess/es	ississ/isis	essess/eses
português o	s	b	ss	bb
portugues				
suomi	is	es	isis	eses
svenska	iss	ess	ississ	essess
vlaams	k	b	kk	bb

En holandés, **aes** se contrae como **as**, pero las dos formas se aceptan en LilyPond. De forma similar, se aceptan tanto **es** como **ees**. Esto se aplica también a **aeses** / **ases** y a **eeses** / **eses**. A veces se definen solamente estos nombres contraídos en los archivos de idioma correspondientes.

```
\relative c'' { a2 as e es a ases e eses }
```



Algunas músicas utilizan microtonos cuyas alteraciones son fracciones de un sostenido o bemol ‘normales’. La tabla siguiente relaciona los sufijos de los nombres de las notas para las alteraciones de un cuarto de tono; aquí, los prefijos *semi-* y *sesqui-* respectivamente significan ‘medio’ y ‘uno y medio’.

Idioma	semi-sostenido	semi-bemol	sesqui-sostenido	sesqui-bemol
nederlands	ih	eh	isih	eseh
català o	qd/qs	qb	tqd/tqs	tqb
catalan				

deutsch	ih	eh	isih	eseh
english	qs	qf	tqs	tqf
español o espanol	cs	cb	tcs	tcb
français	sd	sb	dsd	bsb
italiano	sd	sb	dsd	bsb
norsk	ih	eh	issih/isih	esseh/eseh
português o portugues	sqt	bqt	stqt	btqt
suomi	ih	eh	isih	eseh
svenska	ih	eh	issih	esseh
vlaams	hk	hb	khk	bhb

En alemán existen contracciones similares del nombre para los microtonos como con las notas normales que se describen arriba.

```
\language "deutsch"
```

```
\relative c'' { asah2 eh aih eish }
```



Casi todos los idiomas presentados aquí están asociados comúnmente con la música clásica occidental, también conocida como *Período de la práctica común*. Sin embargo, también están contempladas las alturas y los sistemas de afinación alternativos: véase Sección 2.10.1 [Notación común para músicas no occidentales], página 497.

## Véase también

Glosario musical: Sección “Pitch names” en *Glosario Musical*, Sección “Common Practice Period” en *Glosario Musical*.

Referencia de la notación: Sección 2.10.1 [Notación común para músicas no occidentales], página 497.

Archivos instalados: `scm/define-note-names.scm`.

Fragmentos de código: Sección “Pitches” en *Fragmentos de código*.

### 1.1.2 Modificación de varias notas a la vez

Esta sección trata de la manera de modificar las alturas.

## Comprobación de octava

En el modo relativo, es fácil olvidar una marca de cambio de octava. Las comprobaciones de octava hacen más fácil encontrar estos errores, mediante la presentación de una advertencia y corrigiendo la octava si la nota se encuentra en una octava distinta de lo esperado.

Para comprobar la octava de una nota, especifique la octava absoluta después del símbolo `=`. Este ejemplo genera un mensaje de advertencia (y corrige la altura) porque la segunda nota es la octava absoluta `d''` en lugar de `d'` como indica la corrección de octava.

```
\relative {
  c''2 d='4 d
  e2 f
```

}



La octava de las notas se puede comprobar también con la instrucción `\octaveCheck altura_de_control`. `altura_de_control` se especifica en modo absoluto. Esto comprueba que el intervalo entre la nota anterior y la `altura_de_control` se encuentra dentro de una cuarta (es decir, el cálculo normal para el modo relativo). Si esta comprobación fracasa, se imprime un mensaje de advertencia. Aunque la nota previa no se modifica, las notas posteriores están en relación al valor corregido.

```
\relative {
  c''2 d
  \octaveCheck c'
  e2 f
}
```



Compare los dos compases siguientes. La primera y tercera comprobaciones de `\octaveCheck` fracasan, pero la segunda es correcta.

```
\relative {
  c''4 f g f

  c4
  \octaveCheck c'
  f
  \octaveCheck c'
  g
  \octaveCheck c'
  f
}
```



## Véase también

Fragmentos de código: Sección “Pitches” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “RelativeOctaveCheck” en *Referencia de Funcionamiento Interno*.

## Transposición

Una expresión musical se puede transportar mediante `\transpose`. La sintaxis es

```
\transpose nota_origen nota_destino expresión_musical
```

Esto significa que la `expresión_musical` se transporta el intervalo que hay entre las notas `nota_origen` y `nota_destino`: cualquier nota con la altura de `nota_origen` se cambia por



`nota_destino` y cualquier otra nota se transporta el mismo intervalo. Las dos notas se introducen en modo absoluto.

**Nota:** La música que está dentro de un bloque `\transpose` es absoluta, a no ser que se incluya una instrucción `\relative` dentro del bloque.

Por ejemplo, tomemos una pieza escrita en la tonalidad de Re mayor. Se puede transportar hacia arriba a Mi mayor; observe que también la armadura de la tonalidad se transporta automáticamente.

```
\transpose d e {
  \relative {
    \key d \major
    d'4 fis a d
  }
}
```



Si una particella escrita en Do (*afinación de concierto normal*) se debe tocar con un clarinete en La (para el que un La se escribe como un Do, y que suena una tercera menor por debajo de lo que está escrito), la particella correspondiente se produce mediante:

```
\transpose a c' {
  \relative {
    \key c \major
    c'4 d e g
  }
}
```



Observe que especificamos `\key c \major` de forma explícita. Si no especificamos ninguna tonalidad, las notas se transportan pero no se imprime la armadura.

`\transpose` distingue entre notas enarmónicas: tanto `\transpose c cis` como `\transpose c des` transportan un semitono hacia arriba. La primera versión imprime sostenidos y las notas no se mueven de su lugar en la escala, en cambio la segunda imprime bemoles de la nota siguiente.

```
music = \relative { c' d e f }
\new Staff {
  \transpose c cis { \music }
  \transpose c des { \music }
}
```



`\transpose` también se puede usar para introducir notas escritas para un instrumento transpositor. Los ejemplos anteriores muestran cómo escribir alturas en Do (o en *afinación de concierto*) y tipografiarlas para un instrumento transpositor, pero también es posible el caso contrario

si, por ejemplo, tenemos un conjunto de partes instrumentales y quiere hacer un guión en Do para el director. Por ejemplo, al introducir música para trompeta en Si bemol que comienza por un Mi en la partitura (Re de concierto), se puede escribir:

```
musicaEnSiBemol = { e4 ... }
\transpose c bes, \musicaEnSiBemol
```

Para imprimir esta música en Fa (por ejemplo, al arreglarla para trompa) puede envolver la música existente con otro `\transpose`:

```
musicaEnSiBemol = { e4 ... }
\transpose f c' { \transpose c bes, \musicaEnSiBemol }
```

Para ver más información sobre instrumentos transpositores, consulte [Transposición de los instrumentos], página 28.

## Fragmentos de código seleccionados

### *Transportar música con el menor número de alteraciones*

Este ejemplo utiliza código de Scheme para forzar las modificaciones enarmónicas de las notas, y así tener el menor número de alteraciones accidentales. En este caso se aplican las siguientes reglas:

- Se quitan las dobles alteraciones
- Si sostenido -> Do
- Mi sostenido -> Fa
- Do bemol -> Si
- Fa bemol -> Mi

De esta forma se selecciona el mayor número de notas enarmónicas naturales.

```
#(define (naturalize-pitch p)
  (let ((o (ly:pitch-octave p))
        (a (* 4 (ly:pitch-alteration p)))
        ;; alteration, a, in quarter tone steps,
        ;; for historical reasons
        (n (ly:pitch-notename p)))
    (cond
      ((and (> a 1) (or (eqv? n 6) (eqv? n 2)))
       (set! a (- a 2))
       (set! n (+ n 1)))
      ((and (< a -1) (or (eqv? n 0) (eqv? n 3)))
       (set! a (+ a 2))
       (set! n (- n 1))))
    (cond
      ((> a 2) (set! a (- a 4)) (set! n (+ n 1)))
      ((< a -2) (set! a (+ a 4)) (set! n (- n 1))))
    (if (< n 0) (begin (set! o (- o 1)) (set! n (+ n 7))))
    (if (> n 6) (begin (set! o (+ o 1)) (set! n (- n 7))))
    (ly:make-pitch o n (/ a 4))))

#(define (naturalize music)
  (let ((es (ly:music-property music 'elements))
        (e (ly:music-property music 'element))
        (p (ly:music-property music 'pitch)))
    (if (pair? es)
```

```

      (ly:music-set-property!
       music 'elements
       (map naturalize es)))
(if (ly:music? e)
    (ly:music-set-property!
     music 'element
     (naturalize e)))
(if (ly:pitch? p)
    (begin
      (set! p (naturalize-pitch p))
      (ly:music-set-property! music 'pitch p)))
music))

naturalizeMusic =
#(define-music-function (m)
  (ly:music?)
  (naturalize m))

music = \relative c' { c4 d e g }

\score {
  \new Staff {
    \transpose c ais { \music }
    \naturalizeMusic \transpose c ais { \music }
    \transpose c deses { \music }
    \naturalizeMusic \transpose c deses { \music }
  }
  \layout { }
}

```



## Véase también

Referencia de la notación: [Transposición de los instrumentos], página 28, [Inversión], página 14, [Transformaciones modales], página 16, [Escritura de octava relativa], página 2, [Retrogradación], página 15.

Fragmentos de código: Sección “Pitches” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TransposedMusic” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

La conversión relativa no afecta a las secciones `\transpose`, `\chordmode` ni `\relative` dentro de su argumento. Para usar el modo relativo dentro de música transportada, se debe colocar otro `\relative` dentro de `\transpose`.

No se imprimen alteraciones accidentales triples cuando se usa `\transpose`. En lugar de ello, se usará una nota ‘equivalente enarmónicamente’ (p.ej., Re bemol en lugar de Mi triple bemol).

## Inversión

Una expresión musical se puede invertir y transportar en una sola operación con:

```
\inversion nota-pivote nota-destino expresión_musical
```

La *expresión\_musical* se invierte intervalo a intervalo alrededor de *nota-pivote*, y después se transporta de manera que *nota-pivote* se hace corresponder con *nota-destino*.

```
music = \relative { c' d e f }
\new Staff {
  \music
  \inversion d' d' \music
  \inversion d' ees' \music
}
```



**Nota:** Los motivos a invertir deberían venir expresados en modo absoluto o convertirse previamente a la forma absoluta encerrándolos en un bloque `\relative`.

## Véase también

Referencia de la notación: [Transformaciones modales], página 16, [Retrogradación], página 15, [Transposición], página 11.

## Retrogradación

Se puede revertir una expresión musical para producir su retrogradación:

```
music = \relative { c'8. ees16( fis8. a16 b8.) gis16 f8. d16 }
\new Staff {
  \music
  \retrograde \music
}
```



## Advertencias y problemas conocidos

`\retrograde` es una herramienta más bien sencilla. Puesto que muchos eventos se invierten como en un espejo en lugar de intercambiarse, los trucajes y los modificadores direccionales tienen que añadirse en los elementos de cierre correspondientes: `^` (debe terminar con `^`), todos y cada uno de los `\<` o `\cresc` han de terminar en `\!` o en `\endcresc`, y todo `\>` o `\decr` debe terminar en `\enddecr`. Las instrucciones o sobreescrituras que cambien propiedades y que tengan un efecto duradero, probablemente produzcan sorpresas.

## Véase también

Referencia de la notación: [Inversión], página 14, [Transformaciones modales], página 16, [Transposición], página 11.

## Transformaciones modales

En una composición musical que está basada en una escala, con frecuencia se transforman los motivos de diversas formas. Un motivo se puede *transportar* para que comience en distintos lugares de la escala o puede *invertirse* alrededor de una nota pivote de la escala. También se puede revertir para producir una *retrogradación*, véase [Retrogradación], página 15.

**Nota:** Cualquier nota que no entre en la escala dada, se dejará sin transformar.

### *Transposición modal*

Se puede transportar un motivo dentro de una escala dada con:

```
\modalTranspose nota-origen nota-destino escala motivo
```

Las notas del *motivo* se desplazan dentro de la *escala* el número de grados de la escala dados por el intervalo entre *nota-destino* y *nota-origen*:

```
diatonicScale = \relative { c' d e f g a b }
motif = \relative { c'8 d e f g a b c }
```

```
\new Staff {
  \motif
  \modalTranspose c f \diatonicScale \motif
  \modalTranspose c b, \diatonicScale \motif
}
```



Se pueden especificar escalas ascendentes de cualquier longitud y con cualesquiera intervallos:

```
pentatonicScale = \relative { ges aes bes des ees }
motif = \relative { ees'8 des ges,4 <ges' bes,> <ges bes,> }
```

```
\new Staff {
  \motif
  \modalTranspose ges ees' \pentatonicScale \motif
}
```



Cuando se utiliza con una escala cromática, `\modalTranspose` tiene un efecto similar a `\transpose`, pero con la posibilidad de especificar los nombres de las notas que se quieren usar:

```
chromaticScale = \relative { c' cis d dis e f fis g gis a ais b }
motif = \relative { c'8 d e f g a b c }
```

```
\new Staff {
  \motif
  \transpose c f \motif
  \modalTranspose c f \chromaticScale \motif
}
```

}



### *Inversión modal*

Se puede invertir un motivo dentro de una escala dada alrededor de una nota pivote dada y transportada al mismo tiempo en una única operación, con:

```
\modalInversion nota-pivote nota-destino escala motivo
```

Las notas del *motivo* se colocan al mismo número de grados de distancia de la escala a partir de la *nota-pivote* dentro de la *escala*, pero en la dirección opuesta, y el resultado se desplaza después dentro de la *escala* el número de grados de la escala dados por el intervalo entre la *nota-destino* y la *nota-pivote*.

Así pues, para invertir sencillamente alrededor de una nota de la escala, utilice el mismo valor para *nota-pivote* y *nota-destino*:

```
octatonicScale = \relative { ees' f fis gis a b c d }
motif = \relative { c'8. ees16 fis8. a16 b8. gis16 f8. d16 }
```

```
\new Staff {
  \motif
  \modalInversion fis' fis' \octatonicScale \motif
}
```



Para invertir alrededor de una nota pivote entre dos notas de la escala, invierta alrededor de una de las notas y después transporte en un grado de la escala. Las dos notas especificadas se pueden interpretar como que horquillan entre ellas a la nota pivote:

```
scale = \relative { c' g' }
motive = \relative { c' c g' c, }
```

```
\new Staff {
  \motive
  \modalInversion c' g' \scale \motive
}
```



La operación combinada de inversión y retrogradación produce la inversión retrógrada:

```
octatonicScale = \relative { ees' f fis gis a b c d }
motif = \relative { c'8. ees16 fis8. a16 b8. gis16 f8. d16 }
```

```
\new Staff {
  \motif
  \retrograde \modalInversion c' c' \octatonicScale \motif
}
```

}



## Véase también

Referencia de la notación: [Inversión], página 14, [Retrogradación], página 15, [Transposición], página 11.

### 1.1.3 Imprimir las alturas

Esta sección trata de cómo alterar la presentación de la altura de las notas.

## Clave

Sin ninugna instrucción explícita, la clave predeterminada en LilyPond es la clave “treble” (o clave de *Sol*).

```
c'2 c'
```



Sin embargo, se puede cambiar la clave usando la instrucción `\clef` y el nombre de la clave correspondiente. En los siguientes ejemplos se muestra la posición del *Do central* en distintas claves.

```
\clef treble
c'2 c'
\clef alto
c'2 c'
\clef tenor
c'2 c'
\clef bass
c'2 c'
```



Para ver el repertorio completo de los posibles nombres para las claves, consulte Sección A.10 [Estilos de clave], página 734.

Las claves especiales como las utilizadas en la música *antigua*, se describen en [Claves de la música mensural], página 469, y en [Claves de canto gregoriano], página 476. La música que requiere claves de tablatura se estudia en [Tablaturas predeterminadas], página 371, y en [Tablaturas personalizadas], página 386.

Para emplear claves en las notas guía, véanse las instrucciones `\cueClef` y `\cueDuringWithClef` en [Formateo de las notas guía], página 227.

Al añadir `_8` o `^8` al nombre de la clave, la clave se transpone una octava hacia abajo o hacia arriba, respectivamente, y `_15` y `^15` la transpone dos octavas. Si es necesario se pueden usar otros números enteros. El argumento *nombre\_de\_clave* se debe encerrar entre comillas si contiene caracteres no alfabéticos:

```
\clef treble
```

```

c'2 c'
\clef "treble_8"
c'2 c'
\clef "bass^15"
c'2 c'
\clef "alto_2"
c'2 c'
\clef "G_8"
c'2 c'
\clef "F^5"
c'2 c'

```



Se puede obtener una indicación de octavación opcional encerrando el argumento numérico entre paréntesis o corchetes rectos:

```

\clef "treble_(8)"
c'2 c'
\clef "bass^[15]"
c'2 c'

```



Las alturas se imprimen como si el argumento numérico no tuviera los paréntesis o corchetes.

De forma predeterminada, un cambio de clave que se produce en un salto de línea provoca que el símbolo de clave nuevo se imprima al final de la línea anterior, como una clave *de cortesía*, así como al principio del siguiente. Esta clave *de cortesía* se puede suprimir.

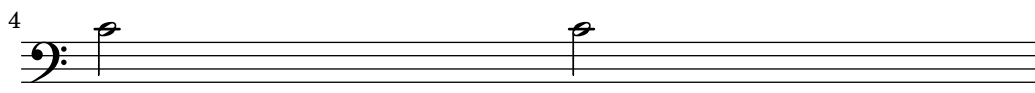
```

\clef treble { c'2 c' } \break
\clef bass { c'2 c' } \break
\clef alto
\set Staff.explicitClefVisibility = #end-of-line-invisible
{ c'2 c' } \break
\unset Staff.explicitClefVisibility
\clef bass { c'2 c' } \break

```







De forma predeterminada, una clave que se ha impreso previamente no se vuelve a imprimir si se emplea de nuevo la misma instrucción `\clef`, y se ignora. La instrucción `\set Staff.forceClef = ##t` modifica este comportamiento.

```
\clef treble
c'1
\clef treble
c'1
\set Staff.forceClef = ##t
c'1
\clef treble
c'1
```



Para ser más exactos, no es la propia instrucción `\clef` la que imprime una clave. Más bien, fija o cambia una propiedad del grabador de clave `Clef_engraver`, que a continuación decide por sí mismo si imprimir o no una clave en el pentagrama actual. La propiedad `forceClef` sobrescribe esta decisión localmente para reimprimir la clave una vez más.

Cuando hay un cambio de clave manual, el glifo de la clave modificada es más pequeño de lo normal. Se puede sobrescribir este comportamiento.

```
\clef "treble"
c'1
\clef "bass"
c'1
\clef "treble"
c'1
\override Staff.Clef.full-size-change = ##t
\clef "bass"
c'1
\clef "treble"
c'1
\revert Staff.Clef.full-size-change
\clef "bass"
c'1
\clef "treble"
c'1
```



## Fragmentos de código seleccionados

### *Trucaje de las propiedades de clave*

La modificación del glifo de la clave, su posición o su octavación, no cambian 'per se' la posición de las siguientes notas del pentagrama. Para conseguir armaduras de tonalidad sobre las líneas del pentagrama adecuadas, también debe especificarse `middleCClefPosition`, con valores positivos

o negativos que mueven el Do **central** hacia arriba o hacia abajo, respectivamente, en relación con la línea central del pentagrama (usualmente la tercera).

Por ejemplo, la instrucción `\clef "treble_8"` equivale a un ajuste de `clefGlyph`, `clefPosition` (que controla la posición vertical de la clave sobre el pentagrama), `middleCPosition` y `clefTransposition`. Se imprime una clave cada vez que se modifica cualquiera de las propiedades excepto `middleCPosition`.

Los siguientes ejemplos muestran las posibilidades cuando se ajustan estas propiedades manualmente. En la primera línea, los cambios manuales preservan el posicionamiento relativo estándar de las claves y las notas, pero no lo hacen en la segunda línea.

```
{
  % The default treble clef
  \key f \major
  c'1
  % The standard bass clef
  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  \set Staff.middleCPosition = #6
  \set Staff.middleCClefPosition = #6
  \key g \major
  c'1
  % The baritone clef
  \set Staff.clefGlyph = #"clefs.C"
  \set Staff.clefPosition = #4
  \set Staff.middleCPosition = #4
  \set Staff.middleCClefPosition = #4
  \key f \major
  c'1
  % The standard choral tenor clef
  \set Staff.clefGlyph = #"clefs.G"
  \set Staff.clefPosition = #-2
  \set Staff.clefTransposition = #-7
  \set Staff.middleCPosition = #1
  \set Staff.middleCClefPosition = #1
  \key f \major
  c'1
  % A non-standard clef
  \set Staff.clefPosition = #0
  \set Staff.clefTransposition = #0
  \set Staff.middleCPosition = #-4
  \set Staff.middleCClefPosition = #-4
  \key g \major
  c'1 \break

  % The following clef changes do not preserve
  % the normal relationship between notes, key signatures
  % and clefs:

  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  c'1
  \set Staff.clefGlyph = #"clefs.G"
```

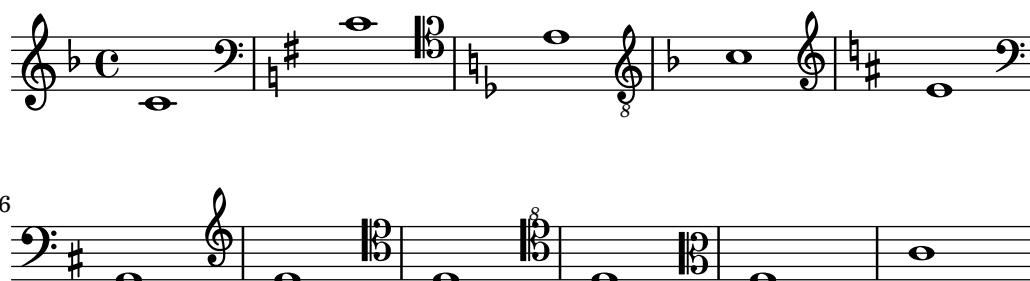
```

c'1
\set Staff.clefGlyph = #"clefs.C"
c'1
\set Staff.clefTransposition = #7
c'1
\set Staff.clefTransposition = #0
\set Staff.clefPosition = #0
c'1

% Return to the normal clef:

\set Staff.middleCPosition = #0
c'1
}

```



## Véase también

Referencia de la notación: [Claves de la música mensural], página 469, [Claves de canto gregoriano], página 476, [Tablaturas predeterminadas], página 371, [Tablaturas personalizadas], página 386, [Formateo de las notas guía], página 227.

Archivos instalados: `scm/parser-clef.scm`.

Fragmentos de código: Sección “Pitches” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Clef-engraver” en *Referencia de Funcionamiento Interno*, Sección “Clef” en *Referencia de Funcionamiento Interno*, Sección “ClefModifier” en *Referencia de Funcionamiento Interno*, Sección “clef-interface” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Los números de octavación adjuntos a las claves se tratan como grobs distintos. Así pues, cualquier `\override` (sobreescritura) efectuada al objeto *Clef* deberá aplicarse también al grob *ClefModifier* como un `\override` diferente.

```

\new Staff \with {
  \override Clef.color = #blue
  \override ClefModifier.color = #red
}

\clef "treble_8" c'4

```



## Armadura de la tonalidad

**Nota:** Las alteraciones accidentales y las armaduras son una frecuente causa de confusión para los nuevos usuarios. En LilyPond, los nombres de las notas son el código de entrada en bruto; la armadura y la clave determinan de qué forma se presenta este código en bruto. Una nota sin alteración como `c` significa ‘Do natural’, si que le afecten la armadura ni la clave. Para ver más información, consulte Sección “Alturas y armaduras” en *Manual de Aprendizaje*.

La armadura indica la tonalidad en que se toca una pieza. Está denotada por un conjunto de alteraciones (bemoles o sostenidos) al comienzo del pentagrama. El establecimiento o modificación de la armadura se hace con la instrucción `\key`:

```
\key nota modo
```

Aquí, *modo* debe ser `\major` o `\minor` para obtener la tonalidad *nota* mayor o *nota* menor, respectivamente. También puede usar los nombres estándar de modo (también conocidos como *modos eclesiásticos*): `\ionian` (jónico), `\dorian` (dórico), `\phrygian` (frigio), `\lydian` (lidio), `\mixolydian` (mixolidio), `\aeolian` (eolio) y `\locrian` (locrio).

```
\relative {
  \key g \major
  fis''1
  f
  fis
}
```



Se pueden definir modos adicionales, escribiendo una lista con la alteración que lleva cada nota de la escala cuando el modo comienza en Do.

```
freygish = #`((0 . ,NATURAL) (1 . ,FLAT) (2 . ,NATURAL)
              (3 . ,NATURAL) (4 . ,NATURAL) (5 . ,FLAT) (6 . ,FLAT))
```

```
\relative {
  \key c \freygish c'4 des e f
  \bar "||" \key d \freygish d es fis g
}
```



Las alteraciones de la armadura de tonalidad se pueden imprimir en distinta octava que en sus posiciones tradicionales, o en más de una octava, usando las propiedades `flat-positions` y `sharp-positions` de `KeySignature`. Las entradas en estas propiedades especifican el rango de posiciones de pentagrama en que se imprimirán las alteraciones. Si en una entrada se especifica una sola posición, las alteraciones se colocan dentro de la octava que termina en dicha posición del pentagrama.

```
\override Staff.KeySignature.flat-positions = #'((-5 . 5))
\override Staff.KeyCancellation.flat-positions = #'((-5 . 5))
```

```

\clef bass \key es \major es g bes d'
\clef treble \bar "||" \key es \major es' g' bes' d''

\override Staff.KeySignature.sharp-positions = #'(2)
\bar "||" \key b \major b' fis' b'2

```



## Fragmentos de código seleccionados

### *Evitar que se impriman becuadros cuando cambia la armadura*

Cuando cambia la armadura de la tonalidad, se imprimen becuadros automáticamente para cancelar las alteraciones de las armaduras anteriores. Esto se puede evitar estableciendo al valor “falso” la propiedad `printKeyCancellation` del contexto `Staff`.

```

\relative c' {
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
  \set Staff.printKeyCancellation = ##f
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
}

```



### *Armaduras de tonalidad no tradicionales*

La muy utilizada instrucción `\key` establece la propiedad `keyAlterations` dentro del contexto `Staff`. Para crear armaduras de tonalidad no estándar, ajuste esta propiedad directamente.

El formato de esta instrucción es una lista:

```

\set Staff.keyAlterations = #`(((octava . paso) . alteración) ((octava . paso) .
alteración) ...)

```

donde, para cada elemento dentro de la lista, `octava` especifica la octava (siendo cero la octava desde el Do central hasta el Si por encima), `paso` especifica la nota dentro de la octava (cero significa Do y 6 significa Si), y `alteración` es `,SHARP`, `,FLAT`, `,DOUBLE-SHARP` etc.

De forma alternativa, el uso del formato más conciso `(paso . alteración)` para cada elemento de la lista especifica que la misma alteración debe estar en todas las octavas. Para escalas microtonales en las que un “sostenido” no son 100 cents, `alteración` se refiere a la proporción de un tono entero de 200 cents.

```

\include "arabic.ly"
\relative do' {
  \set Staff.keyAlterations = #`((0 . ,SEMI-FLAT)
    (1 . ,SEMI-FLAT)
    (2 . ,FLAT)

```

```

(5 . ,FLAT)
(6 . ,SEMI-FLAT))

%\set Staff.extraNatural = ##f
re reb \down reb resd
dod dob dosd \down dob |
dobsb dodsd do do |
}

```



## Véase también

Glosario musical: Sección “church mode” en *Glosario Musical*, Sección “scordatura” en *Glosario Musical*.

Manual de aprendizaje: Sección “Alturas y armaduras” en *Manual de Aprendizaje*.

Fragmentos de código: Sección “Pitches” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “KeyChangeEvent” en *Referencia de Funcionamiento Interno*, Sección “Key\_engraver” en *Referencia de Funcionamiento Interno*, Sección “Key\_performer” en *Referencia de Funcionamiento Interno*, Sección “KeyCancellation” en *Referencia de Funcionamiento Interno*, Sección “KeySignature” en *Referencia de Funcionamiento Interno*, Sección “key-signature-interface” en *Referencia de Funcionamiento Interno*.

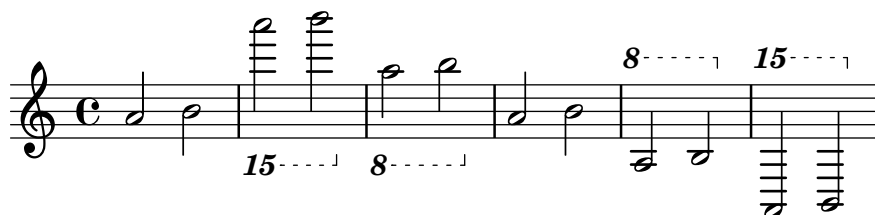
## Corchetes de octava

Los *corchetes de Ottava* introducen un trasporte adicional de una octava para el pentagrama.

```

\relative c'' {
  a2 b
  \ottava #-2
  a2 b
  \ottava #-1
  a2 b
  \ottava #0
  a2 b
  \ottava #1
  a2 b
  \ottava #2
  a2 b
}

```



De forma predeterminada, solo se imprime un número al comienzo del corchete. Este ajuste se puede modificar para que incluya un ordinal abreviado, ya sea como superíndice o en letras de tipo normal (este último estilo se usaba anteriormente por defecto); la tipografía predeterminada en negrita de estos caracteres se puede también alterar, como se explica en [Seleccionar la tipografía y su tamaño], página 266.

El ejemplo siguiente muestra diversas opciones, así como la forma de volver al comportamiento predeterminado actualmente:

```
\relative c' {
  \ottava #1
  a'2 b
  \ottava #2
  a'2 b
  \bar "||"
  \set Staff.ottavationMarkups = #ottavation-ordinals
  \ottava #1
  a,2 b
  \ottava #2
  a'2 b
  \bar "||"
  \override Staff.OttavaBracket.font-series = #'medium
  \set Staff.ottavationMarkups = #ottavation-simple-ordinals
  \ottava #1
  a,2 b
  \ottava #2
  a'2 b
  \bar "||"
  \revert Staff.OttavaBracket.font-series
  \set Staff.ottavationMarkups = #ottavation-numbers
  \ottava #1
  a,2 b
  \ottava #2
  a'2 b
}
```



## Fragmentos de código seleccionados

### *Changing ottava text*

Internally, `\ottava` sets the properties `ottavation` (for example, to `8va` or `8vb`) and `middleCPosition`. To override the text of the bracket, set `ottavation` after invoking `\ottava`.

Short text is especially useful when a brief ottava is used.

```
{
  c'2
  \ottava #1
  \set Staff.ottavation = #"8"
  c''2
  \ottava #0
  c'1
  \ottava #1
  \set Staff.ottavation = #"Text"
  c''1
}
```

}

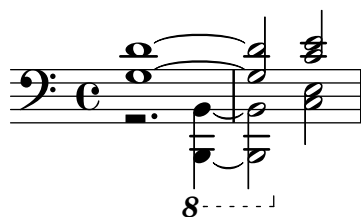


### *Añadir una indicación de octava alta a una sola voz*

Si tiene más de una voz en el mismo pentagrama, el cambio de octavación de una voz transportará la posición de las notas en todas las voces mientras dure el corchete de octava. Si la octavación se quiere aplicar a una voz solamente, se deben ajustar explícitamente la `middleCPosition` y el corchete de octava. En este fragmento de código, el valor de `middleCPosition` para la clave de Fa es normalmente 6, seis posiciones por encima de la línea del Do central, de manera que en la porción de 8va el valor de `middleCPosition` es aún 7 posiciones (una octava) más alta.

```
\layout {
  \context {
    \Staff
    \remove Ottava_spanner_engraver
  }
  \context {
    \Voice
    \consists Ottava_spanner_engraver
  }
}

{
  \clef bass
  << { <g d'>1~ q2 <c' e'> }
  \\
  {
    r2.
    \ottava -1
    <b,,, b,,,>4 ~ |
    q2
    \ottava 0
    <c e>2
  }
  >>
}
```



### *Modifying the Ottava spanner slope*

It is possible to change the slope of the Ottava spanner.

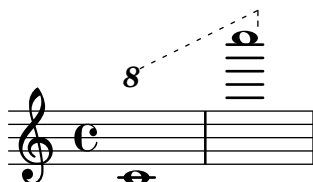
```
\relative c'' {
  \override Staff.OttavaBracket.stencil = #ly:line-spanner::print
  \override Staff.OttavaBracket.bound-details =
```



```

#`((left . ((Y . 0) ; Change the integer here
      (attach-dir . ,LEFT)
      (padding . 0)
      (stencil-align-dir-y . ,CENTER)))
  (right . ((Y . 5) ; Change the integer here
      (padding . 0)
      (attach-dir . ,RIGHT)
      (text . ,(make-draw-dashed-line-markup
        (cons 0 -1.2)))))
\override Staff.OttavaBracket.left-bound-info =
  #ly:line-spanner::calc-left-bound-info-and-text
\override Staff.OttavaBracket.right-bound-info =
  #ly:line-spanner::calc-right-bound-info
\ottava #1
c1
c'''1
}

```



## Véase también

Glosario musical: Sección “octavation” en *Glosario Musical*.

Referencia de la notación: [Seleccionar la tipografía y su tamaño], página 266.

Fragmentos de código: Sección “Pitches” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Ottava\_spanner\_engraver” en *Referencia de Funcionamiento Interno*, Sección “OttavaBracket” en *Referencia de Funcionamiento Interno*, Sección “ottava-bracket-interface” en *Referencia de Funcionamiento Interno*.

## Transposición de los instrumentos

Al tipografiar partituras donde participan instrumentos transpositores, ciertas partes se pueden tipografiar en un tono distinto del *tono de concierto*. En estos casos, se debe especificar la tonalidad del *instrumento transpositor*; de otro modo, la salida MIDI y las notas guía en otras partes producirían alturas incorrectas. Para ver más información sobre partes citadas como guía, consulte [Citar otras voces], página 224.

`\transposition pitch`

El tono usado para `\transposition` debe corresponderse con el sonido real que se oye cuando el instrumento transpositor interpreta un Do central `c'` escrito en el pentagrama. Esta nota se escribe en altura absoluta, por tanto un instrumento que produce un sonido real un tono más agudo que la música impresa (un instrumento en Re) debe usar `\transposition d'`. La instrucción `\transposition` se debe usar *solamente* si las notas *no* se van a escribir en afinación de concierto.

A continuación pueden verse algunas notas para violín y para clarinete en Si bemol, donde las partes se han introducido usando las notas y la armadura tal y como aparecen en la partitura del director. Lo que tocan los dos instrumentos está sonando al unísono.

```

\new GrandStaff <<
  \new Staff = "violin" \with {

```

```

    instrumentName = "Vln"
    midiInstrument = "violin"
}
\relative c'' {
  % not strictly necessary, but a good reminder
  \transposition c'
  \key c \major
  g4( c8) r c r c4
}
\new Staff = "clarinet" \with {
  instrumentName = \markup { Cl (B\flat) }
  midiInstrument = "clarinet"
}
\relative c'' {
  \transposition bes
  \key d \major
  a4( d8) r d r d4
}
>>

```



La `\transposition` se puede cambiar durante la pieza. Por ejemplo, un clarinetista puede cambiar del clarinete en La al clarinete en Si bemol.

```

flute = \relative c'' {
  \key f \major
  \cueDuring "clarinet" #DOWN {
    R1 _\markup\tiny "clarinet"
    c4 f e d
    R1 _\markup\tiny "clarinet"
  }
}
clarinet = \relative c'' {
  \key aes \major
  \transposition a
  aes4 bes c des
  R1^\markup { muta in B\flat }
  \key g \major
  \transposition bes
  d2 g,
}
\addQuote "clarinet" \clarinet
<<
  \new Staff \with { instrumentName = "Flute" }
  \flute
  \new Staff \with { instrumentName = "Cl (A)" }
  \clarinet

```

&gt;&gt;

Flute

clarinet

Cl (A)

muta in B  $\flat$

clarinet

## Véase también

Glosario musical: Sección “concert pitch” en *Glosario Musical*, Sección “transposing instrument” en *Glosario Musical*.

Referencia de la notación: [Citar otras voces], página 224, [Transposición], página 11.

Fragmentos de código: Sección “Pitches” en *Fragmentos de código*.

## Alteraciones accidentales automáticas

Existen muchas convenciones distintas sobre la forma de tipografiar las alteraciones. LilyPond proporciona una función para especificar qué estilo de alteraciones usar. Esta función se invoca como sigue:

```
\new Staff <<
  \accidentalStyle voice
  { ... }
>>
```

El estilo de alteraciones se aplica al **Staff** en curso de forma predeterminada (con la excepción de los estilos **piano** y **piano-cautionary**, que se explican más adelante). Opcionalmente, la función puede tomar un segundo argumento que determina en qué ámbito se debe cambiar el estilo. Por ejemplo, para usar el mismo estilo en todos los pentagramas del **StaffGroup** en curso, use

```
\accidentalStyle StaffGroup.voice
```

Están contemplados los siguientes estilos de alteración. Para dar una muestra de cada uno de los estilos, utilizamos el ejemplo siguiente:

```
musicA = {
  <<
    \relative {
      cis''8 fis, bes4 <a cis>8 f bis4 |
      cis2. <c, g'>4 |
    }
    \\
    \relative {
      ais'2 cis, |
      fis8 b a4 cis2 |
    }
  >>
}
```

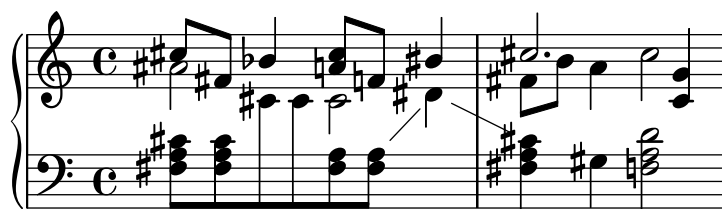
```
musicB = {
  \clef bass
  \new Voice {
    \voiceTwo \relative {
```

```

    <fis a cis>8[ <fis a cis>
    \change Staff = up
    cis' cis
    \change Staff = down
    <fis, a> <fis a>]
    \showStaffSwitch
    \change Staff = up
    dis'4 |
    \change Staff = down
    <fis, a cis>4 gis <f a d>2 |
  }
}
}

\new PianoStaff {
  <<
    \new Staff = "up" {
      \accidentalStyle default
      \musicA
    }
    \new Staff = "down" {
      \accidentalStyle default
      \musicB
    }
  >>
}

```



Observe que las últimas líneas de este ejemplo se pueden sustituir por las siguientes, siempre y cuando queramos usar el mismo estilo en los dos pentagramas.

```

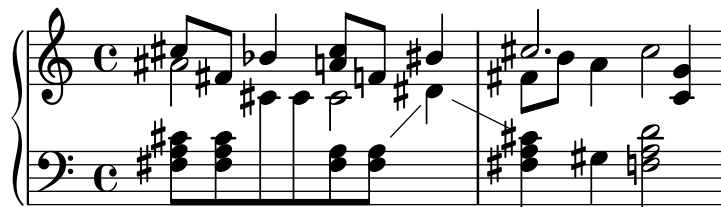
\new PianoStaff {
  <<
    \new Staff = "up" {
      %% cambie la línea siguiente como desee:
      \accidentalStyle Score.default
      \musicA
    }
    \new Staff = "down" {
      \musicB
    }
  >>
}

```

default (predeterminado)

Es el comportamiento de composición tipográfica predeterminado. Corresponde a la práctica común del s. XVIII: las alteraciones accidentales se recuerdan hasta el final del compás en el que aparecen y sólo en la misma octava. Así, en el ejemplo

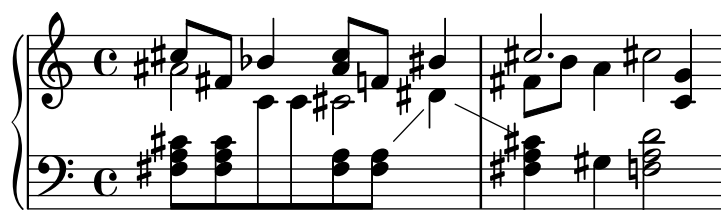
siguiente, no se imprimen becuadros antes del Si natural en el segundo compás ni en el último Do:



#### voice (voz)

El comportamiento normal es recordar las alteraciones accidentales al nivel de **Staff**. En este estilo, sin embargo, se tipografían las alteraciones individualmente para cada voz. Aparte de esto, la regla es similar a **default**.

Como resultado, las alteraciones de una voz no se cancelan en las otras voces, lo que con frecuencia lleva a un resultado no deseado: en el ejemplo siguiente, es difícil determinar si el segundo La se debe tocar natural o sostenido. Por tanto, la opción **voice** se debe usar sólo si las voces se van a leer individualmente por músicos distintos. Si el pentagrama va a utilizarse por parte de un solo músico (p.ej., un director, o en una partitura de piano), entonces se deben usar en su lugar los estilos **modern** o **modern-cautionary**.



#### modern (moderno)

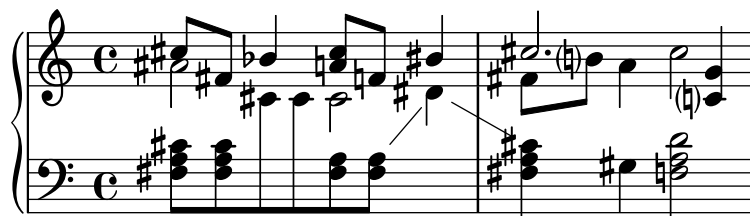
Esta regla corresponde a la práctica común del s. XX. Omite algunos becuadros adicionales, que tradicionalmente se imprimían precediendo a un sostenido que sigue a un doble sostenido, o a un bemol que sigue a un doble bemol. La regla **modern** imprime las mismas alteraciones que el estilo **default**, con dos adiciones que sirven para evitar la ambigüedad: después de alteraciones temporales se imprimen indicaciones de cancelación también en el compás siguiente (para notas en la misma octava) y, en el mismo compás, para notas en octavas distintas. De aquí los becuadros antes del Si natural y del Do en el segundo compás del pentagrama superior:



#### modern-cautionary (moderno de precaución)

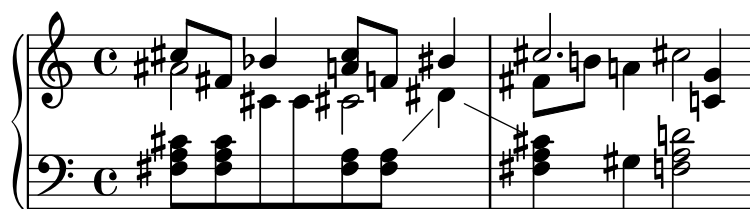
Esta regla es similar a **modern**, pero las alteraciones ‘añadidas’ se imprimen como alteraciones de precaución (entre paréntesis). También se pueden impri-

mir en un tamaño distinto sobrescribiendo la propiedad `font-size` del objeto `AccidentalCautionary`.



#### modern-voice (moderno, para voces)

Esta regla se usa para que puedan leer las alteraciones en varias voces, tanto músicos que tocan una voz como músicos que tocan todas las voces. Se imprimen las alteraciones para cada voz, pero *se cancelan* entre voces dentro del mismo `Staff`. Por tanto, el La en el último compás se cancela porque la cancelación anterior estaba en una voz distinta, y el Re en el pentagrama inferior se cancela a causa de la alteración en otra voz en el compás previo:



#### modern-voice-cautionary (moderno, voz, de precaución)

Esta regla es la misma que `modern-voice`, pero con las alteraciones añadidas (las que el estilo `voice` no imprime) compuestas como de precaución. Incluso aunque todas las alteraciones impresas por el estilo `default` *son* impresas con esta regla, algunas de ellas se tipografían como de precaución.



#### piano

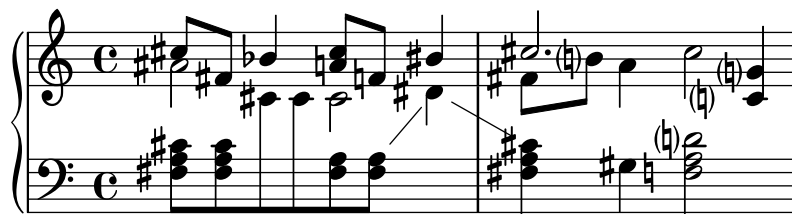
Esta regla refleja la práctica del s.XX para la notación de piano. Su comportamiento es muy similar al estilo `modern`, pero aquí las alteraciones también se cancelan entre distintos pentagramas del mismo grupo `GrandStaff` o `PianoStaff`, de ahí todas las cancelaciones de las últimas notas.

Este estilo de alteración se aplica de manera predeterminada al grupo `GrandStaff` o `PianoStaff` en curso.



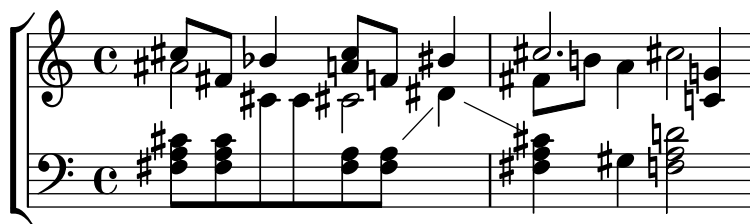
**piano-cautionary** (piano, de precaución)

Igual que `\accidentalStyle piano` pero con las alteraciones añadidas compuestas como de precaución.

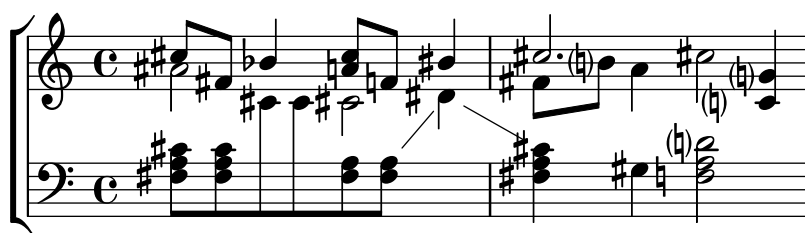
**choral**

Esta regla es una combinación de los estilos `modern-voice` y `piano`. Muestra todas las alteraciones accidentales requeridas para los cantantes que siguen solo su propia voz, así como las alteraciones adicionales necesarias para los lectores que siguen simultáneamente todas las voces de un `ChoirStaff` completo.

Este estilo de alteraciones se aplica de forma predeterminada al contexto `ChoirStaff` actual.

**choral-cautionary**

Es lo mismo que `choral` pero con las alteraciones adicionales impresas como de precaución (o cortesía).

**neo-modern**

Esta regla reproduce una práctica común en la música contemporánea: las alteraciones accidentales se imprimen como en `modern`, pero se vuelven a imprimir si aparece la misma nota otra vez en el mismo compás (excepto si la nota se repite inmediatamente).

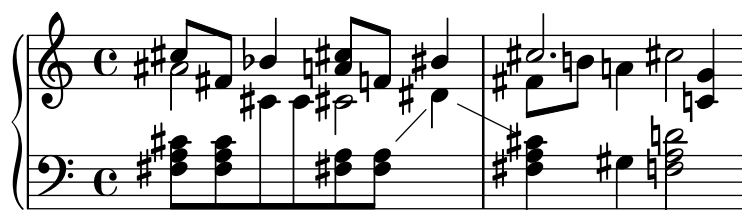


**neo-modern-cautionary**

Esta regla es similar a **neo-modern**, pero las alteraciones ‘adicionales’ se imprimen como alteraciones de precaución (con paréntesis). También se pueden imprimir en un tamaño distinto sobreescribiendo la propiedad `font-size` del objeto `AccidentalCautionary`.

**neo-modern-voice**

Esta regla se usa para alteraciones accidentales sobre varias voces que se han de leer por parte de músicos que tocan una voz, así como por músicos que tocan todas las voces. Las alteraciones se imprimen para cada voz como con **neo-modern**, pero se cancelan para otras voces que están en el mismo pentagrama `Staff`.

**neo-modern-voice-cautionary**

Esta regla es similar a **neo-modern-voice**, pero las alteraciones adicionales se imprimen como alteraciones de precaución.

**dodecaphonic**

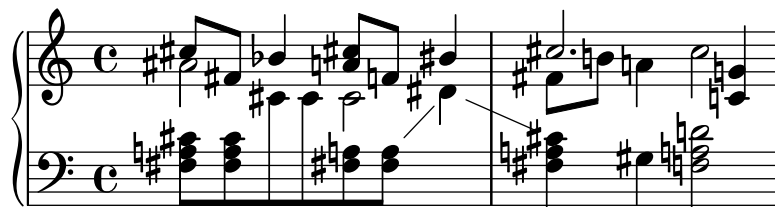
Esta regla refleja una práctica introducida por los compositores de principios del s.XX, en un intento de abolir la jerarquía entre notas naturales y alteradas. Con este estilo, *todas* las notas llevan alteración, incluso becuadros.



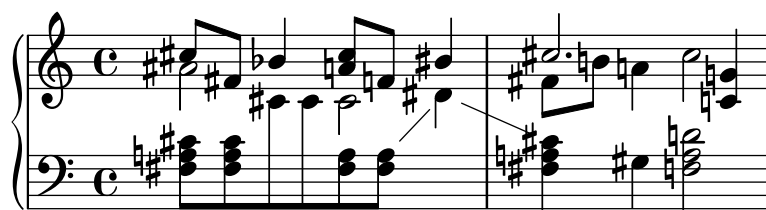


**dodecaphonic-no-repeat**

Como con el estilo de alteraciones dodecafónico *todas* las notas llevan una alteración de forma predeterminada, pero las alteraciones se suprimen cuando hay notas repetidas inmediatamente en el mismo pentagrama.

**dodecaphonic-first**

Similar al estilo de alteraciones dodecafónico, *todas* las notas llevan una alteración, pero solo la primera vez que se encuentran en el compás. Las alteraciones se recuerdan solamente para la octava actual pero a través de las voces.

**teaching (enseñanza)**

Esta regla está pensada para estudiantes, y hace más sencillo crear hojas de escalas con alteraciones de precaución creadas automáticamente. Las alteraciones se imprimen como en el estilo **modern**, pero se añaden alteraciones de precaución para todas las notas sostenidas o bemoles especificadas por la armadura, excepto si la nota se repite inmediatamente.

**no-reset (no restablecer)**

Es el mismo que **default** pero con alteraciones que duran 'para siempre' y no sólo dentro del mismo compás:



**forget** (olvidar)

Es lo opuesto a **no-reset**: Las alteraciones no se recuerdan en absoluto: de aquí que todas las alteraciones se compongan tipográficamente en relación a la armadura de la tonalidad, sin que estén afectadas por lo que viene antes.



## Véase también

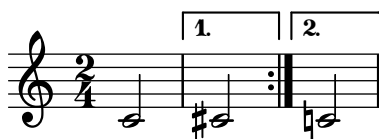
Fragmentos de código: Sección “Pitches” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Accidental” en *Referencia de Funcionamiento Interno*, Sección “Accidental-engraver” en *Referencia de Funcionamiento Interno*, Sección “GrandStaff” en *Referencia de Funcionamiento Interno*, Sección “PianoStaff” en *Referencia de Funcionamiento Interno*, Sección “Staff” en *Referencia de Funcionamiento Interno*, Sección “AccidentalSuggestion” en *Referencia de Funcionamiento Interno*, Sección “AccidentalPlacement” en *Referencia de Funcionamiento Interno*, Sección “accidental-suggestion-interface” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Las notas simultáneas no se tienen en cuenta para la determinación automática de las alteraciones accidentales; sólo se consideran las notas anteriores y la armadura de la tonalidad. Puede ser necesario forzar las alteraciones accidentales con **with !** o **?** cuando la misma nota, con el mismo nombre, ocurre simultáneamente con distintas alteraciones, como en ‘<f! fis!>’.

La cancelación de precaución de alteraciones se hace mirando el compás previo. Sin embargo, en el bloque **\alternative** que sigue a una sección de repetición de primera y segunda vez **\repeat volta N**, se esperaría que la cancelación se calculase utilizando el compás previo *que se ha tocado*, no el compás previo *que se ha impreso*. En el ejemplo siguiente, el Do natural de la casilla de segunda vez no necesita becuadro:



Se puede usar el siguiente rodeo del problema: definir una función que cambie localmente el estilo de alteraciones a **forget**:

```
forget = #(define-music-function (music) (ly:music?) #{
  \accidentalStyle forget
  #music
  \accidentalStyle modern
#})
{
  \accidentalStyle modern
  \time 2/4
  \repeat volta 2 {
    c'2
  }
}
```

```

\alternative {
  cis'
  \forget c'
}

```



## Tesitura

El término *ambitus* o ámbito, denota el rango de notas que abarca una voz dada en una parte musical. También puede denotar el margen de notas que es capaz de tocar un determinado instrumento musical. Los ámbitos se imprimen en las partes vocales de tal manera que los intérpretes puedan determinar con facilidad si cumplen con sus propias posibilidades.

Los ámbitos se presentan al comienzo de la pieza junto a la clave inicial. El rango se especifica gráficamente mediante dos cabezas de nota que representan a las notas inferior y superior. Sólo se imprimen alteraciones si no forman parte de la armadura de la tonalidad.

```

\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

\relative {
  aes' c e2
  cis,1
}

```



## Fragmentos de código seleccionados

### Añadir un ámbito por voz

Se puede añadir un ámbito por cada voz. En este caso, el ámbito se debe desplazar manualmente para evitar colisiones.

```

\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus.X-offset = #2.0
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  }

```

```

    } \relative c' {
      \voiceTwo
      es4 f g as
      b1
    }
  >>

```



### *Ámbitos con varias voces*

La adición del grabador `Ambitus_engraver` al contexto de `Staff` crea un solo ámbito por pentagrama, incluso en el caso de pentagramas con varias voces.

```

\new Staff \with {
  \consists "Ambitus_engraver"
}
<<
  \new Voice \relative c'' {
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
}
>>

```



### *Modificación de la separación en las indicaciones de tesitura*

Es posible ajustar la separación predeterminada entre las notas de la indicación de tesitura y la línea que las une.

```

\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

\new Staff {
  \time 2/4
  % Default setting
  c'4 g''
}

```

```

\new Staff {
  \time 2/4
  \override AmbitusLine.gap = #0
  c'4 g' '
}

\new Staff {
  \time 2/4
  \override AmbitusLine.gap = #1
  c'4 g' '
}

\new Staff {
  \time 2/4
  \override AmbitusLine.gap = #1.5
  c'4 g' '
}

```



## Véase también

Glosario musical: Sección “ambitus” en *Glosario Musical*.

Fragmentos de código: Sección “Pitches” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Ambitus\_engraver” en *Referencia de Funcionamiento Interno*, Sección “Voice” en *Referencia de Funcionamiento Interno*, Sección “Staff” en *Referencia de Funcionamiento Interno*, Sección “Ambitus” en *Referencia de Funcionamiento Interno*, Sección “AmbitusAccidental” en *Referencia de Funcionamiento Interno*, Sección “AmbitusLine” en *Referencia de Funcionamiento Interno*, Sección “AmbitusNoteHead” en *Referencia de Funcionamiento Interno*, Sección “ambitus-interface” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

No se efectúa un tratamiento de las posibles colisiones en caso de varias indicaciones de ámbito simultáneas en distintas voces.

### 1.1.4 Cabeza de las notas

Esta sección propone formas de alterar las cabezas de las figuras.

## Cabezas de nota especiales

Se puede modificar la apariencia de la cabeza de las notas:

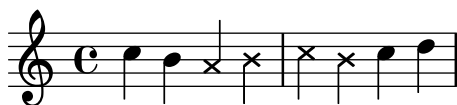
```
\relative c'' {
  c4 b
  \override NoteHead.style = #'cross
  c4 b
  \revert NoteHead.style
  a b
  \override NoteHead.style = #'harmonic
  a b
  \revert NoteHead.style
  c4 d e f
}
```



Para ver todos los estilos de cabeza de las notas, consulte Sección A.9 [Estilos de cabezas de nota], página 734.

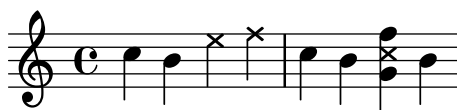
El estilo **cross** (aspas) se usa para representar una amplia variedad de intenciones musicales. Las siguientes instrucciones genéricas predefinidas modifican la forma de la cabeza de las figuras tanto en el contexto de pentagrama normal como en el de tablatura, y se pueden usar para representar cualquier significado musical:

```
\relative {
  c''4 b
  \xNotesOn
  a b c4 b
  \xNotesOff
  c4 d
}
```



La forma de función musical de esta instrucción predefinida se puede usar dentro y fuera de los acordes para generar cabezas de nota en aspa, tanto en el contexto de pentagrama normal como en el de tablatura:

```
\relative {
  c''4 b
  \xNote { e f }
  c b < g \xNote c f > b
}
```



Como sinónimos de `\xNote`, `\xNotesOn` y `\xNotesOff`, se pueden usar `\deadNote`, `\deadNotesOn` y `\deadNotesOff`. El término *dead note* (nota muerta) se utiliza corrientemente por parte de los guitarristas.

También existe una abreviatura similar para las formas en rombo:

```
\relative c'' {
  <c f\harmonic>2 <d a'\harmonic>4 <c g'\harmonic> f\harmonic
}
```



## Instrucciones predefinidas

`\harmonic`, `\xNotesOn`, `\xNotesOff`, `\xNote`.

## Véase también

Fragmentos de código: Sección “Pitches” en *Fragmentos de código*.

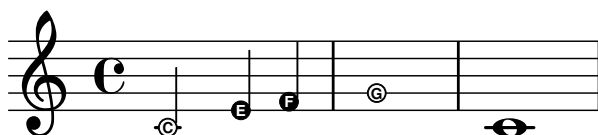
Referencia de la notación: Sección A.9 [Estilos de cabezas de nota], página 734, [Notas en acorde], página 176, [Indicar armónicos y notas tapadas], página 416.

Referencia de funcionamiento interno: Sección “note-event” en *Referencia de Funcionamiento Interno*, Sección “Note\_heads-engraver” en *Referencia de Funcionamiento Interno*, Sección “Ledger\_line-engraver” en *Referencia de Funcionamiento Interno*, Sección “NoteHead” en *Referencia de Funcionamiento Interno*, Sección “LedgerLineSpanner” en *Referencia de Funcionamiento Interno*, Sección “note-head-interface” en *Referencia de Funcionamiento Interno*, Sección “ledger-line-spanner-interface” en *Referencia de Funcionamiento Interno*.

## Cabezas de notas de Notación Fácil

Las notas con cabeza de ‘notación facilitada’ tienen el nombre de la nota (en inglés) dentro de la cabeza. Se usan en la música para principiantes. Para que las letras sean legibles, se deben imprimir en un tamaño grande de fuente tipográfica. Para imprimir con una fuente más grande, véase Sección 4.2.2 [Establecer el tamaño del pentagrama], página 581.

```
 #(set-global-staff-size 26)
 \relative c' {
   \easyHeadsOn
   c2 e4 f
   g1
   \easyHeadsOff
   c,1
 }
```



## Instrucciones predefinidas

`\easyHeadsOn`, `\easyHeadsOff`.

## Fragmentos de código seleccionados

### *Números como notas de notación fácil*

Las cabezas de nota de notación fácil utilizan la propiedad `note-names` del objeto `NoteHead` para determinar lo que aparece dentro de la cabeza. Mediante la sobreescritura de esta propiedad, es posible imprimir números que representen el grado de la escala.

Se puede crear un grabador simple que haga esto para la cabeza de cada nota que ve.

```
#(define Ez_numbers_engraver
  (make-engraver
    (acknowledgers
      ((note-head-interface engraver grob source-engraver)
        (let* ((context (ly:translator-context engraver))
              (tonic-pitch (ly:context-property context 'tonic))
              (tonic-name (ly:pitch-notename tonic-pitch))
              (grob-pitch
                (ly:event-property (event-cause grob) 'pitch))
              (grob-name (ly:pitch-notename grob-pitch))
              (delta (modulo (- grob-name tonic-name) 7))
              (note-names
                (make-vector 7 (number->string (1+ delta))))))
          (ly:grob-set-property! grob 'note-names note-names))))))

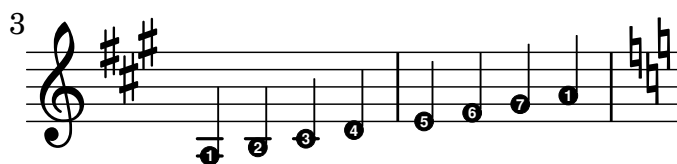
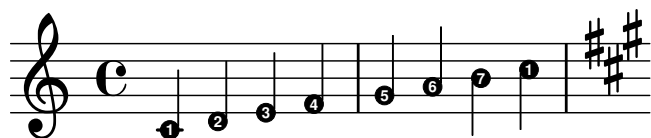
#(set-global-staff-size 26)

\layout {
  ragged-right = ##t
  \context {
    \Voice
    \consists \Ez_numbers_engraver
  }
}

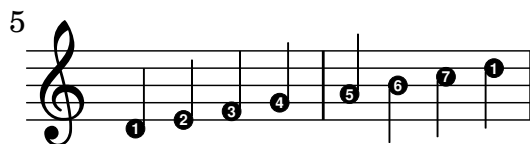
\relative c' {
  \easyHeadsOn
  c4 d e f
  g4 a b c \break

  \key a \major
  a,4 b cis d
  e4 fis gis a \break

  \key d \dorian
  d,4 e f g
  a4 b c d
}
```







## Véase también

Referencia de la notación: Sección 4.2.2 [Establecer el tamaño del pentagrama], página 581.

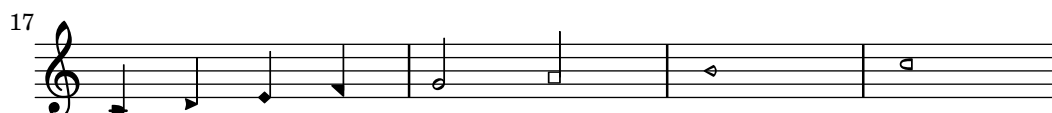
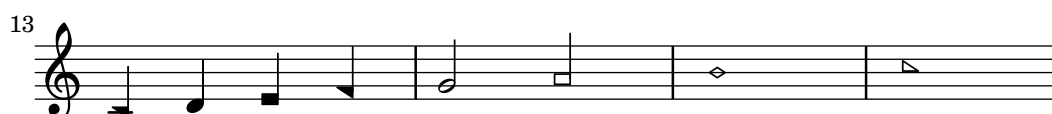
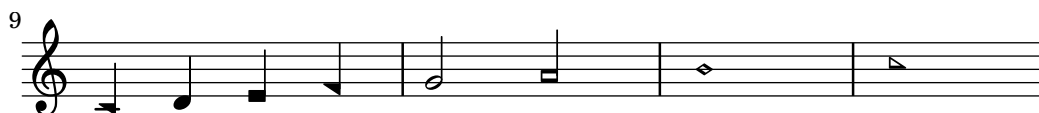
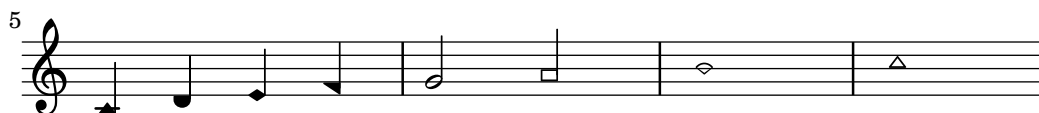
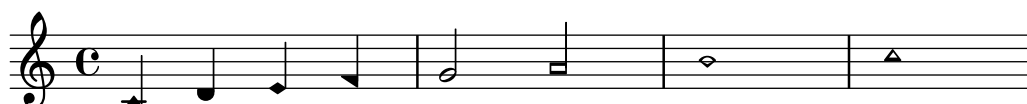
Fragmentos de código: Sección “Pitches” en *Fragmentos de código*.

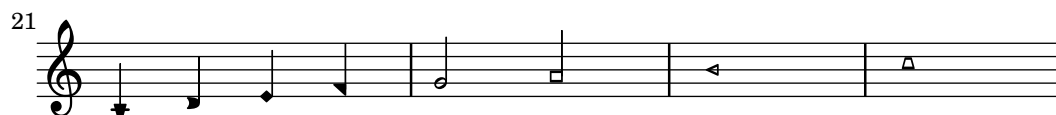
Referencia de funcionamiento interno: Sección “note-event” en *Referencia de Funcionamiento Interno*, Sección “Note\_heads\_engraver” en *Referencia de Funcionamiento Interno*, Sección “NoteHead” en *Referencia de Funcionamiento Interno*, Sección “note-head-interface” en *Referencia de Funcionamiento Interno*.

## Cabezas de notas con formas diversas

En la notación de cabezas con forma, la forma de la cabeza corresponde a la función armónica de una nota dentro de la escala. Esta notación se hizo popular en los libros americanos de canciones durante el s.XIX. Las cabezas de nota con formas se pueden producir en los estilos ‘Sacred Harp’, ‘Southern Harmony’, Funk (Harmonia Sacra), Walker y Aiken ‘(Christian Harmony)’:

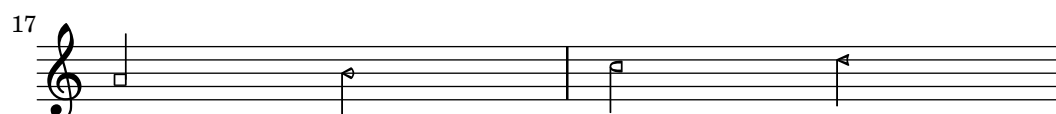
```
\relative c'' {
  \aikenHeads
  c, d e f g2 a b1 c \break
  \aikenThinHeads
  c,4 d e f g2 a b1 c \break
  \sacredHarpHeads
  c,4 d e f g2 a b1 c \break
  \southernHarmonyHeads
  c,4 d e f g2 a b1 c \break
  \funkHeads
  c,4 d e f g2 a b1 c \break
  \walkerHeads
  c,4 d e f g2 a b1 c \break
}
```





Las formas se determinan en función del grado de la escala, donde la tónica está determinada por la instrucción `\key`. Cuando se escribe en un tono menor, la nota de la escala se puede determinar a partir del relativo mayor:

```
\relative c'' {
  \key a \minor
  \aikenHeads
  a b c d e2 f g1 a \break
  \aikenHeadsMinor
  a,4 b c d e2 f g1 a \break
  \aikenThinHeadsMinor
  a,4 b c d e2 f g1 a \break
  \sacredHarpHeadsMinor
  a,2 b c d \break
  \southernHarmonyHeadsMinor
  a2 b c d \break
  \funkHeadsMinor
  a2 b c d \break
  \walkerHeadsMinor
  a2 b c d \break
}
```



## Instrucciones predefinidas

`\aikenHeads`, `\aikenHeadsMinor`, `\aikenThinHeads`, `\aikenThinHeadsMinor`, `\funkHeads`, `\funkHeadsMinor`, `\sacredHarpHeads`, `\sacredHarpHeadsMinor`, `\southernHarmonyHeads`, `\southernHarmonyHeadsMinor`, `\walkerHeads`, `\walkerHeadsMinor`.

## Fragmentos de código seleccionados

### *Aiken head thin variant noteheads*

Aiken head white notes get harder to read at smaller staff sizes, especially with ledger lines. Losing interior white space makes them appear as quarter notes.

```
\score {
  {
    \aikenHeads
    c''2 a' c' a

    % Switch to thin-variant noteheads
    \set shapeNoteStyles = ##(doThin reThin miThin
                           faThin sol laThin tiThin)
    c'' a' c' a
  }
}
% END EXAMPLE
```



### *Aplicar estilos de cabeza según la nota de la escala*

La propiedad `shapeNoteStyles` se puede usar para definir varios estilos de cabezas de nota para cada grado de la escala (según esté establecido por la armadura o por la propiedad `tonic`). Esta propiedad requiere un conjunto de símbolos, que pueden ser puramente arbitrarios (se permiten expresiones geométricas como `triangle`, triángulo, `cross`, aspas, y `xcircle`, círculo con aspas) o basados en una antigua tradición americana de grabado (ciertos nombres de nota latinos también se permiten).

Dicho esto, para imitar antiguos cancioneros americanos, existen varios estilos predefinidos de cabezas de nota disponibles a través de instrucciones de abreviatura como `\aikenHeads` o `\sacredHarpHeads`.

Este ejemplo muestra distintas formas de obtener cabezas de notas con forma, y muestra la capacidad de transportar una melodía sin perder la correspondencia entre las funciones armónicas y los estilos de cabezas de nota.

```
fragment = {
  \key c \major
  c2 d
  e2 f
  g2 a
  b2 c
}

\new Staff {
  \transpose c d
```

```

\relative c' {
  \set shapeNoteStyles = ##(do re mi fa
                        #f la ti)

  \fragment
}

\break

\relative c' {
  \set shapeNoteStyles = ##(cross triangle fa #f
                        mensural xcircle diamond)

  \fragment
}

```



Para ver todos los estilos de formas de cabeza de nota, consulte Sección A.9 [Estilos de cabezas de nota], página 734.

## Véase también

Fragmentos de código: Sección “Pitches” en *Fragmentos de código*.

Referencia de la notación: Sección A.9 [Estilos de cabezas de nota], página 734.

Referencia de funcionamiento interno: Sección “note-event” en *Referencia de Funcionamiento Interno*, Sección “Note\_heads\_engraver” en *Referencia de Funcionamiento Interno*, Sección “NoteHead” en *Referencia de Funcionamiento Interno*, Sección “note-head-interface” en *Referencia de Funcionamiento Interno*.

## Improvisación

La improvisación se denota a veces mediante cabezas de nota en forma de barra inclinada, donde el ejecutante puede elegir cualquier nota pero con el ritmo especificado. Estas cabezas de nota se crean así:

```

\new Voice \with {
  \consists "Pitch_squash_engraver"
} \relative {
  e''8 e g a a16( bes) a8 g
  \improvisationOn
  e8 ~
  2 ~ 8 f4 f8 ~
  2
  \improvisationOff
  a16( bes) a8 g e

```

}



## Instrucciones predefinidas

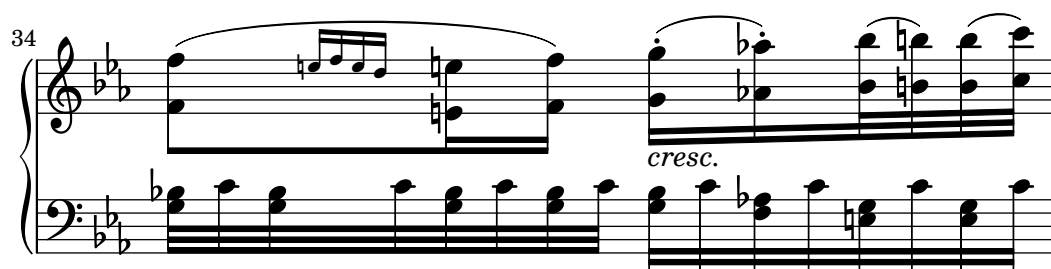
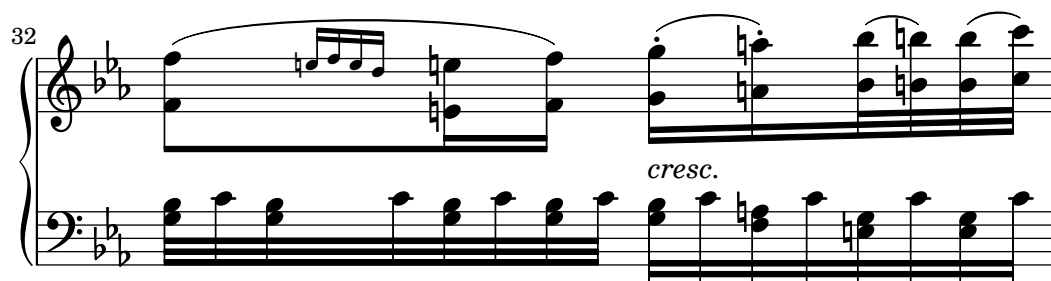
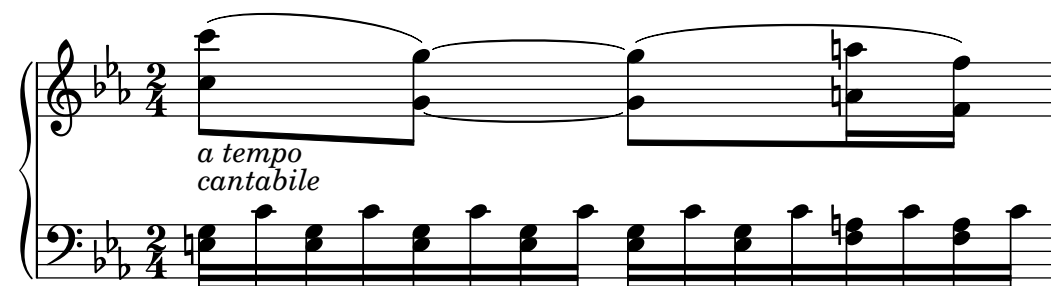
`\improvisationOn, \improvisationOff.`

## Véase también

Fragmentos de código: Sección “Pitches” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Pitch\_squash\_engraver” en *Referencia de Funcionamiento Interno*, Sección “Voice” en *Referencia de Funcionamiento Interno*, Sección “RhythmicStaff” en *Referencia de Funcionamiento Interno*.

## 1.2 Duraciones



Esta sección trata de los ritmos, los silencios, las duraciones, las barras y los compases.

### 1.2.1 Escritura de las duraciones (valores rítmicos)

#### Duración de las notas

La duración de las notas se introduce mediante números y puntos. El número que se escribe está basado en el valor recíproco de la longitud de la nota. Respecto a la redonda, esto es el número de veces que dicha nota cabe en una redonda. Por ejemplo, una negra se escribe usando un 4 (puesto que es 1/4 de redonda), mientras que una blanca se escribe con un 2 (por ser 1/2 de redonda), una corchea se escribe como 8 y así sucesivamente. Se pueden especificar duraciones tan cortas como 1/1024 de redonda (sin nombre en español) pero cualquier valor más corto, aunque sea posible, solo se puede introducir como grupos de notas unidas por una barra. Véase también Sección 1.2.4 [Barras], página 88.

Para notas mayores de la redonda se deben usar las instrucciones `\longa` (que es una breve doble) y `\breve`. Se puede escribir una nota con la duración de una cuádruple breve mediante `\maxima`, pero esto está contemplado solamente dentro de la notación musical antigua. Para ver más detalles, consulte Sección 2.9 [Notación antigua], página 463.

```
\relative {
  \time 8/1
  c''\longa c\breve c1 c2
  c4 c8 c16 c32 c64 c128 c128
}
```



Aquí se pueden ver las mismas duraciones con el barrado automático desactivado.

```
\relative {
  \time 8/1
  \autoBeamOff
  c''\longa c\breve c1 c2
  c4 c8 c16 c32 c64 c128 c128
}
```



Las duraciones aisladas (aquellas que no especifican ninguna altura) en una secuencia musical toman la altura de la nota o acorde anterior.

```
\relative {
  \time 8/1
  c'' \longa \breve 1 2
  4 8 16 32 64 128 128
}
```



Las alturas aisladas (notas sin ninguna duración especificada) que aparecen dentro de una secuencia musical, obtienen su duración de la nota o acorde anterior. Si no hay ninguna duración anterior, entonces el valor predeterminado que se utiliza para la nota siempre es 4, una negra.

```
\relative { a' a a2 a a4 a a1 a }
```



Para obtener duraciones de notas con puntillo, escriba un punto (.) después del número. Las notas con doble puntillo se especifican escribiendo dos puntos, y así sucesivamente.

```
\relative { a'4 b c4. b8 a4. b4.. c8. }
```



Para evitar la superposición con las líneas del pentagrama, los puntillos se desplazan normalmente hacia arriba. Sin embargo, en situaciones de polifonía, se pueden colocar manualmente por encima o por debajo según sea necesario. Véase Sección 5.4.2 [Dirección y posición], página 661.

Ciertas duraciones no se pueden representar solamente mediante números y puntillos; puede ser necesario emplear dos o más notas ligadas entre sí. Véase [Ligaduras de unión], página 58.

Para especificar duraciones que puedan alinear las notas con las sílabas de la letra, véase Sección 2.1 [Música vocal], página 286.

Las notas también se pueden espaciar proporcionalmente a su duración, véase Sección 4.5.5 [Notación proporcional], página 611.

## Instrucciones predefinidas

```
\autoBeamOn, \autoBeamOff, \dotsUp, \dotsDown, \dotsNeutral.
```

## Fragmentos de código seleccionados

### *Formas alternativas de la figura breve*

Las figuras de breve también están disponibles con dos líneas verticales a los lados de la cabeza en lugar de una sola línea y la forma en estilo barroco.

```
\relative c'' {
  \time 4/2
  c\breve |
  \override Staff.NoteHead.style = #'altdefault
  b\breve
  \override Staff.NoteHead.style = #'baroque
  b\breve
  \revert Staff.NoteHead.style
  a\breve
}
```



## Modificar el número de puntillos de una nota

La cantidad de puntillos de una nota se puede modificar independientemente de los puntillos que se escriben después de la nota.

```
\relative c' {
  c4.. a16 r2 |
  \override Dots.dot-count = #4
  c4.. a16 r2 |
  \override Dots.dot-count = #0
  c4.. a16 r2 |
  \revert Dots.dot-count
  c4.. a16 r2 |
}
```



## Véase también

Glosario musical: Sección “breve” en *Glosario Musical*, Sección “longa” en *Glosario Musical*, Sección “maxima” en *Glosario Musical*, Sección “note value” en *Glosario Musical*, Sección “Duration names notes and rests” en *Glosario Musical*.

Referencia de la notación: Sección 1.2.4 [Barras], página 88, [Ligaduras de unión], página 58, [Plicas], página 245, Sección 1.2.1 [Escritura de las duraciones (valores rítmicos)], página 49, Sección 1.2.2 [Escritura de silencios], página 62, Sección 2.1 [Música vocal], página 286, Sección 2.9 [Notación antigua], página 463, Sección 4.5.5 [Notación proporcional], página 611.

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Dots” en *Referencia de Funcionamiento Interno*, Sección “DotColumn” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Aunque no existe un límite fundamental para las duraciones de los silencios (tanto para el más largo como para el más corto), pero el número de glifos es limitado: sólo se pueden imprimir desde el silencio de 1/1024 de redonda (sin nombre en español) hasta el de `\maxima` (8 redondas).

## Grupos especiales

Los grupos especiales se obtienen a partir de una expresión musical con la instrucción `\tuplet`, multiplicando la rapidez de la expresión musical por una fracción:

```
\tuplet fracción { música }
```

El numerador de la fracción se imprime encima o debajo de las notas, opcionalmente con un corchete. El grupo especial más común es el tresillo, en el que 3 notas tienen el valor que normalmente tienen 2:

```
\relative {
  a'2 \tuplet 3/2 { b4 4 4 }
  c4 c \tuplet 3/2 { b4 a g }
}
```





Si se están escribiendo pasajes con muchos grupos de valoración especial, resulta fastidioso tener que escribir una instrucción `\tuplet` distinta para cada grupo. Es posible especificar la duración de un conjunto de grupos directamente antes de la música para que así se agrupen automáticamente:

```
\relative {
  g'2 r8 \tuplet 3/2 8 { cis16 d e e f g g f e }
}
```



Se pueden colocar manualmente los corchetes de tresillo encima o debajo de la pauta:

```
\relative {
  \tupletUp \tuplet 3/2 { c''8 d e }
  \tupletNeutral \tuplet 3/2 { c8 d e }
  \tupletDown \tuplet 3/2 { f,8 g a }
  \tupletNeutral \tuplet 3/2 { f8 g a }
}
```



Los grupos pueden anidarse unos dentro de otros:

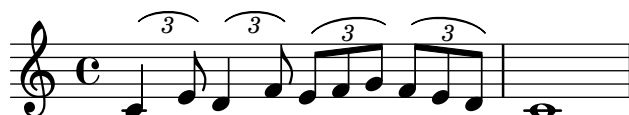
```
\relative {
  \autoBeamOff
  c''4 \tuplet 5/4 { f8 e f \tuplet 3/2 { e[ f g] } } f4
}
```



La modificación de los grupos especiales anidados que comienzan en el mismo instante musical se debe hacer con `\tweak`; véase Sección 5.3.4 [La instrucción `\tweak`], página 650.

Se puede sustituir el corchete recto de un grupo de valoración especial con una ligadura curva, a la manera de muchas ediciones antiguas:

```
\relative {
  \tuplet 3/2 4 {
    \override TupletBracket.tuplet-slur = ##t
    c'4 e8 d4 f8
    \override TupletBracket.bracket-visibility = ##t
    e f g f e d
  } c1
}
```



De forma predeterminada, solo se imprime el corchete si no están unidas por una barra todas las notas que comprende el grupo; en ciertos casos (por ejemplo con las ligaduras, como en el ejemplo de arriba) podría ser preferible modificar este comportamiento, a través de la propiedad `bracket-visibility` tal y como se detalla en uno de los siguientes fragmentos de código.

De manera más general se pueden mostrar u ocultar los objetos `TupletBracket` o `TupletNumber`, o ambos, como se explica en Sección 5.4.7 [Visibilidad de los objetos], página 670; sin embargo, una forma más flexible de modificar la duración de las notas sin imprimir un grupo especial (ni un corchete) se explica también en [Escalar las duraciones], página 56.

## Instrucciones predefinidas

`\tupletUp`, `\tupletDown`, `\tupletNeutral`.

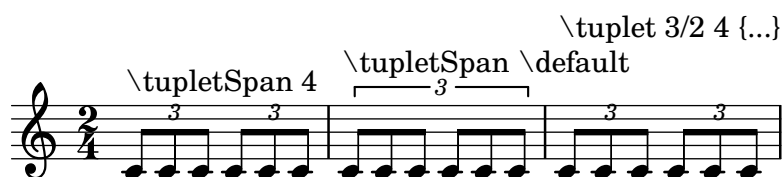
## Fragmentos de código seleccionados

### *Entering several tuplets using only one `\tuplet` command*

The property `tupletSpannerDuration` sets how long each of the tuplets contained within the brackets after `\tuplet` should last. Many consecutive tuplets can then be placed within a single `\tuplet` expression, thus saving typing.

There are several ways to set `tupletSpannerDuration`. The command `\tupletSpan` sets it to a given duration, and clears it when instead of a duration `\default` is specified. Another way is to use an optional argument with `\tuplet`.

```
\relative c' {
  \time 2/4
  \tupletSpan 4
  \tuplet 3/2 { c8^"\tupletSpan 4" c c c c c }
  \tupletSpan \default
  \tuplet 3/2 { c8^"\tupletSpan \default" c c c c c }
  \tuplet 3/2 4 { c8^"\tuplet 3/2 4 {...}" c c c c c }
}
```



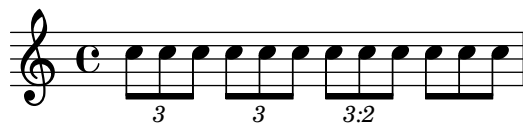
### *Cambiar el número del grupo especial*

De forma predeterminada sólo se imprime sobre el corchete de grupo el numerador del grupo especial, o sea, el numerador del argumento de la instrucción `\tuplet`.

De forma alternativa, se puede imprimir un quebrado en la forma numerador:denominador del número del grupo, o eliminar el número.

```
\relative c'' {
  \tuplet 3/2 { c8 c c }
  \tuplet 3/2 { c8 c c }
  \override TupletNumber.text = #tuplet-number::calc-fraction-text
  \tuplet 3/2 { c8 c c }
  \omit TupletNumber
  \tuplet 3/2 { c8 c c }
```

}



### *Números de agrupación especial distintos a los predeterminados*

LilyPond también proporciona funciones de formato para imprimir números de grupo especial diferentes a la propia fracción, así como para añadir una figura al número o a la fracción de la agrupación.

```
\relative c'' {
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-tuplet-denominator-text 7)
  \tuplet 3/2 { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-tuplet-fraction-text 12 7)
  \tuplet 3/2 { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      (tuplet-number::non-default-tuplet-fraction-text 12 7)
      (ly:make-duration 3 0))
  \tuplet 3/2 { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      tuplet-number::calc-denominator-text
      (ly:make-duration 2 0))
  \tuplet 3/2 { c8 c8 c8 c8 c8 c8 }
  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      tuplet-number::calc-fraction-text
      (ly:make-duration 2 0))
  \tuplet 3/2 { c8 c8 c8 c8 c8 c8 }
  \once \override TupletNumber.text =
    #(tuplet-number::fraction-with-notes
      (ly:make-duration 2 1) (ly:make-duration 3 0))
  \tuplet 3/2 { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-fraction-with-notes 12
      (ly:make-duration 3 0) 4 (ly:make-duration 2 0))
  \tuplet 3/2 { c4. c4. c4. c4. }
}
```



### *Controlar la visibilidad de los corchetes de grupo especial*

El comportamiento predeterminado de la visibilidad de los corchetes de grupo de valoración especial es imprimir el corchete a no ser que haya una barra de la misma longitud que el grupo especial. Para controlar la visibilidad de los corchetes de grupo, establezca la propiedad `'bracket-visibility` a `#t` (imprimir el corchete siempre), `#'if-no-beam` (imprimir el corchete solamente si no hay barra, el comportamiento predeterminado), o `#f` (no imprimir nunca el corchete). Este último equivale de hecho a omitir el objeto `TupletBracket` de la salida impresa.

```
music = \relative c' {
  \tuplet 3/2 { c16[ d e ] f8]
  \tuplet 3/2 { c8 d e }
  \tuplet 3/2 { c4 d e }
}

\new Voice {
  \relative c' {
    << \music s4^"default" >>
    \override TupletBracket.bracket-visibility = #'if-no-beam
    << \music s4^"'if-no-beam" >>
    \override TupletBracket.bracket-visibility = ##t
    << \music s4^"#t" >>
    \override TupletBracket.bracket-visibility = ##f
    << \music s4^"#f" >>
    \omit TupletBracket
    << \music s4^"omit" >>
  }
}
```



### *Permitir saltos de línea dentro de grupos especiales con barra*

Este ejemplo artificial muestra cómo se pueden permitir tanto los saltos de línea manuales como los automáticos dentro de un grupo de valoración especial unido por una barra. Observe que estos grupos sincopados se deben barrar manualmente.

```
\layout {
  \context {
    \Voice
    % Permit line breaks within triplets
    \remove "Forbid_line_break_engraver"
    % Allow beams to be broken at line breaks
    \override Beam.breakable = ##t
  }
}
\relative c'' {
  a8
  \repeat unfold 5 { \tuplet 3/2 { c[ b a] } }
  % Insert a manual line break within a triplet
  \tuplet 3/2 { c[ b \bar "" \break a] }
  \repeat unfold 5 { \tuplet 3/2 { c[ b a] } }
  c8
}
```



### Véase también

Glosario musical: Sección “triplet” en *Glosario Musical*, Sección “tuplet” en *Glosario Musical*, Sección “polymetric” en *Glosario Musical*.

Manual de aprendizaje: Sección “Métodos de trucaje” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 661, Sección 5.4.7 [Visibilidad de los objetos], página 670, [Gestión del tiempo], página 126, [Escalar las duraciones], página 56, Sección 5.3.4 [La instrucción `\tweak`], página 650, [Notación polimétrica], página 81.

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TupletBracket” en *Referencia de Funcionamiento Interno*, Sección “TupletNumber” en *Referencia de Funcionamiento Interno*, Sección “TimeScaledMusic” en *Referencia de Funcionamiento Interno*.

### Escalar las duraciones

La duración de las figuras, silencios o acordes se puede multiplicar por un factor N/M añadiendo `*N/M` (o `*N` si M es 1) a la duración. También se pueden añadir factores usando expresiones de Scheme que se evalúan a un número o a una duración musical como `*#(ly:music-length music)`. Esto es muy útil para escalar una duración de ‘1’ y que una nota o silencio multicompa se estire hasta una longitud derivada de una variable musical.

La adición de un factor no afectará a la apariencia de las notas o silencios que se producen, pero la duración alterada se usará para calcular la posición dentro del compás y para establecer la duración en la salida MIDI. Los factores de multiplicación se pueden combinar en la forma  $*L*M/N$ . Los factores son parte de la duración: si no especificamos una duración para las notas siguientes, la duración por omisión que se toma de la nota anterior incluirá cualquier factor de escala que se haya aplicado.

En el siguiente ejemplo las tres primeras notas duran exactamente dos partes, pero no se imprime ningún corchete de tresillo.

```
\relative {
  \time 2/4
  % Alter durations to triplets
  a'4*2/3 gis a
  % Normal durations
  a4 a
  % Double the duration of chord
  <a d>4*2
  % Duration of quarter, appears like sixteenth
  b16*4 c4
}
```



La duración de los silencios espaciadores también se puede modificar mediante un multiplicador. Esto es útil para saltar muchos compases, como por ejemplo `s1*23`.

De la misma forma, se pueden comprimir por una fracción trozos de música más largos, como si cada nota, acorde o silencio tuviera la fracción como multiplicador. Esto dejará intacta la apariencia de la *música*, pero la duración interna de las notas se multiplicará por el factor de escala dado, habitualmente *numerador/denominador*. He aquí un ejemplo que muestra cómo se puede comprimir y expandir la música:

```
\relative {
  \time 2/4
  % Normal durations
  <c' a>4 c8 a
  % Scale music by *2/3
  \scaleDurations 2/3 {
    <c a f>4. c8 a f
  }
  % Scale music by *2
  \scaleDurations 2 {
    <c' a>4 c8 b
  }
}
```



Una aplicación de esta instrucción se encuentra en la notación polimétrica, véase [Notación polimétrica], página 81.

## Véase también

Referencia de la notación: [Grupos especiales], página 51, [Silencios invisibles], página 64, [Notación polimétrica], página 81.

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

## Advertencias y problemas conocidos

El cálculo de la posición dentro de un compás debe tener en cuenta todos los factores de escalado que se han aplicado a las notas dentro de ese compás, y cualquier acarreo fraccionado desde los compases anteriores. Este cálculo se efectúa utilizando números racionales. Si un numerador o denominador intermedio en dicho proceso de cálculo sobrepasa la cantidad de  $2^{30}$ , la ejecución y el tipografiado se detendrán en ese punto sin indicar ningún error.

## Ligaduras de unión

Una ligadura de unión conecta dos notas adyacentes de la misma altura. La ligadura en efecto extiende la longitud de una nota.

**Nota:** No deben confundirse las ligaduras de unión con las *ligaduras de expresión*, que indican articulación, ni con las *ligaduras de fraseo*, que indican el fraseo musical. Una ligadura de unión es tan sólo una manera de extender la duración de una nota, algo parecido a lo que hace el puntillo.

Se introduce una ligadura de unión escribiendo el símbolo de la tilde curva (~) después de la primera de cada pareja de notas que se quieren unir. Esto indica que la nota se une a la siguiente, que debe tener la misma altura.

```
{ a'2~ 4~ 16 r r8 }
```



Las ligaduras de unión pueden hacer uso de la interpretación ‘última altura explícita’ para las duraciones aisladas:

```
{ a'2~ 4~ 16 r r8 }
```



Se usan ligaduras de unión bien cuando la nota atraviesa la barra de compás o bien cuando no se pueden usar puntillos para denotar el ritmo. También se deben usar ligaduras cuando las notas atraviesan subdivisiones del compás de mayor duración:

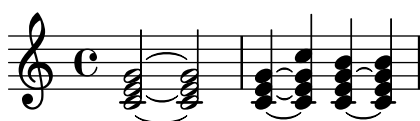
```
\relative {
  r8~"sí" c'4.~ 4 r4 |
  r8~"no" c2~ 8 r4
}
```



Si necesitamos ligar muchas notas a través de las líneas divisorias, nos podría resultar más fácil utilizar la división automática de las notas, véase [División automática de las notas], página 84. Este mecanismo divide automáticamente las notas largas y las liga a través de las barras de compás.

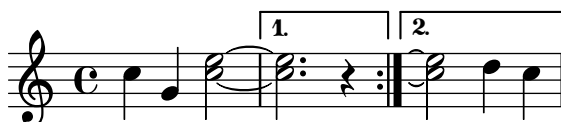
Cuando se aplica una ligadura de unión a un acorde, se conectan todas las cabezas de las notas cuyas alturas coinciden. Si no coincide ningún par de cabezas, no se crea ninguna ligadura. Los acordes se pueden ligar parcialmente colocando las ligaduras dentro del acorde.

```
\relative c' {
  <c e g>2~ 2 |
  <c e g>4~ <c e g c>
  <c~ e g~ b> <c e g b> |
}
```



Cuando la segunda vez de una repetición comienza con una nota ligada, es necesario especificar la ligadura repetida como sigue:

```
\relative {
  \repeat volta 2 { c' g <c e>2~ }
  \alternative {
    % First alternative: following note is tied normally
    { <c e>2. r4 }
    % Second alternative: following note has a repeated tie
    { <c e>2\repeatTie d4 c }
  }
}
```



Las ligaduras *L.v.* (*laissez vibrer*, dejar vibrar) indican que las notas no se deben apagar al final. Se usan en la notación para piano, arpa y otros instrumentos de cuerda y percusión. Se pueden introducir de la siguiente manera:

```
<c' f' g'>1\laissezVibrer
```



Es posible hacer manualmente que las ligaduras de unión se curven hacia arriba o hacia abajo; véase Sección 5.4.2 [Dirección y posición], página 661.

Se puede hacer que las ligaduras de expresión sean discontinuas, punteadas o una combinación de continuas y discontinuas.

```
\relative c' {
  \tieDotted
  c2~ 2
  \tieDashed
  c2~ 2
}
```



```

\tieHalfDashed
c2~ 2
\tieHalfSolid
c2~ 2
\tieSolid
c2~ 2
}

```



Se pueden especificar patrones de discontinuidad personalizados:

```

\relative c' {
  \tieDashPattern #0.3 #0.75
  c2~ 2
  \tieDashPattern #0.7 #1.5
  c2~ 2
  \tieSolid
  c2~ 2
}

```



Las definiciones de patrones de discontinuidad para las ligaduras de unión tienen la misma estructura que las definiciones de patrones de discontinuidad para las ligaduras de expresión. Para ver más información acerca de los patrones de discontinuidad complejos, consulte los fragmentos de código bajo [Ligaduras de expresión], página 140.

Sobreescriba las propiedades de disposición *whiteout* y *layer* de los objetos que puedan causar una discontinuidad en las ligaduras de unión.

```

\relative {
  \override Tie.layer = #-2
  \override Staff.TimeSignature.layer = #-1
  \override Staff.KeySignature.layer = #-1
  \override Staff.TimeSignature.whiteout = ##t
  \override Staff.KeySignature.whiteout = ##t
  b'2 b~
  \time 3/4
  \key a \major
  b r4
}

```



## Instrucciones predefinidas

`\tieUp`, `\tieDown`, `\tieNeutral`, `\tieDotted`, `\tieDashed`, `\tieDashPattern`,  
`\tieHalfDashed`, `\tieHalfSolid`, `\tieSolid`.

## Fragmentos de código seleccionados

### *Uso de ligaduras en los arpeggios*

En ocasiones se usan ligaduras de unión para escribir los arpeggios. En este caso, las dos notas ligadas no tienen que ser consecutivas. Esto se puede conseguir estableciendo la propiedad `tieWaitForNote` al valor `#t`. La misma funcionalidad es de utilidad, por ejemplo, para ligar un trémolo a un acorde, pero en principio también se puede usar para notas normales consecutivas.

```
\relative c' {
  \set tieWaitForNote = ##t
  \grace { c16[ ~ e ~ g] ~ } <c, e g>2
  \repeat tremolo 8 { c32 ~ c' ~ } <c c,>1
  e8 ~ c ~ a ~ f ~ <e' c a f>2
  \tieUp
  c8 ~ a
  \tieDown
  \tieDotted
  g8 ~ c g2
}
```



### *Grabado manual de las ligaduras*

Se pueden grabar a mano las ligaduras modificando la propiedad `tie-configuration` del objeto `TieColumn`. El primer número indica la distancia a partir de la tercera línea del pentagrama en espacios de pentagrama, y el segundo número indica la dirección (1 = hacia arriba, -1 = hacia abajo).

Observe que LilyPond distingue entre valores exactos e inexactos para el primer número. Si se está usando un valor exacto (p.ej., un entero o una fracción como  $(/ 4 5)$ ), el valor sirve como una posición vertical aproximada que después se ajusta por parte de LilyPond para que la ligadura evite las líneas del pentagrama. Si se usa un valor inexacto, como un número de coma flotante, se toma como la posición vertical sin más ajustes posteriores.

```
\relative c' {
  <c e g>2~ <c e g>
  \override TieColumn.tie-configuration =
    #'((0.0 . 1) (-2.0 . 1) (-4.0 . 1))
  <c e g>2~ <c e g>
  \override TieColumn.tie-configuration =
    #'((0 . 1) (-2 . 1) (-4 . 1))
  <c e g>2~ <c e g>
}
```



### Véase también

Glosario musical: Sección “tie” en *Glosario Musical*, Sección “laissez vibrer” en *Glosario Musical*.

Referencia de la notación: [Ligaduras de expresión], página 140, [División automática de las notas], página 84.

Fragmentos de código: Sección “slurs” en *Fragmentos de código*, Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “LaissezVibrerTie” en *Referencia de Funcionamiento Interno*, Sección “LaissezVibrerTieColumn” en *Referencia de Funcionamiento Interno*, Sección “TieColumn” en *Referencia de Funcionamiento Interno*, Sección “Tie” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Un cambio de pentagrama cuando hay una ligadura activa no producirá una ligadura inclinada.

Los cambios de clave o de octava durante una ligadura de unión no están bien definidos realmente. En estos casos puede ser preferible una ligadura de expresión.

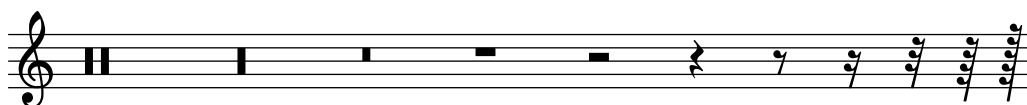
### 1.2.2 Escritura de silencios

Los silencios se escriben como parte de la música dentro de las expresiones musicales.

#### Silencios

Los silencios se introducen como notas con el nombre `r`. Las duraciones mayores que la redonda utilizan las instrucciones predefinidas que se muestran aquí:

```
\new Staff {
  % These two lines are just to prettify this example
  \time 16/1
  \omit Staff.TimeSignature
  % Print a maxima rest, equal to four breves
  r\maxima
  % Print a longa rest, equal to two breves
  r\longa
  % Print a breve rest
  r\breve
  r1 r2 r4 r8 r16 r32 r64 r128
}
```



Los silencios de un compás, centrados en medio del compás, se deben hacer con silencios multicompas. Se pueden usar para un solo compás así como para muchos compases, y se tratan en [Silencios de compás completo], página 65.

Para especificar explícitamente la posición vertical de un silencio, escriba una nota seguida de `\rest`. Se colocará un silencio en la posición en que debería aparecer la nota. Esto posibilita la aplicación manual precisa de formato a la música polifónica, ya que el formateador automático de colisiones de silencios no mueve estos silencios.

```
\relative { a'4\rest d4\rest }
```



## Fragmentos de código seleccionados

*Estilos de silencios*

Los silencios se pueden imprimir en distintos estilos.

```
\new Staff \relative c {
  \omit Score.TimeSignature
  \cadenzaOn

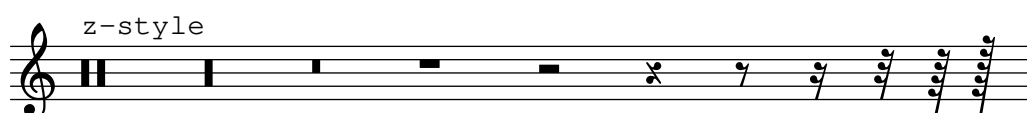
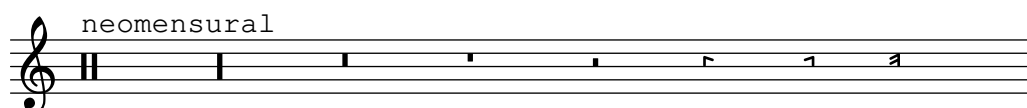
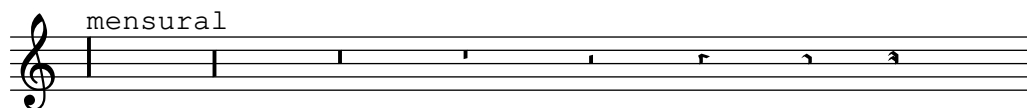
  \override Staff.Rest.style = #'mensural
  r\maxima^\markup \typewriter { mensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""
  \break

  \override Staff.Rest.style = #'neomensural
  r\maxima^\markup \typewriter { neomensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""
  \break

  \override Staff.Rest.style = #'classical
  r\maxima^\markup \typewriter { classical }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
  \bar ""
  \break

  \override Staff.Rest.style = #'z
  r\maxima^\markup \typewriter { z-style }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
  \bar ""
  \break

  \override Staff.Rest.style = #'default
  r\maxima^\markup \typewriter { default }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
}
```





## Véase también

Glosario musical: Sección “breve” en *Glosario Musical*, Sección “longa” en *Glosario Musical*, Sección “maxima” en *Glosario Musical*.

Referencia de la notación: [Silencios de compás completo], página 65.

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Rest” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

No existe un límite fundamental respecto de las duraciones de los silencios (tanto para el más corto como para el más largo), pero el número de glifos es limitado: hay silencios desde 1/1024 de redonda (sin nombre en español) hasta la máxima (8 redondas).

## Silencios invisibles

Un silencio invisible (también conocido como ‘skip’ o desplazamiento) se puede introducir como una nota con el nombre `s`:

```
\relative c'' {
  c4 c s c |
  s2 c |
}
```



Los silencios de separación sólo están disponible en el modo de notas y en el modo de acordes. En otras situaciones, por ejemplo, cuando se introduce la letra, se usa la instrucción `\skip` para producir un desplazamiento de una cierta magnitud temporal. `\skip` requiere una duración explícita, pero se ignora si las sílabas de la letra toman sus duraciones de las notas de una melodía asociada, a través de `\addlyrics` o de `\lyricsto`.

```
<<
{
  a'2 \skip2 a'2 a'2
}
\new Lyrics {
  \lyricmode {
    foo2 \skip 1 bla2
  }
}
>>
```



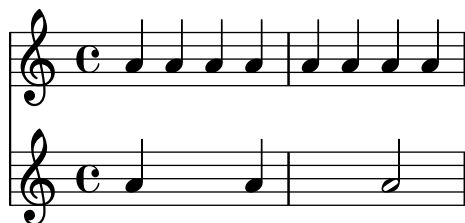
Dado que `\skip` es una instrucción, no afecta a las duraciones por omisión de las notas siguientes, a diferencia de `s`.

```
<<
```

```

{
  \repeat unfold 8 { a'4 }
}
{
  a'4 \skip 2 a' |
  s2 a'
}
>>

```



Un silencio de separación produce implícitamente contextos `Staff` y `Voice` si no existe ninguno, igual que las notas y los silencios normales:

```
{ s1 s s }
```



`\skip` tan sólo desplaza un tiempo musical; no produce ninguna salida, de ninguna clase.

```

% This is valid input, but does nothing
{ \skip 1 \skip1 \skip 1 }

```

## Véase también

Manual de aprendizaje: Sección “Visibilidad y color de los objetos” en *Manual de Aprendizaje*.

Referencia de la notación: [Notas ocultas], página 242, Sección 5.4.7 [Visibilidad de los objetos], página 670.

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “SkipMusic” en *Referencia de Funcionamiento Interno*.

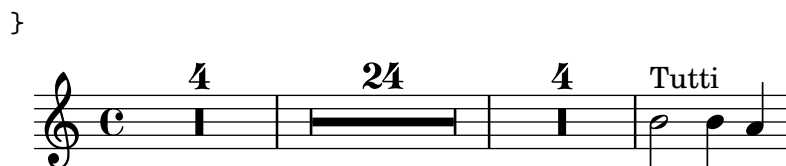
## Silencios de compás completo

Los silencios de uno o más compases completos se introducen como notas con el nombre `R` en mayúscula. Su duración se escribe de forma idéntica a la que se utiliza para las notas, incluso con la capacidad de utilizar factores multiplicadores de la duración, como se explica en [Escalar las duraciones], página 56:

```

% Rest measures contracted to single measure
\compressMMRests {
  R1*4
  R1*24
  R1*4
  b'2^"Tutti" b'4 a'4
}

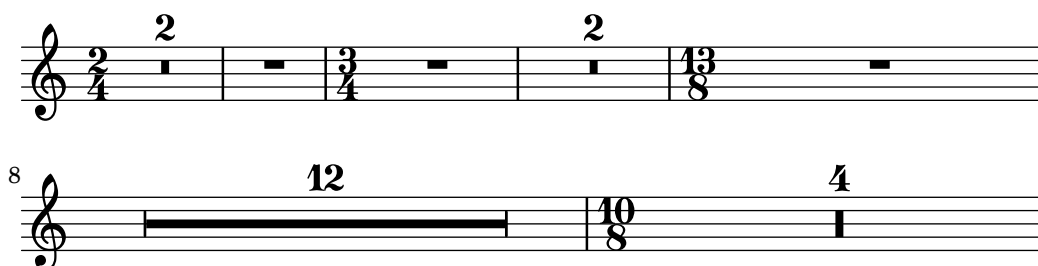
```



El ejemplo anterior también muestra cómo comprimir varios compases vacíos seguidos, como se explica en [Compresión de los compases vacíos], página 233.

La duración de un silencio multicomás siempre debe ser igual a la longitud de uno o varios compases. Por tanto, algunas indicaciones de compás requieren la utilización de puntillos o fracciones:

```
\compressMMRests {
  \time 2/4
  R1 | R2 |
  \time 3/4
  R2. | R2.*2 |
  \time 13/8
  R1*13/8 | R1*13/8*12 |
  \time 10/8
  R4*5*4 |
}
```



Un silencio de un compás completo se imprime como un silencio de redonda o de breve, centrado en el compás, según el tipo de compás vigente.

```
\time 4/4
R1 |
\time 6/4
R1*3/2 |
\time 8/4
R1*2 |
```



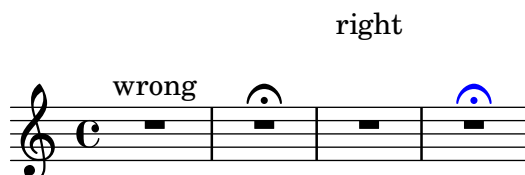
Se pueden añadir elementos de marcado a los silencios multicomás.

```
\compressMMRests {
  \time 3/4
  R2.*10~\markup { \italic "ad lib." }
}
```



**Nota:** Los elementos de marcado y articulaciones que se añaden a un silencio multicomás son objetos del tipo `MultiMeasureRestText` y `MultiMeasureRestScript`, no `TextScript` y `Script`. Las sobrescripciones de propiedades deben ir dirigidas hacia el objeto correcto, o se ignorarán. Véase el ejemplo siguiente:

```
% This fails, as the wrong object name is specified
\override TextScript.padding = #5
\override Script.color = #blue
R1~"wrong"
R1\fermata
% This is the correct object name to be specified
\override MultiMeasureRestText.padding = #5
\override MultiMeasureRestScript.color = #blue
R1~"right"
R1\fermata
```



Cuando un silencio multicomás sigue inmediatamente al establecimiento de un compás parcial con `\partial`, es posible que no se emitan las advertencias correspondientes de comprobación de compás.

## Instrucciones predefinidas

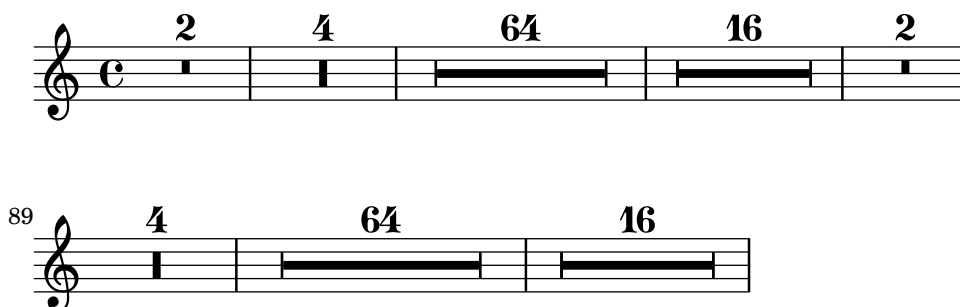
`\textLengthOn`, `\textLengthOff`, `\compressMMRests`.

## Fragmentos de código seleccionados

### *Multi-measure rest length control*

Multi-measure rests have length according to their total duration which is under the control of `MultiMeasureRest.space-increment`. Note that the default value is 2.0.

```
\relative c' {
  \compressEmptyMeasures
  R1*2 R1*4 R1*64 R1*16
  \override Staff.MultiMeasureRest.space-increment = 2.5
  R1*2 R1*4 R1*64 R1*16
}
```





### *Posicionar los silencios multicomás*

A diferencia de los silencios normales, no existe una instrucción predefinida para modificar la posición predeterminada de un símbolo de silencio multicomás sobre el pentagrama, adjuntándolo a una nota, independientemente de cuál sea su forma. Sin embargo, en la música polifónica los silencios multicomás de las voces de numeración par e impar están separados verticalmente. La colocación de los silencios multicomás se puede controlar como se ve a continuación:

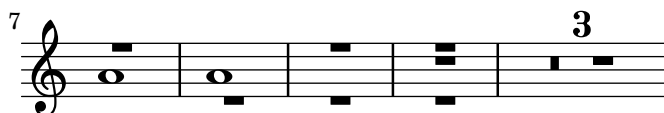
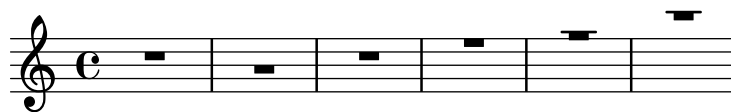
```
\relative c'' {
  % Multi-measure rests by default are set under the fourth line
  R1
  % They can be moved using an override
  \override MultiMeasureRest.staff-position = #-2
  R1
  \override MultiMeasureRest.staff-position = #0
  R1
  \override MultiMeasureRest.staff-position = #2
  R1
  \override MultiMeasureRest.staff-position = #3
  R1
  \override MultiMeasureRest.staff-position = #6
  R1
  \revert MultiMeasureRest.staff-position
  \break

  % In two Voices, odd-numbered voices are under the top line
  << { R1 } \\\ { a1 } >>
  % Even-numbered voices are under the bottom line
  << { a1 } \\\ { R1 } >>
  % Multi-measure rests in both voices remain separate
  << { R1 } \\\ { R1 } >>

  % Separating multi-measure rests in more than two voices
  % requires an override
  << { R1 } \\\ { R1 } \\\
    \once \override MultiMeasureRest.staff-position = #0
    { R1 }
  >>

  % Using compressed bars in multiple voices requires another override
  % in all voices to avoid multiple instances being printed
  \compressMMRests
  <<
    \revert MultiMeasureRest.direction
    { R1*3 }
    \\\
    \revert MultiMeasureRest.direction
    { R1*3 }
  >>
```

}

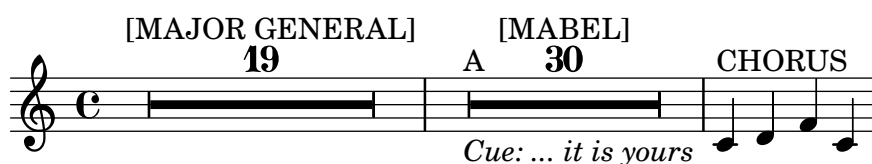


### *Marcado de silencios multicompás*

Los elementos de marcado aplicados a un silencio multicompás se centran encima o debajo de éste. Los elementos de marcado extensos que se adjuntan a silencios multicompás no producen la expansión del compás. Para expandir un silencio multicompás de forma que quepa todo el marcado, utilice un acorde vacío con un marcado aplicado antes del silencio multicompás.

El texto aplicado a un silencio sparador de esta forma se alinea por la izquierda a la posición en que la nota estaría situada dentro del compás, pero si la longitud del compás está determinada por la longitud del texto, éste aparecerá centrado.

```
\relative c' {
  \compressMMRests {
    \textLengthOn
    <>^\markup { [MAJOR GENERAL] }
    R1*19
    <>_\markup { \italic { Cue: ... it is yours } }
    <>^\markup { A }
    R1*30^\markup { [MABEL] }
    \textLengthOff
    c4^\markup { CHORUS } d f c
  }
}
```



### **Véase también**

Glosario musical: Sección “multi-measure rest” en *Glosario Musical*.

Referencia de la notación: Sección 1.2 [Duraciones], página 48, [Escalar las duraciones], página 56, [Compresión de los compases vacíos], página 233, Sección 1.8 [Texto], página 254, Sección 1.8.2 [Formatear el texto], página 264, [Guiones de texto], página 257.

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “MultiMeasureRest” en *Referencia de Funcionamiento Interno*, Sección “MultiMeasureRestNumber” en *Referencia de Funcionamiento Interno*, Sección “MultiMeasureRestScript” en *Referencia de Funcionamiento Interno*, Sección “MultiMeasureRestText” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

En las digitaciones encima de silencios multicompás (p. ej., R1\*10-4), la cifra de la digitación puede chocar con el número del compás.

No hay ninguna forma de condensar automáticamente muchos silencios en un solo silencio multicompás.

Los silencios multicompás no toman parte en las colisiones de silencios.

### 1.2.3 Impresión de las duraciones

#### Indicación de compás

La indicación de compás se establece como sigue:

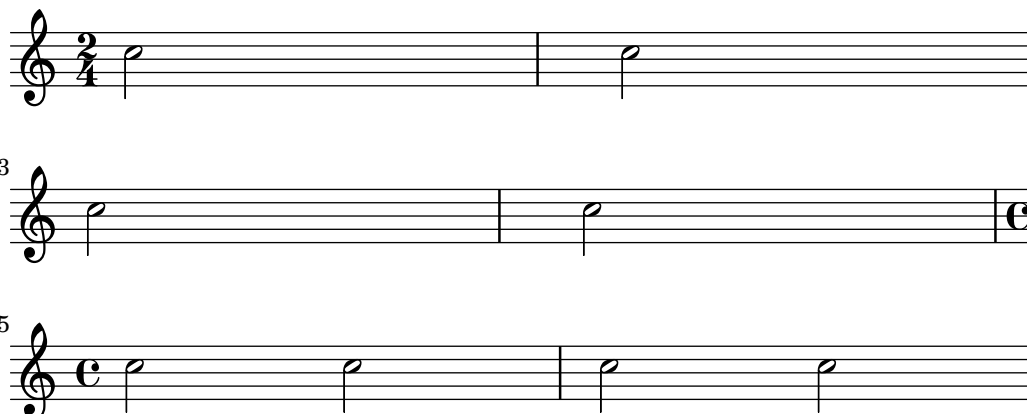
```
\time 2/4 c''2
\time 3/4 c''2.
```



Se estudian los cambios de indicación de compás en el medio de un compás en [Anacrusas], página 78.

La indicación de compás se imprime al comienzo de una pieza y siempre que hay un cambio de compás. Si se produce un cambio al final de una línea, se imprime una indicación de advertencia en dicho lugar. Se puede modificar este comportamiento predeterminado, véase Sección 5.4.7 [Visibilidad de los objetos], página 670.

```
\relative c'' {
  \time 2/4
  c2 c
  \break
  c c
  \break
  \time 4/4
  c c c c
}
```



El símbolo de compás que se usa en 2/2 y 4/4 se puede cambiar a un estilo numérico:

```
\relative c'' {
  % Default style
  \time 4/4 c1
}
```

```

\time 2/2 c1
% Change to numeric style
\numericTimeSignature
\time 4/4 c1
\time 2/2 c1
% Revert to default style
\defaultTimeSignature
\time 4/4 c1
\time 2/2 c1
}

```



Las indicaciones de compás de la música mensural se tratan en [Indicaciones de compás de la música mensural], página 470.

Además de ajustar la indicación de compás que se imprime, la instrucción `\time` establece también los valores de las propiedades basadas en el tipo de compás `baseMoment`, `beatStructure` y `beamExceptions`. Los valores predefinidos de estas propiedades están en `scm/time-signature-settings.scm`.

El valor predeterminado de `beatStructure` puede sobrescribirse dentro de la propia instrucción `\time` escribiéndolo como primer argumento opcional:

```

\score {
  \new Staff {
    \relative {
      \time 2,2,3 7/8
      \repeat unfold 7 { c'8 } |
      \time 3,2,2 7/8
      \repeat unfold 7 { c8 } |
    }
  }
}

```



De forma alternativa, los valores predeterminados de todas estas variables basadas en la indicación de compás, incluidas `baseMoment` y `beamExceptions`, se pueden establecer juntas. Los valores se pueden fijar independientemente para varios tipos de compás distintos. Los nuevos valores tienen efecto cuando se ejecuta una instrucción `\time` posterior con el mismo valor del tipo de compás:

```

\score {
  \new Staff {
    \relative c' {
      \overrideTimeSignatureSettings
      4/4          % timeSignatureFraction
      1/4          % baseMomentFraction
      3,1          % beatStructure
      #'()         % beamExceptions
      \time 4/4
    }
  }
}

```

```

\repeat unfold 8 { c8 } |
}
}
}

```



`\overrideTimeSignatureSettings` toma cuatro argumentos:

1. *timeSignatureFraction*, una fracción que describe el tipo de compás al que se aplican estos valores.
2. *baseMomentFraction*, una fracción que contiene el numerador y denominador de la unidad de medida básica de ese tipo de compás.
3. *beatStructure*, una lista de Scheme que indica la estructura de los pulsos del compás, en unidades del momento base.
4. *beamExceptions*, una lista-A que contiene cualesquiera reglas de barrado para el tipo de compás que vayan más allá de terminar en cada pulso, como se describe en [Establecer el comportamiento de las barras automáticas], página 91.

Los valores modificados de las propiedades predeterminadas del tipo de compás se pueden restaurar a los valores originales:

```

\score {
  \relative {
    \repeat unfold 8 { c'8 } |
    \overrideTimeSignatureSettings
      4/4      % timeSignatureFraction
      1/4      % baseMomentFraction
      3,1      % beatStructure
      #'()     % beamExceptions
    \time 4/4
    \repeat unfold 8 { c8 } |
    \revertTimeSignatureSettings 4/4
    \time 4/4
    \repeat unfold 8 { c8 } |
  }
}

```



Se pueden establecer diferentes valores de las propiedades predeterminadas del tipo de compás para los distintos pentagramas moviendo el `Timing_translator` y el `Default_bar_line_engraver` del contexto `Score` al contexto `Staff`.

```

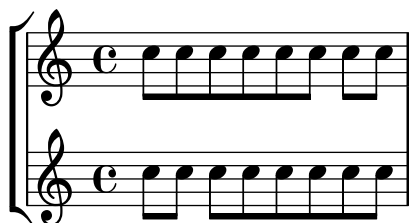
\score {
  \new StaffGroup <<
    \new Staff {
      \overrideTimeSignatureSettings
        4/4      % timeSignatureFraction
        1/4      % baseMomentFraction
        3,1      % beatStructure

```

```

        #'()          % beamExceptions
    \time 4/4
    \repeat unfold 8 {c''8}
}
\new Staff {
    \overrideTimeSignatureSettings
        4/4          % timeSignatureFraction
        1/4          % baseMomentFraction
        1,3          % beatStructure
        #'()          % beamExceptions
    \time 4/4
    \repeat unfold 8 {c''8}
}
>>
\layout {
    \context {
        \Score
        \remove "Timing_translator"
        \remove "Default_bar_line_engraver"
    }
    \context {
        \Staff
        \consists "Timing_translator"
        \consists "Default_bar_line_engraver"
    }
}
}

```



Otro método para cambiar estas variables relacionadas con el tipo de compás, que evita la reimpresión de la indicación de compás en el momento del cambio, se muestra en [Establecer el comportamiento de las barras automáticas], página 91.

## Instrucciones predefinidas

`\numericTimeSignature`, `\defaultTimeSignature`.

## Fragmentos de código seleccionados

### *Indicación de compás imprimiendo sólo el numerador (en lugar de la fracción)*

A veces, la indicación de compás no debe imprimir la fracción completa (p.ej. 7/4), sino sólo el numerador (7 en este caso). Esto se puede hacer fácilmente utilizando `\override Staff.TimeSignature.style = #'single-digit` para cambiar el estilo permanentemente. Usando `\revert Staff.TimeSignature.style`, se puede revertir el cambio. Para aplicar el estilo de un dígito único a una sola indicación de compás, utilice la instrucción `\override` y anteponga la instrucción `\once`.

```

\relative c'' {
  \time 3/4
  c4 c c
  % Change the style permanently
  \override Staff.TimeSignature.style = #'single-digit
  \time 2/4
  c4 c
  \time 3/4
  c4 c c
  % Revert to default style:
  \revert Staff.TimeSignature.style
  \time 2/4
  c4 c
  % single-digit style only for the next time signature
  \once \override Staff.TimeSignature.style = #'single-digit
  \time 5/4
  c4 c c c c
  \time 2/4
  c4 c
}

```



## Véase también

Glosario musical: Sección “time signature” en *Glosario Musical*

Referencia de la notación: [Indicaciones de compás de la música mensural], página 470, [Establecer el comportamiento de las barras automáticas], página 91, [Gestión del tiempo], página 126.

Archivos instalados: `scm/time-signature-settings.scm`.

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TimeSignature” en *Referencia de Funcionamiento Interno*, Sección “Timing-translator” en *Referencia de Funcionamiento Interno*.

## Indicaciones metronómicas

Es muy sencillo escribir una indicación metronómica básica:

```

\relative {
  \tempo 4 = 120
  c'2 d
  e4. d8 c2
}

```



También se pueden imprimir indicaciones metronómicas como un intervalo entre dos números:

```

\relative {
  \tempo 4 = 40 - 46
}

```

```
c'4. e8 a4 g
b,2 d4 r
}
```



Se pueden usar indicaciones de tempo con texto:

```
\relative {
  \tempo "Allegretto"
  c'4 e d c
  b4. a16 b c4 r4
}
```



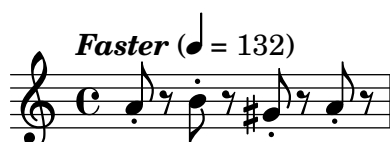
La combinación de una indicación metronómica y un texto hace que la marca de metrónomo se encierre entre paréntesis automáticamente:

```
\relative {
  \tempo "Allegro" 4 = 160
  g'4 c d e
  d4 b g2
}
```



En general, el texto puede ser cualquier objeto de marcado:

```
\relative {
  \tempo \markup { \italic Faster } 4 = 132
  a'8-. r8 b-. r gis-. r a-. r
}
```

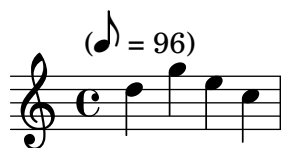


Se puede escribir una indicación metronómica entre paréntesis sin ninguna indicación textual, escribiendo una cadena vacía en la entrada:

```
\relative {
  \tempo "" 8 = 96
  d'4 g e c
}
```

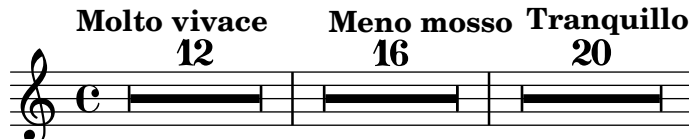


}



En una particella de un instrumento con períodos de silencio largos (véase [Silencios de compás completo], página 65), en ocasiones se suceden muy cerca distintas indicaciones de tempo. La instrucción `\markLengthOn` aporta un espacio horizontal adicional para evitar que las indicaciones de tempo se superpongan, y `\markLengthOff` restablece el comportamiento predeterminado de ignorar las indicaciones de tempo para el espaciado horizontal.

```
\compressMMRests {
  \markLengthOn
  \tempo "Molto vivace"
  R1*12
  \tempo "Meno mosso"
  R1*16
  \markLengthOff
  \tempo "Tranquillo"
  R1*20
}
```



## Fragmentos de código seleccionados

### *Impresión de indicaciones metronómicas y letras de ensayo debajo del pentagrama*

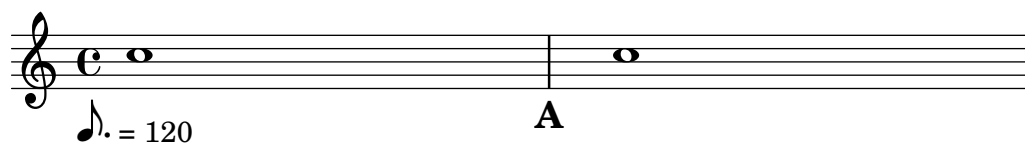
De forma predeterminada, las indicaciones metronómicas y las letras de ensayo se imprimen encima del pentagrama. Para colocarlas debajo del pentagrama, simplemente ajustamos adecuadamente la propiedad `direction` de `MetronomeMark` o de `RehearsalMark`.

```
\layout {
  indent = 0
  ragged-right = ##f
}

{
  % Metronome marks below the staff
  \override Score.MetronomeMark.direction = #DOWN
  \tempo 8. = 120
  c''1

  % Rehearsal marks below the staff
  \override Score.RehearsalMark.direction = #DOWN
  \mark \default
  c''1
}
```

}



### *Cambiar el tempo sin indicación metronómica*

Para cambiar el tempo en la salida MIDI sin imprimir nada, hacemos invisible la indicación metronómica:

```
\score {
  \new Staff \relative c' {
    \tempo 4 = 160
    c4 e g b
    c4 b d c
    \set Score.tempohideNote = ##t
    \tempo 4 = 96
    d,4 fis a cis
    d4 cis e d
  }
  \layout { }
  \midi { }
}
```



### *Crear indicaciones metronómicas en modo de marcado*

Se pueden crear indicaciones metronómicas nuevas en modo de marcado, pero no cambian el tempo en la salida MIDI.

```
\relative c' {
  \tempo \markup {
    \concat {
      (
        \smaller \general-align #Y #DOWN \note {16.} #1
        " = "
        \smaller \general-align #Y #DOWN \note {8} #1
      )
    }
  }
  c1
  c4 c' c,2
}
```



Para ver más detalles, consulte Sección 1.8.2 [Formatear el texto], página 264.

## Véase también

Glosario musical: Sección “metronome” en *Glosario Musical*, Sección “metronomic indication” en *Glosario Musical*, Sección “tempo indication” en *Glosario Musical*, Sección “metronome mark” en *Glosario Musical*.

Referencia de la notación: Sección 1.8.2 [Formatear el texto], página 264, Sección 3.5 [Creación de salida MIDI], página 552, [Silencios de compás completo], página 65.

Fragmentos de código: Sección “Staff notation” en *Fragmentos de código*.

Referencia de funcionamiento interno:: Sección “MetronomeMark” en *Referencia de Funcionamiento Interno*.

## Anacrusas

Los compases parciales como las *anacrusas* o partes *al alzar* se escriben usando la instrucción `\partial`:

```
\partial duración
```

Cuando se usa `\partial` al principio de la partitura, *duración* es el tiempo de anacrusa, la longitud de la música que precede al primer compás.

```
\relative {
  \time 3/4
  \partial 4.
  r4 e'8 | a4 c8 b c4 |
}
```



Cuando se usa `\partial` después del comienzo de la partitura, *duración* es la longitud *restante* del compás actual. No crea un nuevo compás con numeración.

```
\relative {
  \set Score.barNumberVisibility = #all-bar-numbers-visible
  \override Score.BarNumber.break-visibility =
    #end-of-line-invisible
  \time 9/8
  d'4.~ 4 d8 d( c) b | c4.~ 4. \bar "||"
  \time 12/8
  \partial 4.
  c8( d) e | f2.~ 4 f8 a,( c) f |
}
```



La instrucción `\partial` es *necesaria* cuando cambia la indicación de compás en medio de un compás, pero también puede usarse sola.

```
\relative {
  \set Score.barNumberVisibility = #all-bar-numbers-visible
  \override Score.BarNumber.break-visibility =
    #end-of-line-invisible
  \time 6/8
```

```

\partial 8
e'8 | a4 c8 b[ c b] |
\partial 4
r8 e,8 | a4 \bar "||"
\partial 4
r8 e8 | a4
c8 b[ c b] |
}

```



La instrucción `\partial` fija la propiedad `Timing.measurePosition`, que un número racional que indica cuánto tiempo ha transcurrido dentro del compás actual.

## Véase también

Glosario musical: Sección “anacrusis” en *Glosario Musical*.

Referencia de la notación: [Notas de adorno], página 120.

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Timing\_translator” en *Referencia de Funcionamiento Interno*.

## Música sin compasear

En la música medida se insertan líneas divisorias y se calculan los números de compás automáticamente. En música sin metro (es decir, cadencias), esto no es deseable y se puede ‘desactivar’ usando la instrucción `\cadenzaOn`, para después ‘reactivarlo’ en el lugar adecuado usando `\cadenzaOff`.

```

\relative c'' {
  c4 d e d
  \cadenzaOn
  c4 c d8[ d d] f4 g4.
  \cadenzaOff
  \bar "||"
  d4 e d c
}

```



La numeración de compases se continúa al final de la cadencia.

```

\relative c'' {
  % Show all bar numbers
  \override Score.BarNumber.break-visibility = #all-visible
  c4 d e d
  \cadenzaOn
  c4 c d8[ d d] f4 g4.
  \cadenzaOff
  \bar "||"
  d4 e d c
}

```

}



Al insertar una instrucción `\bar` dentro de una cadencia no se inicia un compás nuevo, incluso aunque se imprima una línea divisoria. Así pues, las alteraciones accidentales (cuyo efecto se suele suponer que permanece hasta el final del compás) serán válidas aún después de la línea divisoria que se imprime por parte de `\bar`. Si se quieren imprimir alteraciones accidentales posteriores tendrán que insertarse manualmente alteraciones forzadas o de cortesía, véase [Alteraciones accidentales], página 6.

```
\relative c'' {
  c4 d e d
  \cadenzaOn
  cis4 d cis d
  \bar "|"
  % First cis is printed without alteration even if it's after a \bar
  cis4 d cis! d
  \cadenzaOff
  \bar "|"
}
```



El barrado automático se desactiva mediante `\cadenzaOn`. Por tanto, todo el barrado en las cadencias se debe introducir manualmente. Véase [Barras manuales], página 100.

```
\relative {
  \repeat unfold 8 { c''8 }
  \cadenzaOn
  cis8 c c c c
  \bar""|"
  c8 c c
  \cadenzaOff
  \repeat unfold 8 { c8 }
}
```



Estas instrucciones predefinidas afectan a todas las pautas de la partitura, aunque aparezcan en un solo contexto **Voice**. Para cambiar esto, traslade el `Timing_translator` del contexto **Score** al contexto **Staff**, véase [Notación polimétrica], página 81.

## Instrucciones predefinidas

`\cadenzaOn`, `\cadenzaOff`.

## Véase también

Glosario musical: Sección “cadenza” en *Glosario Musical*.

Referencia de la notación: Sección 5.4.7 [Visibilidad de los objetos], página 670, [Notación polimétrica], página 81, [Barras manuales], página 100, [Alteraciones accidentales], página 6.

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

## Advertencias y problemas conocidos

Sólo se insertan saltos de línea y de página automáticamente en las líneas divisorias, por lo que es necesario insertar manualmente líneas divisorias ‘invisibles’ en fragmentos largos de música sin medir para permitir los saltos:

```
\bar ""
```

## Notación polimétrica

Está contemplada la notación polimétrica, ya sea explícitamente o mediante la modificación manual de la indicación visible del compás y/o el escalado de la duración de las notas.

### *Pentagramas con distintas indicaciones de compás y compases de igual longitud*

Establezca una indicación de compás común para cada pentagrama, y fije `timeSignatureFraction` a la fracción deseada. Luego use la función `\scaleDurations` para escalar las duraciones de las notas en cada pauta a la indicación de compás común.

En el siguiente ejemplo, se usa en paralelo música con compases de 3/4, 9/8 y 10/8. En el segundo pentagrama, las duraciones mostradas se multiplican por 2/3 (pues  $2/3 * 9/8 = 3/4$ ), y en el tercer pentagrama, las duraciones que se muestran están multiplicadas por 3/5 (pues  $3/5 * 10/8 = 3/4$ ). Con frecuencia habrá que insertar las barras de forma manual, pues el escalado de las duraciones afecta a las reglas de barrado automático.

```
\relative <<
\new Staff {
  \time 3/4
  c'4 c c |
  c4 c c |
}
\new Staff {
  \time 3/4
  \set Staff.timeSignatureFraction = 9/8
  \scaleDurations 2/3
  \repeat unfold 6 { c8[ c c] }
}
\new Staff {
  \time 3/4
  \set Staff.timeSignatureFraction = 10/8
  \scaleDurations 3/5 {
    \repeat unfold 2 { c8[ c c] }
    \repeat unfold 2 { c8[ c] } |
    c4. c \tuplet 3/2 { c8[ c c] } c4
  }
}
```

&gt;&gt;



### *Pentagramas con distintas indicaciones de compás y longitudes de compás distintas*

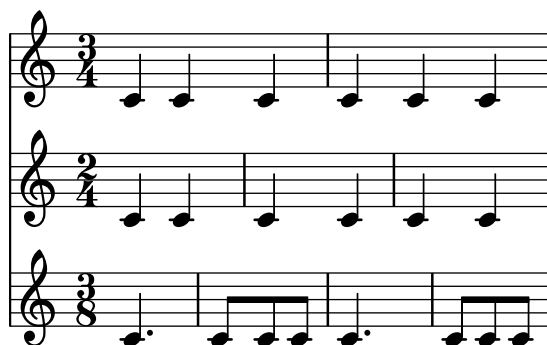
Se puede dar a cada pentagrama su propia indicación de compás independiente trasladando los grabadores `Timing_translator` y `Default_bar_line_engraver` al contexto de `Staff`.

```
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Default_bar_line_engraver"
  }
  \context {
    \Staff
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  }
}

% Now each staff has its own time signature.

\relative <<
  \new Staff {
    \time 3/4
    c'4 c c |
    c4 c c |
  }
  \new Staff {
    \time 2/4
    c4 c |
    c4 c |
    c4 c |
  }
  \new Staff {
    \time 3/8
    c4. |
    c8 c c |
    c4. |
    c8 c c |
  }
}
```

&gt;&gt;



### *Indicaciones de compás compuesto*

Se crean usando la función `\compoundMeter`. La sintaxis es:

```
\compoundMeter #'(lista de listas)
```

La construcción más simple es una lista única, en la que el *último* número indica el denominador de la indicación de compás y los anteriores son los numeradores.

```
\relative {
  \compoundMeter #'((2 2 2 8))
  \repeat unfold 6 c'8 \repeat unfold 12 c16
}
```



Se pueden construir compases más complejos utilizando listas adicionales. Asimismo, los ajustes de barrado se ajustarán dependiendo de los valores.

```
\relative {
  \compoundMeter #'((1 4) (3 8))
  \repeat unfold 5 c'8 \repeat unfold 10 c16
}
```

```
\relative {
  \compoundMeter #'((1 2 3 8) (3 4))
  \repeat unfold 12 c'8
}
```



### Véase también

Glosario musical: Sección “polymetric” en *Glosario Musical*, Sección “polymetric time signature” en *Glosario Musical*, Sección “meter” en *Glosario Musical*.



Referencia de la notación: [Barras automáticas], página 88, [Barras manuales], página 100, [Indicación de compás], página 70, [Escalar las duraciones], página 56.

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TimeSignature” en *Referencia de Funcionamiento Interno*, Sección “Timing\_translator” en *Referencia de Funcionamiento Interno*, Sección “Default\_bar\_line\_engraver” en *Referencia de Funcionamiento Interno*, Sección “Staff” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Aunque las notas que se producen en el mismo momento en distintos pentagramas se colocan en la misma posición horizontal, las barras de compás en cada pauta pueden hacer que el espaciado de notas sea menos regular según cada tipo de compás.

La utilización de un bloque `midi` con notación polimétrica puede causar advertencias inesperadas de comprobación de compás. En este caso traslade el `Timing_translator` del contexto de `Score` al contexto de `Staff` dentro del bloque `midi`.

```
\midi {
  \context {
    \Score
    \remove "Timing_translator"
  }
  \context {
    \Staff
    \consists "Timing_translator"
  }
}
```

## División automática de las notas

Las notas largas se pueden convertir automáticamente en notas ligadas. Se hace mediante la sustitución del `Note_heads_engraver` por el `Completion_heads_engraver`. De forma similar, los silencios largos que sobrepasan líneas de compás se dividen automáticamente sustituyendo el grabador `Rest_engraver` con el grabador `Completion_rest_engraver`. En el ejemplo siguiente, las notas y los silencios que atraviesan la barra de compás se dividen, y además las notas se unen mediante una ligadura.

```
\new Voice \with {
  \remove "Note_heads_engraver"
  \consists "Completion_heads_engraver"
  \remove "Rest_engraver"
  \consists "Completion_rest_engraver"
}
\relative {
  c'2. c8 d4 e f g a b c8 c2 b4 a g16 f4 e d c8. c2 r1*2
}
```



Estos grabadores dividen todas las notas y silencios largos en la barra de compás, e inserta ligaduras en las notas. Uno de sus usos es depurar partituras complejas: si los compases no están completos, las ligaduras mostrarán exactamente cuánto le falta a cada compás.

La propiedad `completionUnit` fija una duración preferida para las notas divididas.

```
\new Voice \with {
  \remove "Note_heads_engraver"
  \consists "Completion_heads_engraver"
} \relative {
  \time 9/8 g\breve. d''4. \bar "||"
  \set completionUnit = #(ly:make-moment 3 8)
  g\breve. d4.
}
```



Estos grabadores dividen las notas que tienen la duración escalada, como las de los tresillos, en notas que tienen el mismo factor de escala que la nota original de la entrada.

```
\new Voice \with {
  \remove "Note_heads_engraver"
  \consists "Completion_heads_engraver"
} \relative {
  \time 2/4 r4
  \tuplet 3/2 {g'4 a b}
  \scaleDurations 2/3 {g a b}
  g4*2/3 a b
  \tuplet 3/2 {g4 a b}
  r4
}
```



## Véase también

Glosario musical: Sección “tie” en *Glosario Musical*

Manual de aprendizaje: Sección “Explicación de los grabadores” en *Manual de Aprendizaje*, Sección “Añadir y eliminar grabadores” en *Manual de Aprendizaje*.

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Note\_heads\_engraver” en *Referencia de Funcionamiento Interno*, Sección “Completion\_heads\_engraver” en *Referencia de Funcionamiento Interno*, Sección “Rest\_engraver” en *Referencia de Funcionamiento Interno*, Sección “Completion\_rest\_engraver” en *Referencia de Funcionamiento Interno*, Sección “Forbid\_line\_break\_engraver” en *Referencia de Funcionamiento Interno*.


## Advertencias y problemas conocidos

Por consistencia con el comportamiento anterior, las notas y silencios que tienen una duración mayor de un compás, como `c1*2`, se dividen en notas sin ningún factor de escala, `{ c1 c1 }`. La propiedad `completionFactor` controla este comportamiento, y al darle el valor `#f` podemos hacer que las notas y silencios divididos tengan el mismo factor de escala que las duraciones de las notas originales de la entrada.

## Mostrar los ritmos de la melodía

A veces podemos querer mostrar solamente el ritmo de una melodía. Esto se puede hacer con un pentagrama de ritmo. Todas las alturas de las notas se convierten en barras inclinadas, y el pentagrama tiene una sola línea:

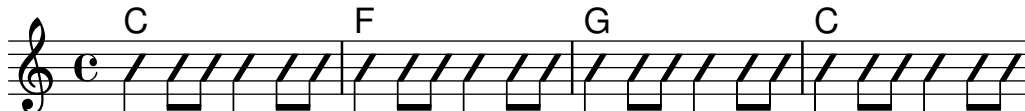
```
<<
  \new RhythmicStaff {
    \new Voice = "myRhythm" \relative {
      \time 4/4
      c'4 e8 f g2
      r4 g g f
      g1
    }
  }
  \new Lyrics {
    \lyricsto "myRhythm" {
      This is my song
      I like to sing
    }
  }
>>
```



This is my song      I like to    sing

Las tablas de acordes de guitarra ofrecen a menudo los ritmos de rasgado. Esto se puede hacer con el grabador `Pitch_squash_engraver` y `\improvisationOn`.

```
<<
  \new ChordNames {
    \chordmode {
      c1 f g c
    }
  }
  \new Voice \with {
    \consists "Pitch_squash_engraver"
  } \relative c'' {
    \improvisationOn
    c4 c8 c c4 c8 c
    f4 f8 f f4 f8 f
    g4 g8 g g4 g8 g
    c4 c8 c c4 c8 c
  }
>>
```



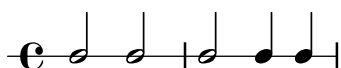
También se puede usar música que contenga acordes como entrada para `RhythmicStaff` y para usarla con el grabador `Pitch_squash_engraver` si los acordes se reducen primero a notas individuales con la función musical `\reduceChords`:

```
\new RhythmicStaff {
```

```

\time 4/4
\reduceChords {
  <c>2
  <e>2
  <c e g>2
  <c e g>4
  <c e g>4
}
}

```



## Instrucciones predefinidas

\improvisationOn, \improvisationOff, \reduceChords.

## Fragmentos de código seleccionados

### *Ritmos rasqueados de guitarra*

Para la música de guitarra, es posible mostrar los ritmos de rasqueo, además de las notas de la melodía, acordes y diagramas de posiciones.

```

\include "predefined-guitar-fretboards.ly"
<<
  \new ChordNames {
    \chordmode {
      c1 | f | g | c
    }
  }
  \new FretBoards {
    \chordmode {
      c1 | f | g | c
    }
  }
  \new Voice \with {
    \consists "Pitch_squash_engraver"
  } {
    \relative c'' {
      \improvisationOn
      c4 c8 c c4 c8 c
      f4 f8 f f4 f8 f
      g4 g8 g g4 g8 g
      c4 c8 c c4 c8 c
    }
  }
  \new Voice = "melody" {
    \relative c'' {
      c2 e4 e4
      f2. r4
      g2. a4
      e4 c2.
    }
  }
}

```

```

\new Lyrics {
  \lyricsto "melody" {
    This is my song.
    I like to sing.
  }
}
>>

```

## Véase también

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “RhythmicStaff” en *Referencia de Funcionamiento Interno*, Sección “Pitch\_squash\_engraver” en *Referencia de Funcionamiento Interno*.

### 1.2.4 Barras

#### Barras automáticas

De manera predeterminada, las barras de corchea se insertan automáticamente:

```

\relative c' {
  \time 2/4 c8 c c c
  \time 6/8 c8 c c c8. c16 c8
}

```

Cuando estas decisiones automáticas no son lo bastante buenas, se pueden escribir los barrados de forma explícita; véase [Barras manuales], página 100. Las barras *se deben* introducir manualmente si se quieren extender por encima de los silencios.

Si no se necesita el barrado automático, se puede desactivar con `\autoBeamOff` y activarse con `\autoBeamOn`:

```
\relative c' {
  c4 c8 c8. c16 c8. c16 c8
  \autoBeamOff
  c4 c8 c8. c16 c8.
  \autoBeamOn
  c16 c8
}
```



**Nota:** Si se usan barras para indicar los melismas de las canciones, entonces se debe desactivar el barrado automático con `\autoBeamOff` e indicar las barras manualmente. La utilización de `\partCombine` con `\autoBeamOff` puede producir resultados no deseados. Véanse los fragmentos de código para mayor información.

Se pueden crear patrones de barrado que difieran de los valores automáticos predeterminados; véase [Establecer el comportamiento de las barras automáticas], página 91.

## Instrucciones predefinidas

`\autoBeamOff`, `\autoBeamOn`.

## Fragmentos de código seleccionados

### *Barras que atraviesan saltos de línea*

Normalmente están prohibidos los saltos de línea si las barras atraviesan las líneas divisorias. Se puede cambiar este comportamiento como se muestra aquí:

```
\relative c'' {
  \override Beam.breakable = ##t
  c8 c[ c] c[ c] c[ c] c[ \break
  c8] c[ c] c[ c] c[ c] c
}
```



### *Cambiar el salto de las barras en ángulo*

Se insertan automáticamente barras en ángulo cuando se detecta un intervalo muy grande entre las notas. Se puede hacer un ajuste fino de este comportamiento a través de la propiedad `auto-knee-gap`. Se traza una barra doblada si el salto es mayor que el valor de `auto-knee-gap`.

más el ancho del objeto barra (que depende de la duración de las notas y de la inclinación de la barra). De forma predeterminada `auto-knee-gap` está establecido a 5.5 espacios de pentagrama.

```
{
  f8 f''8 f8 f''8
  \override Beam.auto-knee-gap = #6
  f8 f''8 f8 f''8
}
```



### *PartCombine y autoBeamOff*

Puede ser difícil comprender la función de `\autoBeamOff` cuando se usa con `\partCombine`.

Puede ser preferible usar

```
\set Staff.autoBeaming = ##f
```

en su lugar, para asegurarse de que el barrado automático se desactiva para todo el pentagrama.

`\partCombine` funciona aparentemente con tres voces: plica arriba cuando hay una sola voz, plica abajo con la otra, y plica arriba cuando está combinada.

Una llamada a `\autoBeamOff` en el primer argumento de `partCombine` se aplica a la voz que está activa en el momento en que se procesa la llamada, ya sea voz única con la plica hacia arriba, hacia abajo o combinadas. Una llamada a `\autoBeamOff` en el segundo argumento se aplica a la voz que está sola con la plica abajo.

Para poder usar `\autoBeamOff` con el objeto de detener todo el barrado automático cuando se usa con `\partCombine`, es necesario hacer tres llamadas a `\autoBeamOff`.

```
{
  %\set Staff.autoBeaming = ##f % turns off all autobeamning
  \partCombine
  {
    \autoBeamOff % applies to split up stems
    \repeat unfold 4 a'16
    %\autoBeamOff % applies to combined up stems
    \repeat unfold 4 a'8
    \repeat unfold 4 a'16
  }
  {
    \autoBeamOff % applies to down stems
    \repeat unfold 4 f'8
    \repeat unfold 8 f'16 |
  }
}
```



## Véase también

Referencia de la notación: [Barras manuales], página 100, [Establecer el comportamiento de las barras automáticas], página 91.

Archivos instalados: `scm/auto-beam.scm`.

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Auto-beam-engraver” en *Referencia de Funcionamiento Interno*, Sección “Beam-engraver” en *Referencia de Funcionamiento Interno*, Sección “Beam” en *Referencia de Funcionamiento Interno*, Sección “BeamEvent” en *Referencia de Funcionamiento Interno*, Sección “BeamForbidEvent” en *Referencia de Funcionamiento Interno*, Sección “beam-interface” en *Referencia de Funcionamiento Interno*, Sección “unbreakable-spanner-interface” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Las propiedades de una barra vienen determinadas al *comienzo* de su construcción y cualquier cambio adicional en las propiedades de la barra que se produzca antes de que la barra se ha completado no tendrá efecto hasta que inicie la *siguiente* barra nueva.

## Establecer el comportamiento de las barras automáticas

Cuando está habilitado el barrado automático, la colocación de las barras automáticas viene determinada por tres propiedades de contexto: `baseMoment`, `beatStructure` y `beamExceptions`. Los valores predeterminados de estas variables se pueden sobrescribir como se describe más abajo, o de forma alternativa los propios valores predeterminados se pueden cambiar como se explica en [Indicación de compás], página 70.

Si hay definida una regla de `beamExceptions` para el compás en curso, se usa dicha regla para determinar la colocación de las barras; se ignoran los valores de `baseMoment` y `beatStructure`.

Si no está definida ninguna regla de `beamExceptions` para el tipo de compás en curso, la colocación de las barras está determinada por los valores de `baseMoment` y `beatStructure`.

## *Barrado basado en baseMoment y beatStructure*

De forma predeterminada, las reglas de `beamExceptions` están definidas para los compases más comunes, y así las reglas de `beamExceptions` se deben desactivar si pretendemos que el barrado automático esté basado en `baseMoment` y `beatStructure`. Las reglas de `beamExceptions` se desactivan mediante

```
\set Timing.beamExceptions = #'()
```

Cuando el valor de `beamExceptions` se ha fijado a `#'()`, ya sea debido a un ajuste explícito o a causa de que no hay ninguna regla de `beamExceptions` definida internamente para el compás actual, los puntos finales de las barras están en los pulsos según viene determinado por las propiedades de contexto `baseMoment` y `beatStructure`. `beatStructure` es una lista de Scheme que define la longitud de cada pulso dentro del compás en unidades de `baseMoment`. De forma predeterminada, `baseMoment` es una unidad más que el denominador del compás. De forma predeterminada también, cada unidad de longitud `baseMoment` es un único pulso.

Observe que existen valores de `beatStructure` y de `baseMoment` diferentes para cada indicación de compás. Los cambios que se hacen a estas variables se aplican solamente al tipo de compás en vigor, por lo que dichos cambios se deben escribir después de la instrucción `\time` que da comienzo a una sección nueva con un tipo de compás distinto, no antes. Los valores nuevos que se dan a una indicación de compás concreta se retienen y se vuelven a aplicar cuando ese tipo de compás vuelve a establecerse.

```
\relative c'' {  
  \time 5/16
```



```

c16^"default" c c c c |
% beamExceptions are unlikely to be defined for 5/16 time,
% but let's disable them anyway to be sure
\set Timing.beamExceptions = #'()
\set Timing.beatStructure = 2,3
c16^(2+3)" c c c c |
\set Timing.beatStructure = 3,2
c16^(3+2)" c c c c |
}

```



```

\relative {
  \time 4/4
  a'8^"default" a a a a a a
  % Disable beamExceptions because they are definitely
  % defined for 4/4 time
  \set Timing.beamExceptions = #'()
  \set Timing.baseMoment = #(ly:make-moment 1/4)
  \set Timing.beatStructure = 1,1,1,1
  a8^"changed" a a a a a a
}

```



Los cambios en los ajustes de barrado se pueden limitar a contextos específicos. Si no se incluye ningún ajuste en un contexto de nivel más bajo, se aplican los ajustes del contexto circundante.

```

\new Staff {
  \time 7/8
  % No need to disable beamExceptions
  % as they are not defined for 7/8 time
  \set Staff.beatStructure = 2,3,2
  <<
    \new Voice = one {
      \relative {
        a'8 a a a a a a
      }
    }
    \new Voice = two {
      \relative {
        \voiceTwo
        \set Voice.beatStructure = 1,3,3
        f'8 f f f f f f
      }
    }
  >>
}

```

}



En caso de usar varias voces, se debe especificar el contexto `Staff` si queremos aplicar el barrado a todas las voces del pentagrama:

```
\time 7/8
% rhythm 3-1-1-2
% Change applied to Voice by default -- does not work correctly
% Because of autogenerated voices, all beating will
% be at baseMoment (1 . 8)
\set beatStructure = 3,1,1,2
<< \relative {a'8 a a a16 a a a a8 a} \\ \relative {f'4. f8 f f f} >>

% Works correctly with context Staff specified
\set Staff.beatStructure = 3,1,1,2
<< \relative {a'8 a a a16 a a a a8 a} \\ \relative {f'4. f8 f f f} >>
```



El valor de `baseMoment` se puede ajustar para cambiar el comportamiento de las barras, si se desea. Cuando se hace, el valor de `beatStructure` se debe fijar de manera que sea compatible con el nuevo valor de `baseMoment`.

```
\time 5/8
% No need to disable beamExceptions
% as they are not defined for 5/8 time
\set Timing.baseMoment = #(ly:make-moment 1/16)
\set Timing.beatStructure = 7,3
\repeat unfold 10 { a'16 }
```



`baseMoment` es un *momento*, una unidad de duración musical. Se crea una cantidad del tipo *momento* por medio de la función de Scheme `ly:make-moment`. Para ver más información acerca de esta función, consulte [Gestión del tiempo], página 126.

De forma predeterminada `baseMoment` está fijado a una unidad más que el denominador del compás. Todas las excepciones a este valor predeterminado están en `scm/time-signature-settings.scm`.

### ***Barrado basado en beamExceptions***

Las reglas de autobarrado especiales (distintas de terminar una barra sobre un pulso) están definidas en la propiedad `beamExceptions`.

El valor de `beamExceptions`, que es una estructura de datos de Scheme bastante compleja, se genera fácilmente con la función `\beamExceptions`. Esta función recibe uno o más patrones rítmicos barrados manualmente y con un compás de duración (los compases se deben separar

por una barra de comprobación de compás | porque la función no tiene otra manera de discernir la longitud del compás). He aquí un ejemplo sencillo:

```
\relative c'' {
  \time 3/16
  \set Timing.beatStructure = 2,1
  \set Timing.beamExceptions =
    \beamExceptions { 32[ 32] 32[ 32] 32[ 32] }
  c16 c c |
  \repeat unfold 6 { c32 } |
}
```



**Nota:** Un valor de `beamExceptions` debe ser una lista de excepciones *completa*. Esto es, toda excepción que se tenga que aplicar debe estar incluida en este ajuste. No es posible añadir, eliminar o cambiar sólo una de las excepciones. Aunque esto puede parecer engorroso, significa que no es necesario conocer los ajustes de barrado actuales para poder especificar un patrón de barrado nuevo.

Cuando cambia el compás, se fijan los valores predeterminados de `Timing.baseMoment`, `Timing.beatStructure` y `Timing.beamExceptions`. Un ajuste en el tipo de compás da como resultado un reinicio de los ajustes de barrado automático para el contexto `Timing` al comportamiento predeterminado.

```
\relative a' {
  \time 6/8
  \repeat unfold 6 { a8 }
  % group (4 + 2)
  \set Timing.beatStructure = 4,2
  \repeat unfold 6 { a8 }
  % go back to default behavior
  \time 6/8
  \repeat unfold 6 { a8 }
}
```



Los ajustes de barrado automático predeterminados para un tipo de compás están determinados en el archivo `scm/time-signature-settings.scm`. La forma de cambiar los ajustes predeterminados de barrado automático para un tipo de compás se describe en [Indicación de compás], página 70.

Muchos ajustes de barrado automáticos para un tipo de compás contienen una entrada para `beamExceptions`. Por ejemplo, el compás de 4/4 trata de unir el compás en dos partes si solo hay corcheas. La regla `beamExceptions` puede sobrescribir el ajuste `beatStructure` si no se reinicia `beamExceptions`.

```
\time 4/4
\set Timing.baseMoment = #(ly:make-moment 1/8)
```

```

\set Timing.beatStructure = 3,3,2
% This won't beam (3 3 2) because of beamExceptions
\repeat unfold 8 {c''8} |
% This will beam (3 3 2) because we clear beamExceptions
\set Timing.beamExceptions = #'()
\repeat unfold 8 {c''8}

```



De forma similar, las corcheas en compás de 3/4 se unen mediante una sola barra para todo el compás, de forma predeterminada. Para unir las corcheas en 3/4 mediante una barra en cada parte, reinicie `beamExceptions`.

```

\time 3/4
% by default we beam in (6) due to beamExceptions
\repeat unfold 6 {a'8} |
% This will beam (1 1 1) due to default baseMoment and beatStructure
\set Timing.beamExceptions = #'()
\repeat unfold 6 {a'8}

```



En la música tipografiada de los períodos clásico y romántico, con frecuencia las barras comienzan a mitad de un compás en 3/4, pero la práctica moderna es evitar la falsa impresión de 6/8 (véase Gould, pág. 153). Se producen situaciones similares en el compás de 3/8. Este comportamiento viene controlado mediante la propiedad de contexto `beamHalfMeasure`, que tiene efecto solamente sobre indicaciones de compás que tienen la cifra 3 en el numerador:

```

\relative a' {
  \time 3/4
  r4. a8 a a |
  \set Timing.beamHalfMeasure = ##f
  r4. a8 a a |
}

```



### *Cómo funciona el barrado automático*

Cuando se habilita el barrado automático, la colocación de las barras de corchea automáticas viene determinada por las propiedades de contexto `baseMoment`, `beatStructure` y `beamExceptions`.

Son de aplicación las siguientes reglas, en orden de prioridad, cuando se determina el aspecto de las barras:

- Si está especificada una barra manual con [...], fijar la barra tal y como se ha especificado; en caso contrario,
- si está definida en `beamExceptions` una regla de barrado para este tipo de barra, utilizarla para determinar los lugares válidos en que pueden terminar las barras; en caso contrario,

- si está definida en `beamExceptions` una regla de final de barra para un tipo de barra más largo, utilizarla para determinar los lugares válidos en que pueden terminar las barras; en caso contrario,
- usar los valores de `baseMoment` y de `beatStructure` para determinar los finales de los pulsos dentro del compás, y terminar las barras al final de los pulsos.

En las reglas anteriores, el tipo de barra *beam-type* es la duración de la nota más breve dentro del grupo unido por una barra.

Las reglas de barrado predefinidas están en el archivo `scm/time-signature-settings.scm`.

## Fragmentos de código seleccionados

### *Subdivisión de barras*

Las barras de semicorchea, o de figuras más breves, no se subdividen de forma predeterminada. Esto es, las tres (o más) barras se amplían sin romperse sobre grupos completos de notas. Este comportamiento se puede modificar para subdividir las barras en subgrupos mediante el establecimiento de la propiedad `subdivideBeams`. Cuando está activada, las barras se subdividen a intervalos definidos por el valor actual de `baseMoment` mediante la reducción de las barras repetidas a una sola entre los subgrupos. Observe que el valor predeterminado de `baseMoment` es uno más que el denominador del tipo de compás actual, si no se fija explícitamente. Se debe ajustar a una fracción que da la duración del subgrupo de barras utilizando la función `ly:make-moment`, como se ve en este fragmento de código. Asimismo, cuando se modifica `baseMoment`, se debería cambiar también `beatStructure` para que corresponda al `baseMoment` nuevo:

```
\relative c'' {
  c32[ c c c c c c c]
  \set subdivideBeams = ##t
  c32[ c c c c c c c]

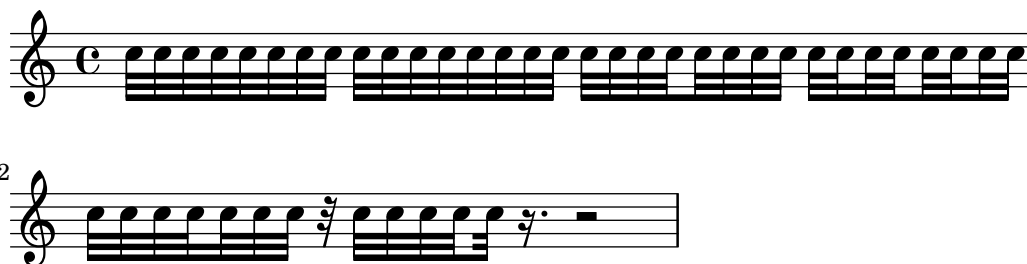
  % Set beam sub-group length to an eighth note
  \set baseMoment = #(ly:make-moment 1/8)
  \set beatStructure = 2,2,2,2
  c32[ c c c c c c c]

  % Set beam sub-group length to a sixteenth note
  \set baseMoment = #(ly:make-moment 1/16)
  \set beatStructure = 4,4,4,4
  c32[ c c c c c c c]

  % Shorten beam by 1/32
  \set baseMoment = #(ly:make-moment 1/8)
  \set beatStructure = 2,2,2,2
  c32[ c c c c c c] r32

  % Shorten beam by 3/32
  \set baseMoment = #(ly:make-moment 1/8)
  \set beatStructure = 2,2,2,2
  c32[ c c c c] r16.
  r2
```

}



### *Barras que se atienen al pulso estrictamente*

Se puede hacer que las barras secundarias apunten en la dirección del pulso o fracción a que pertenecen. La primera barra evita los corchetes sueltos (que es el comportamiento predeterminado); la segunda barra sigue el pulso o fracción estrictamente.

```
\relative c'' {
  \time 6/8
  a8. a16 a a
  \set strictBeatBeaming = ##t
  a8. a16 a a
}
```



### *Símbolos de dirección y símbolos de agrupación de compás*

La agrupación de pulsos dentro de un compás está controlada por la propiedad de contexto `beatStructure`. Hay establecidos valores de `beatStructure` para muchos tipos de compases en `scm/time-signature-settings.scm`. Los valores de `beatStructure` se pueden cambiar o establecer con `\set`. Como alternativa, se puede usar `\time` para establecer tanto el compás como la estructura de pulsos. Para ello, especificamos la agrupación interna de los pulsos del compás como una lista de números (en la sintaxis de Scheme) antes de la indicación de compás.

`\time` se aplica al contexto `Timing`, por lo que no restablece los valores de `beatStructure` ni de `baseMoment` que se establezcan en otros contextos de nivel inferior, como `Voice`.

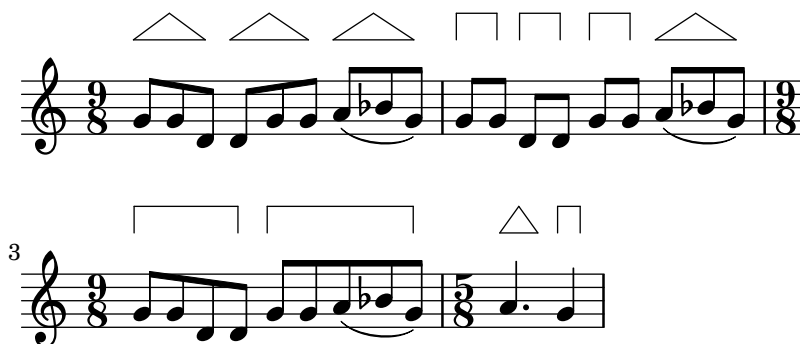
Si el grabador `Measure_grouping_engraver` está incluido en uno de los contextos de presentación, se imprimirán signos de agrupación de pulsos. Estos símbolos facilitan la lectura de música moderna rítmicamente compleja. En este ejemplo, el compás de 9/8 se agrupa según dos patrones distintos utilizando los dos métodos, mientras que el compás de 5/8 se agrupa de acuerdo con el ajuste predeterminado que está en `scm/time-signature-settings.scm`:

```
\score {
  \new Voice \relative c'' {
    \time 9/8
    g8 g d d g g a( bes g) |
    \set Timing.beatStructure = 2,2,2,3
    g8 g d d g g a( bes g) |
    \time 4,5 9/8
    g8 g d d g g a( bes g) |
    \time 5/8
    a4. g4 |
  }
}
```

```

\layout {
  \context {
    \Staff
    \consists "Measure_grouping_engraver"
  }
}

```



### *Finales de barra en el contexto Score*

Las reglas de final de barra especificadas en el contexto `Score` se aplican a todos los pentagramas, pero se pueden modificar tanto en los niveles de `Staff` como de `Voice`:

```

\relative c'' {
  \time 5/4
  % Set default beaming for all staves
  \set Score.baseMoment = #(ly:make-moment 1/8)
  \set Score.beatStructure = 3,4,3
  <<
    \new Staff {
      c8 c c c c c c c c c
    }
    \new Staff {
      % Modify beaming for just this staff
      \set Staff.beatStructure = 6,4
      c8 c c c c c c c c c
    }
    \new Staff {
      % Inherit beaming from Score context
      <<
        {
          \voiceOne
          c8 c c c c c c c c c
        }
        % Modify beaming for this voice only
        \new Voice {
          \voiceTwo
          \set Voice.beatStructure = 6,4
          a8 a a a a a a a a a
        }
      >>
    }
  >>
}
>>

```

}



## Véase también

Referencia de la notación: [Indicación de compás], página 70.

Archivos de inicio: `scm/time-signature-settings.scm`.

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Auto.beam engraver” en *Referencia de Funcionamiento Interno*, Sección “Beam” en *Referencia de Funcionamiento Interno*, Sección “Beam-ForbidEvent” en *Referencia de Funcionamiento Interno*, Sección “beam-interface” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

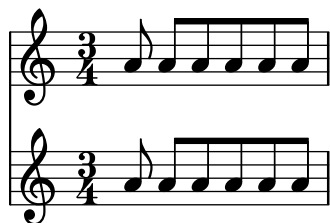
Si una partitura termina mientras una barra de corchea automática no ha terminado y aún acepta notas, esta última barra no se imprime en absoluto. Lo mismo sirve para las voces polifónicas introducidas con `<< ... \ \ ... >>`. Si una voz polifónica termina mientras una barra de corchea automática aún admite notas, no se imprime. El rodeo para estos problemas es aplicar el barrado manual a la última barra de la voz o partitura.

De forma predeterminada, el traductor `Timing` recibe el nombre del contexto `Score` como alias. Esto significa que el establecimiento del compás en una pauta afectará al barrado de las otras pautas también. Así, un ajuste en el compás en un pentagrama tardío reiniciará el barrado personalizado que se había ajustado en un pentagrama más temprano. Una forma de evitar este problema es ajustar la indicación de compás en un pentagrama solamente.

```
<<
  \new Staff {
    \time 3/4
    \set Timing.baseMoment = #(ly:make-moment 1/8)
    \set Timing.beatStructure = 1,5
    \set Timing.beamExceptions = #'()
    \repeat unfold 6 { a'8 }
  }
  \new Staff {
    \repeat unfold 6 { a'8 }
  }
```



&gt;&gt;



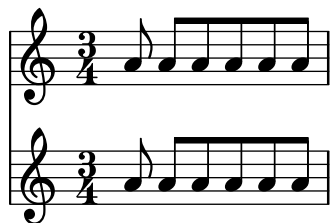
Los ajustes de barrado predeterminados para dicho compás también se pueden cambiar, de forma que siempre se utilice siempre el barrado deseado. La manera de efectuar cambios en los ajustes de barrado automático para una indicación de compás se describe en [Indicación de compás], página 70.

&lt;&lt;

```
\new Staff {
  \overrideTimeSignatureSettings
    3/4                % timeSignatureFraction
    1/8                % baseMomentFraction
    1,5                % beatStructure
    #'()               % beamExceptions
  \time 3/4
  \repeat unfold 6 { a'8 }
}

\new Staff {
  \time 3/4
  \repeat unfold 6 { a'8 }
}
```

&gt;&gt;



## Barras manuales

En ciertos casos puede ser preciso sobreescribir el algoritmo de barrado automático. Por ejemplo, el barrador automático no escribe barras por encima de los silencios o las líneas divisorias, y en las partituras corales el barrado se ajusta con frecuencia para que siga la medida de la letra en vez de la de las notas. Tales barras se especifican manualmente marcando los puntos de comienzo y final con [ y ]

$$\backslash \text{relative} \{ r_4 \ r_8 [ \ g' \ a \ r ] \ r \ g [ \ | \ a ] \ r \}$$


La dirección de las barras se puede establecer manualmente utilizando indicadores de dirección:

```
\relative { c''8^[ d e] c,_[ d e f g] }
```



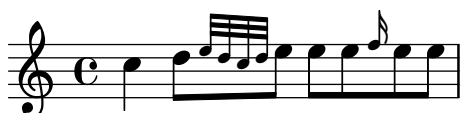
Se pueden marcar notas individuales con `\noBeam` para evitar que resulten unidas por una barra:

```
\relative {
  \time 2/4
  c''8 c\noBeam c c
}
```



Se pueden producir al mismo tiempo barras de notas de adorno y barras normales. Las notas de adorno sin barra no se colocan dentro de las barras de notas normales.

```
\relative {
  c''4 d8[
    \grace { e32 d c d }
    e8] e[ e
    \grace { f16 }
    e8 e]
}
```



Se puede conseguir un control incluso más estricto sobre las barras estableciendo las propiedades `stemLeftBeamCount` y `stemRightBeamCount`. Especifican el número de barras que se dibujarán en los lados izquierdo y derecho, respectivamente, de la nota siguiente. Si cualquiera de estas dos propiedades está ajustada a un valor, dicho valor se usará una sola vez, y luego se borrará. En este ejemplo, el último Fa se imprime con sólo una barra en el lado izquierdo, es decir, la barra de corchea del grupo como un todo.

```
\relative a' {
  a8[ r16 f g a]
  a8[ r16
  \set stemLeftBeamCount = #2
  \set stemRightBeamCount = #1
  f16
  \set stemLeftBeamCount = #1
  g16 a]
}
```



## Instrucciones predefinidas

`\noBeam.`

## Fragmentos de código seleccionados

### *Corchetes rectos y extremos de barra sueltos*

Son posibles tanto los corchetes rectos sobre notas sueltas como extremos de barra sueltos en figuras unidas, con una combinación de `stemLeftBeamCount`, `stemRightBeamCount` e indicadores de barra `[]` emparejados.

Para corchetes rectos que apunten a la derecha sobre notas sueltas, use indicadores de barra emparejados `[]` y establezca `stemLeftBeamCount` a cero (véase el ejemplo 1).

Para corchetes rectos que apunten a la izquierda, establezca en su lugar `stemRightBeamCount` (ejemplo 2).

Para extremos sueltos que apunten a la derecha al final de un conjunto de notas unidas, establezca `stemRightBeamCount` a un valor positivo. Y para extremos sueltos que apunten a la izquierda al principio de un conjunto de notas unidas, establezca `stemLeftBeamCount` en su lugar (ejemplo 3).

A veces, para una nota suelta rodeada de silencios tiene sentido que lleve los dos extremos sueltos del corchete plano, apuntando a derecha e izquierda. Hágalo solamente con indicadores de barra emparejados `[ ]` (ejemplo 4).

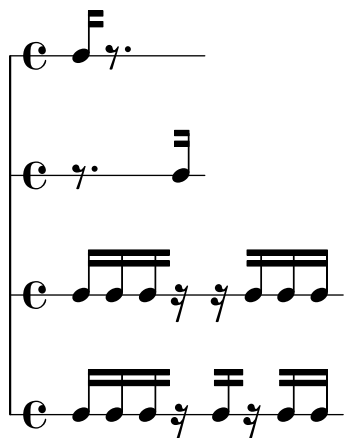
(Observe que `\set stemLeftBeamCount` siempre equivale a `\once \set`. En otras palabras, los ajustes de la cantidad de barras no se recuerdan, y por ello el par de corchetes planos aplicados a la nota Do semicorchea `c'16[]` del último ejemplo no tiene nada que ver con el `\set` de dos notas por detrás.)

```
\score {
  <<
    % Example 1
    \new RhythmicStaff {
      \set stemLeftBeamCount = #0
      c16[]
      r8.
    }
    % Example 2
    \new RhythmicStaff {
      r8.
      \set stemRightBeamCount = #0
      16[]
    }
    % Example 3
    \new RhythmicStaff {
      16 16
      \set stemRightBeamCount = #2
      16 r r
      \set stemLeftBeamCount = #2
      16 16 16
    }
    % Example 4
    \new RhythmicStaff {
      16 16
      \set stemRightBeamCount = #2
```

```

16 r16
16[]
r16
\set stemLeftBeamCount = #2
16 16
}
>>
}

```



## Véase también

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 661, [Notas de adorno], página 120.

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Beam” en *Referencia de Funcionamiento Interno*, Sección “BeamEvent” en *Referencia de Funcionamiento Interno*, Sección “Beam\_engraver” en *Referencia de Funcionamiento Interno*, Sección “beam-interface” en *Referencia de Funcionamiento Interno*, Sección “Stem\_engraver” en *Referencia de Funcionamiento Interno*.

## Barras progresivas

Las barras progresivas se usan para indicar que un pequeño grupo de notas se debe tocar a una velocidad creciente (o decreciente), sin cambiar el tempo general de la pieza. El ámbito de la barra progresiva se debe indicar manualmente usando [ y ], el efecto de ángulo de la barra se inicia especificando una dirección en la propiedad **grow-direction** del objeto **Beam**.

Si queremos que la colocación de las notas y el sonido de la salida MIDI refleje el *ritardando* o *accelerando* indicado por la barra progresiva, las notas se deben agrupar como una expresión musical delimitada por llaves y precedida de una instrucción **featheredDurations** que especifica la razón entre las duraciones de la primera y la última notas dentro del grupo.

Los corchetes rectos muestran el ámbito de la barra y las llaves muestran qué notas han de modificar sus duraciones. Normalmente delimitarían el mismo grupo de notas, pero no es un requisito: las dos instrucciones son independientes.

En el ejemplo siguiente las ocho semicorcheas ocupan exactamente el mismo tiempo que una blanca, pero la primera nota tiene la mitad de duración que la última, con las notas intermedias alargándose gradualmente. Las cuatro primeras fusas se aceleran gradualmente, mientras que las últimas cuatro fusas están a un tempo constante.

```

\relative c' {
  \override Beam.grow-direction = #LEFT

```

```

\featherDurations #(ly:make-moment 2/1)
{ c16[ c c c c c c c c] }
\override Beam.grow-direction = #RIGHT
\featherDurations #(ly:make-moment 2/3)
{ c32[ d e f] }
% revert to non-feathered beams
\override Beam.grow-direction = #'()
{ g32[ a b c] }
}

```



El espaciado en la salida impresa representa las duraciones de las notas de una forma sólo aproximada, pero la salida MIDI es exacta.

## Instrucciones predefinidas

`\featherDurations.`

## Véase también

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

## Advertencias y problemas conocidos

La instrucción `\featherDurations` sólo funciona con fragmentos musicales muy breves, y cuando los números de la fracción son pequeños.

### 1.2.5 Compases

#### Barras de compás

Las líneas divisorias delimitan a los compases pero también se pueden usar para indicar las repeticiones. Normalmente, las líneas divisorias normales se insertan de manera automática en la salida impresa en lugares que están basados en el compás actual.

Las barras de compás sencillas insertadas automáticamente se pueden cambiar por otros tipos con la instrucción `\bar`. Por ejemplo, se suele poner una doble barra de cierre al final de la pieza:

```
\relative { e'4 d c2 \bar "||." }
```



No deja de ser válida la última nota de un compás si no termina sobre la línea divisoria automática: se supone que la nota se prolonga sobre el compás siguiente. Pero una secuencia larga de dichos compases prolongados puede hacer que la música aparezca comprimida o incluso que se salga de la página. Esto es a causa de que los saltos de línea automáticos solamente se producen al final de compases completos, es decir, cuando todas las notas han finalizado antes de que el compás termine.

**Nota:** Una duración incorrecta puede hacer que se trate de evitar la producción de saltos de línea, dando como resultado una línea de música con una compresión exagerada o música que se sale de la página.

También se permiten saltos de línea en las barras insertadas manualmente incluso dentro de compases incompletos. Para permitir un salto de línea donde no hay ninguna línea divisoria visible, utilice lo siguiente:

```
\bar ""
```

De esta manera se insertará una barra invisible de compás y se hará posible el salto de línea en este punto, sin forzarlo. No se incrementa el contador de los números de compás. Para forzar un salto de línea, consulte Sección 4.3.1 [Saltos de línea], página 582.

Esta y otras líneas divisorias especiales se pueden insertar manualmente en cualquier punto. Cuando coinciden con el final de un compás, sustituyen a la línea divisoria simple que se habría insertado automáticamente en dicho lugar. Cuando no coinciden con el final de un compás, se inserta la línea especificada en ese punto en la salida impresa.

Observe que las líneas divisorias manuales son puramente visuales. No afectan a ninguna de las propiedades que una barra normal afectaría, como a los números de compás, alteraciones, saltos de línea, etc. No afectan tampoco al cálculo y colocación de las líneas divisorias subsiguientes. Cuando una divisoria se coloca manualmente donde ya existe una divisoria normal, los efectos de la línea original no se alteran.

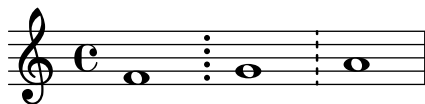
Están disponibles para su inserción manual dos tipos de líneas divisorias simples y cinco tipos de doble barra:

```
\relative {
  f'1 \bar "|"
  f1 \bar "."
  g1 \bar "||"
  a1 \bar ".|"
  b1 \bar ".."
  c1 \bar "|.|"
  d1 \bar "|."
  e1
}
```



así como la barra de puntos y la discontinua:

```
\relative {
  f'1 \bar ";;"
  g1 \bar "!"
  a1
}
```



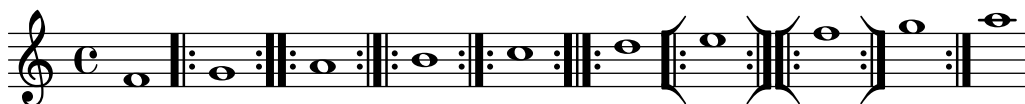
y nueve tipos de barra de repetición:

```
\relative {
  f'1 \bar ".|:"
  g1 \bar ":\.:"
  a1 \bar ":\.|"
  b1 \bar ":\.:"
  c1 \bar ":\.|"
}
```

```

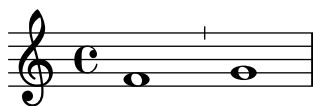
d1 \bar "[|:"
e1 \bar "[:|]"
f1 \bar "[:|]"
g1 \bar "[:|]"
a1
}

```



Además se puede imprimir una línea divisoria como una marca corta:

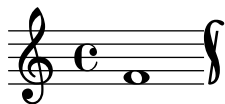
```
f'1 \bar "'" g'1
```



Sin embargo, dado que las mencionadas marcas se utilizan usualmente en el canto gregoriano, es preferible en este caso utilizar `\divisioMinima`, que se describe en la sección [Divisiones], página 478, dentro de Canto gregoriano.

LilyPond contempla la notación del canto kievano y ofrece una línea divisoria especial kievana:

```
f'1 \bar "k"
```



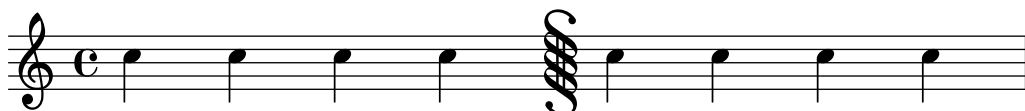
Pueden verse más detalles de esta notación explicados en Sección 2.9.5 [Tipografiado del canto kievano en notación cuadrada], página 486.

Para los símbolos de segno en línea, existen tres tipos de barras de compás que se diferencian en su comportamiento en los saltos de línea:

```

\relative c'' {
  c4 c c c
  \bar "S-||"
  c4 c c c \break
  \bar "S-||"
  c4 c c c
  \bar "S"
  c4 c c c \break
  \bar "S"
  c4 c c c
  \bar "S-S"
  c4 c c c \break
  \bar "S-S"
  c1
}

```

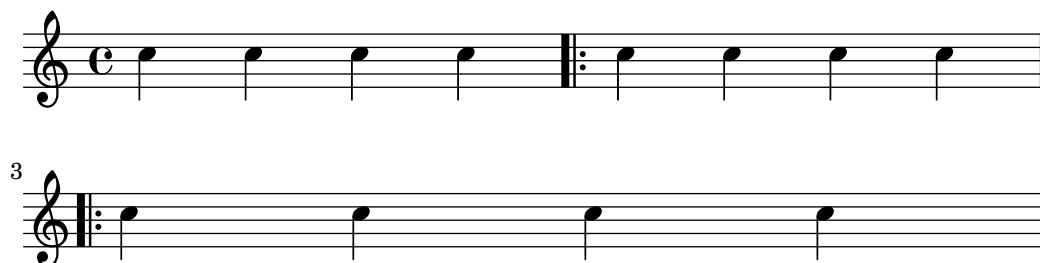




Aunque se pueden insertar manualmente barras de compás con significado de repeticiones, no se reconocen como repeticiones por parte de LilyPond. Las secciones repetidas se introducen mejor utilizando las diversas instrucciones de repetición (véase Sección 1.4 [Repeticiones], página 158), que imprimen automáticamente las barras correspondientes.

Además se puede especificar ".|:-||", que equivale a ".|:" excepto en los saltos de línea, en que produce una doble barra al final de la línea y una repetición izquierda (de comienzo) al principio de la línea siguiente.

```
\relative c' {
  c4 c c c
  \bar ".|:-||"
  c4 c c c \break
  \bar ".|:-||"
  c4 c c c
}
```



Para combinaciones de repeticiones con el símbolo de segno, existen seis variantes diferentes:

```
\relative c' {
  c4 c c c
  \bar " :|.S"
  c4 c c c \break
  \bar " :|.S"
  c4 c c c
  \bar " :|.S-S"
  c4 c c c \break
  \bar " :|.S-S"
  c4 c c c
  \bar "S.|:-S"
  c4 c c c \break
  \bar "S.|:-S"
  c4 c c c
  \bar "S.|"
  c4 c c c \break
}
```



```

\bar "S. | : "
c4 c c c
\bar " : | . S. | : "
c4 c c c \break
\bar " : | . S. | : "
c4 c c c
\bar " : | . S. | : - S "
c4 c c c \break
\bar " : | . S. | : - S "
c1
}

```



Además, existe una instrucción `\inStaffSegno` que crea una barra de compás con símbolo de segno, situada en conjunción con una línea de repetición adecuada si se utiliza con una instrucción `\repeat volta`, véase [Repeticiones normales], página 159.

Se pueden definir tipos nuevos de líneas divisorias con `\defineBarLine`:

```
\defineBarLine tipo_de_barra #'(final comienzo extensión)
```

Además de *bartype* (la cadena de caracteres que se usará después para referirnos a la nueva línea divisoria), admite tres valores: los dos primeros determinan la apariencia de la línea divisoria cuando se produce en el salto de línea, en cuyo caso los glifos dados primero y segundo se

imprimen respectivamente al *final* del sistema y al *principio* del siguiente. El tercer glifo proporcionado solamente es de relevancia en sistemas de más de un pentagrama (véase [Agrupar pentagramas], página 202), donde se usa como una barra de extensión, que se imprime entre los pentagramas.

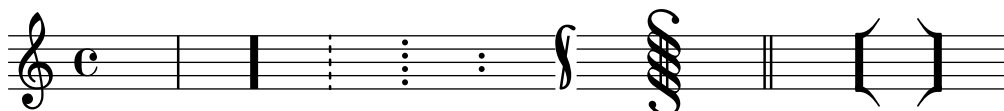
Las variables de `\defineBarLine` pueden incluir la cadena de caracteres ‘vacía’ `""`, que equivale a imprimir una línea divisoria invisible. También se puede establecer al valor falso `#f` que no imprime línea divisoria alguna.

Después de la definición, la nueva línea divisoria se puede utilizar mediante `\bar tipo_de_barra`.

Actualmente están disponibles diez elementos de barra de compás:

```
\defineBarLine ":" #'(":" "")
\defineBarLine "=" #'(=" " "" "")
\defineBarLine "[" #'("[" "" "")
\defineBarLine "]" #'("]" "" "")
```

```
\new Staff {
  s1 \bar "|"
  s1 \bar "."
  s1 \bar "!"
  s1 \bar ";"
  s1 \bar ":"
  s1 \bar "k"
  s1 \bar "S"
  s1 \bar "="
  s1 \bar "["
  s1 \bar "]"
  s1 \bar ""
}
```



La línea divisoria `"="` proporciona la barra de doble extensión, usada en combinación con el símbolo de segno. No se recomienda usarla como una doble línea divisoria fina aislada; para ello es casi siempre preferible `\bar "||"`.

El signo `"-"` inicia anotaciones a las barras de compás, que son útiles para distinguir entre aquellas que tienen idéntica apariencia pero distinto comportamiento en los saltos de línea y/o diferentes barras de extensión. La par que sigue al signo `"-"` no se usa para construir la barra de compás.

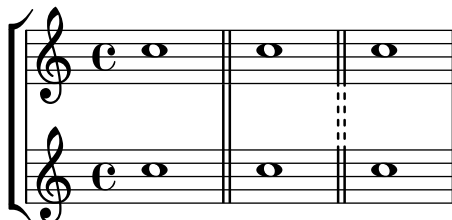
```
\defineBarLine "||-dashedSpan" #'("||" "" "!!")
```

```
\new StaffGroup <<
  \new Staff \relative c'' {
    c1 \bar "||"
    c1 \bar "||-dashedSpan"
    c1
  }
  \new Staff \relative c'' {
    c1
    c1
  }
```

```

      c1
    }
  >>

```



Además, el carácter del espacio en blanco " " sirve como contenedor para definir barras de extensión correctamente alineadas a las barras principales:

```

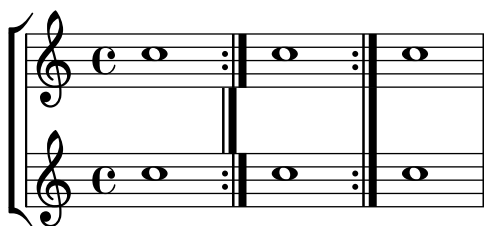
\defineBarLine ":|.-wrong" #'(" :|. " " |.")
\defineBarLine ":|.-right" #'(" :|. " " " |.")

```

```

\new StaffGroup <<
  \new Staff \relative c'' {
    c1 \bar ":|.-wrong"
    c1 \bar ":|.-right"
    c1
  }
  \new Staff \relative c'' {
    c1
    c1
    c1
  }
>>

```



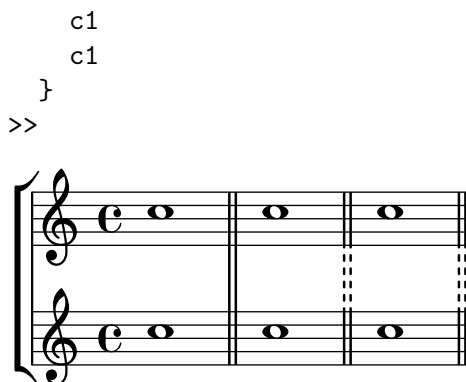
Los nuevos tipos de línea divisoria definidos mediante el uso de `\defineBarLine` pueden, a su vez, utilizarse dentro de una *segunda* definición de línea divisoria. Tales definiciones ‘anidadas’ hacen posible usar glifos personalizados en lugares en los que no sería posible hacerlo de otra forma, como en el extremo final de los sistemas:

```

\defineBarLine "||-dashEverywhere" #'("!!" "!!" "!!")
\defineBarLine "||-advancedDashSpan" #'("||-dashEverywhere" " " "!!")

\new StaffGroup <<
  \new Staff \relative c'' {
    c1 \bar "||"
    c1 \bar "||-advancedDashSpan"
    c1 \bar "||-advancedDashSpan"
  }
  \new Staff \relative c'' {
    c1

```



Si se necesitan elementos adicionales, LilyPond provee una forma sencilla de definirlos. Para ver más información sobre cómo modificar o añadir barras de compás, consulte el archivo `scm/bar-line.scm`.

En las partituras con muchos pentagramas, una instrucción `\bar` en uno de ellos se aplica automáticamente a todos los demás. Las líneas resultantes se conectan entre los distintos pentagramas de un `StaffGroup`, `PianoStaff` o `GrandStaff`.

```

<<
  \new StaffGroup <<
    \new Staff \relative {
      e'4 d
      \bar "||"
      f4 e
    }
    \new Staff \relative { \clef bass c'4 g e g }
  >>
  \new Staff \relative { \clef bass c'2 c2 }
>>

```



La instrucción `'\bar tipo de barra'` es una forma corta de hacer `'\set Timing.whichBar = tipo de barra'`. Cuando `whichBar` se establece con el valor de una cadena, se crea una línea divisoria de ese tipo.

El tipo de barra predeterminado que se usa para las líneas divisorias insertadas automáticamente es `"|"`. Se puede cambiar en cualquier momento con `'\set Timing.defaultBarType = tipo de barra'`.

## Véase también

Referencia de la notación: Sección 4.3.1 [Saltos de línea], página 582, Sección 1.4 [Repeticiones], página 158, [Agrupar pentagramas], página 202.

Archivos instalados: `scm/bar-line.scm`.

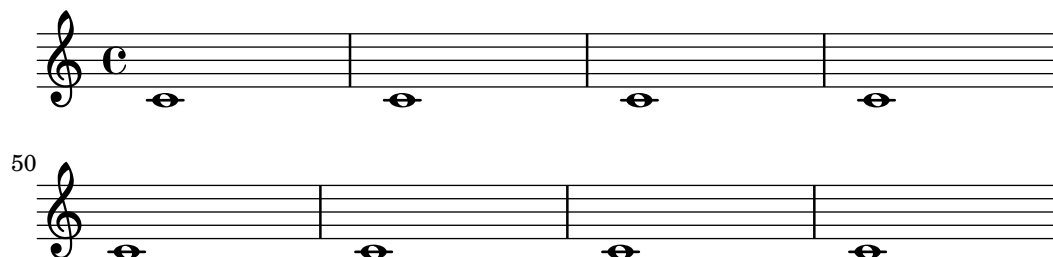
Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “BarLine” en *Referencia de Funcionamiento Interno* (creada al nivel de **Staff** (pentagrama)), Sección “SpanBar” en *Referencia de Funcionamiento Interno* (a través de los pentagramas), Sección “Timing\_translator” en *Referencia de Funcionamiento Interno* (para las propiedades del contador de tiempo Timing).

## Numeración de compases

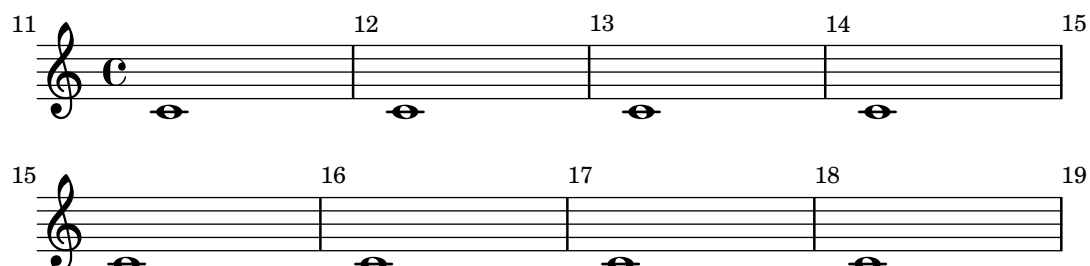
Por defecto, los números de compás se imprimen al principio de la línea, excepto la primera. El número propiamente dicho se almacena en la propiedad **currentBarNumber**, que normalmente se actualiza automáticamente para cada compás. También se puede establecer manualmente:

```
\relative c' {
  c1 c c c
  \break
  \set Score.currentBarNumber = #50
  c1 c c c
}
```



Los números de compás se pueden tipografiar a intervalos regulares en vez de al principio de cada línea. Para hacerlo se debe sobrescribir el comportamiento predeterminado de forma que se puedan imprimir en otros lugares. Esto viene controlado por la propiedad **break-visibility** de **BarNumber**. Toma tres valores que se pueden fijar al valor **#t** o **#f** para especificar si el número de compás correspondiente es visible o no. El orden de los tres valores es **visible al final de la línea, visible en mitad de la línea, visible al principio de la línea**. En el ejemplo siguiente se imprimen los números de compás en todos los sitios posibles:

```
\relative c' {
  \override Score.BarNumber.break-visibility = ##(#t #t #t)
  \set Score.currentBarNumber = #11
  % Permit first bar number to be printed
  \bar ""
  c1 | c | c | c |
  \break
  c1 | c | c | c |
}
```



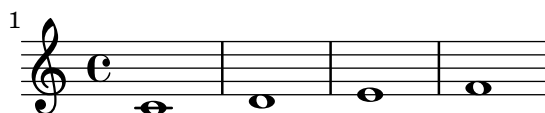
## Fragmentos de código seleccionados

### *Imprimir el número de compás en el primer compás*

De forma predeterminada se suprime el número del primer compás de una partitura si es menor o igual a '1'. Al establecer el valor de `barNumberVisibility` a `all-bar-numbers-visible`, se puede imprimir cualquier número de compás para el primer compás y todos los siguientes. Observe que, para que esto funcione, se debe insertar una línea divisoria vacía antes de la primera nota.

```
\layout {
  indent = 0
  ragged-right = ##t
}

\relative c' {
  \set Score.barNumberVisibility = #all-bar-numbers-visible
  \bar ""
  c1 | d | e | f \break
  g1 | e | d | c
}
```



### *Imprimir números de compás a intervalos regulares*

Se pueden imprimir los números de compás a intervalos regulares mediante el establecimiento de la propiedad `barNumberVisibility`. Aquí los números de compás se imprimen a cada dos compases excepto al final de la línea.

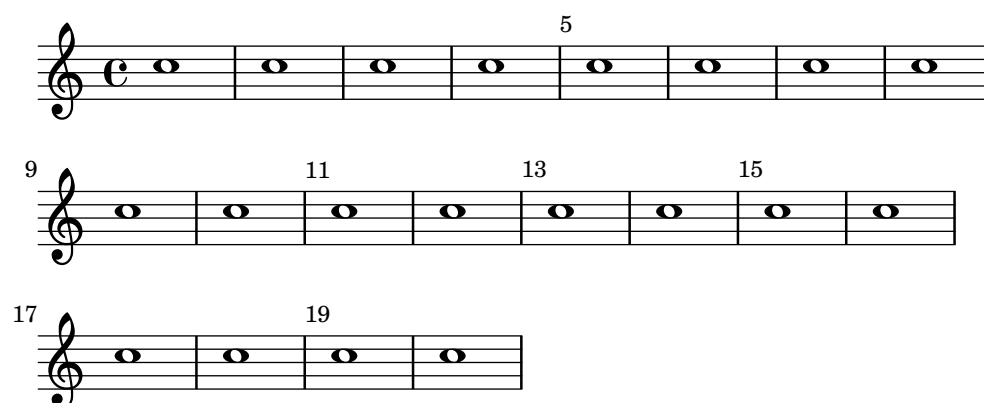
```
\relative c' {
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \set Score.currentBarNumber = #11
  % Permit first bar number to be printed
  \bar ""
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
  c1 | c | c | c | c
  \break
  c1 | c | c | c | c
}
```



### *Printing bar numbers with changing regular intervals*

Using the `set-bar-number-visibility` context function, bar number intervals can be changed.

```
\relative c' {
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \context Score \applyContext #(set-bar-number-visibility 4)
  \repeat unfold 10 c'1
  \context Score \applyContext #(set-bar-number-visibility 2)
  \repeat unfold 10 c
}
```



### *Imprimir números de compás dentro de rectángulos o circunferencias*

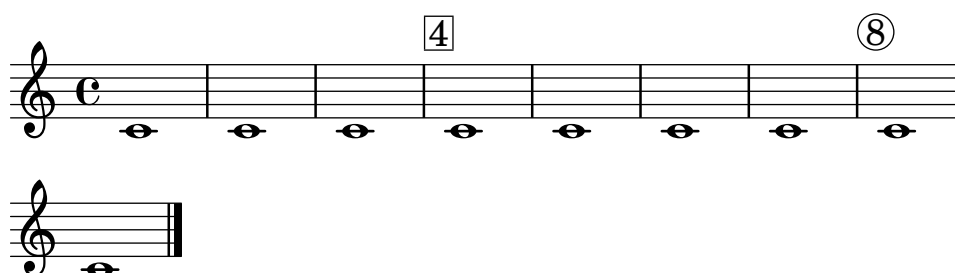
Los números de compás también se pueden imprimir dentro de rectángulos o de circunferencias.

```
\relative c' {
  % Prevent bar numbers at the end of a line and permit them elsewhere
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 4)

  % Increase the size of the bar number by 2
  \override Score.BarNumber.font-size = #2

  % Draw a box round the following bar number(s)
  \override Score.BarNumber.stencil
    = #(make-stencil-boxer 0.1 0.25 ly:text-interface::print)
  \repeat unfold 5 { c1 }

  % Draw a circle round the following bar number(s)
  \override Score.BarNumber.stencil
    = #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
  \repeat unfold 4 { c1 } \bar "|."
}
```



### *Numeración de compases alternativa*

Se pueden seleccionar dos métodos alternativos para la numeración de compases, especiales para cuando hay repeticiones.

```
\relative c'{
  \set Score.alternativeNumberingStyle = #'numbers
  \repeat volta 3 { c4 d e f | }
  \alternative {
    { c4 d e f | c2 d \break }
    { f4 g a b | f4 g a b | f2 a | \break }
    { c4 d e f | c2 d }
  }
  c1 \break
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \repeat volta 3 { c,4 d e f | }
  \alternative {
    { c4 d e f | c2 d \break }
    { f4 g a b | f4 g a b | f2 a | \break }
    { c4 d e f | c2 d }
  }
  }
c1
}
```

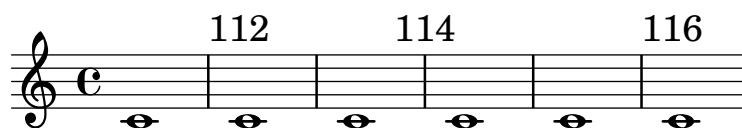
The image displays six staves of musical notation, each illustrating a different alternative numbering style for repeated measures. The notation is in treble clef with a common time signature (C). The first staff, labeled '1.', shows a first ending with a repeat sign and a first ending bracket. The second staff, labeled '2', shows a second ending with a repeat sign and a second ending bracket. The third staff, labeled '2', shows a third ending with a repeat sign and a third ending bracket. The fourth staff, labeled '5', shows a first ending with a repeat sign and a first ending bracket. The fifth staff, labeled '6b', shows a second ending with a repeat sign and a second ending bracket. The sixth staff, labeled '6c', shows a third ending with a repeat sign and a third ending bracket.



### *Alineación de los números de compás*

Los números de compás se alinean de forma predeterminada por la derecha con su objeto padre. Éste es por lo general el borde izquierdo de una línea o, si los números se imprimen dentro de la línea, el lado izquierdo de una línea divisoria. Los números también se pueden situar directamente sobre la barra de compás o alineados por la izquierda con ella.

```
\relative c' {
  \set Score.currentBarNumber = #111
  \override Score.BarNumber.break-visibility = #all-visible
  % Increase the size of the bar number by 2
  \override Score.BarNumber.font-size = #2
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
  c1 | c1
  % Center-align bar numbers
  \override Score.BarNumber.self-alignment-X = #CENTER
  c1 | c1
  % Left-align bar numbers
  \override Score.BarNumber.self-alignment-X = #LEFT
  c1 | c1
}
```



### *Suprimir los números de compás de toda la partitura*

Se pueden eliminar completamente los números de compás quitando el grabador `Bar_number_engraver` del contexto de `Score`.

```
\layout {
  \context {
    \Score
    \omit BarNumber
    % or:
    %\remove "Bar_number_engraver"
  }
}

\relative c'' {
  c4 c c c \break
  c4 c c c
}
```



## Véase también

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “BarNumber” en *Referencia de Funcionamiento Interno*, Sección “Bar\_number\_engraver” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Los números de compás pueden colisionar con el corchete `StaffGroup`, si hay uno en la parte de arriba. Para solucionarlo, se puede usar la propiedad de relleno `padding` de `BarNumber` para colocar el número correctamente. Consulte Sección “StaffGroup” en *Referencia de Funcionamiento Interno* y Sección “BarNumber” en *Referencia de Funcionamiento Interno* para ver más información.

## Comprobación de compás y de número de compás

Las comprobaciones de compás ayudan a detectar errores en las duraciones. Una comprobación de compás se escribe usando el símbolo de la barra vertical, `|`, en cualquier lugar donde se espera que caiga una línea divisoria. Si se encuentran líneas de comprobación de compás en otros lugares, se imprime una lista de advertencias en el archivo log de registro, mostrando los números de línea y columna en que han fallado las comprobaciones de compás. En el siguiente ejemplo, la segunda comprobación de compás avisará de un error.

```
\time 3/4 c2 e4 | g2 |
```

Una duración incorrecta produce una partitura completamente desbaratada, especialmente si la partitura es polifónica, de manera que la mejor forma de empezar a corregir la entrada es buscar sistemáticamente la existencia de comprobaciones de compás fallidas y duraciones incorrectas.

Si se producen varias comprobaciones de compás seguidas por valor de la misma duración musical, solo aparece el primer mensaje de advertencia. Así se consigue que el mensaje esté concentrado en la fuente del error de pulso.

Las comprobaciones de compás también se pueden insertar dentro de la letra de las canciones:

```
\lyricmode {
  \time 2/4
  Twin -- kle | Twin -- kle |
}
```

Observe que las comprobaciones de compás dentro de la letra de las canciones se evalúan en el momento musical de la sílaba *siguiente* a la comprobación de compás que se procesa. Si la letra está asociada con las notas de una voz que tiene un silencio al principio del compás, no se puede poner ninguna sílaba al comienzo de ese compás y se imprime un mensaje de advertencia si se escribe una comprobación de compás dentro de la letra, en esa posición.

También es posible redefinir la acción que se lleva a cabo cuando se encuentra una barra vertical o símbolo de comprobación de compás, `|`, en el código de entrada, de forma que haga algo distinto a una comprobación de compás. Se hace asignando una expresión musical a `"|"`, el símbolo de barra vertical. En el ejemplo siguiente se establece `|` de forma que inserte una doble línea divisoria cuando aparece en el código de entrada, en vez de comprobar el final de un compás.

```
"|" = \bar "||"
{
  c'2 c' |
  c'2 c'
  c'2 | c'
  c'2 c'
```

}



Al copiar piezas musicales grandes puede servir de ayuda comprobar que el número de compás de LilyPond corresponde al original desde el que está copiando. Esto se puede comprobar con `\barNumberCheck`, por ejemplo:

```
\barNumberCheck #123
```

imprime una advertencia si el `currentBarNumber` (número del compás actual) no es el 123 en el momento de la compilación.

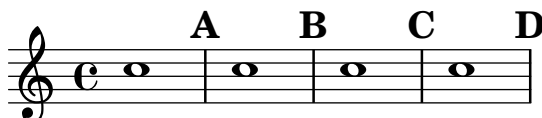
## Véase también

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

## Llamadas de ensayo

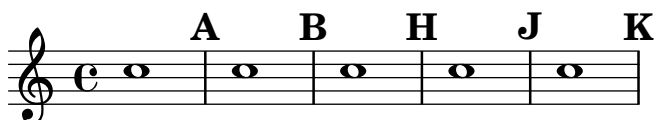
Para imprimir una letra de ensayo, utilice la orden `\mark`.

```
\relative c'' {
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
}
```



La letra de ensayo se incrementa automáticamente si usa `\mark \default`, pero también puede utilizar un número entero como argumento para establecer la indicación manualmente. El valor que se utilizará se almacena dentro de la propiedad `rehearsalMark`.

```
\relative c'' {
  c1 \mark \default
  c1 \mark \default
  c1 \mark #8
  c1 \mark \default
  c1 \mark \default
}
```



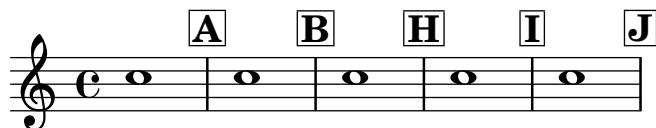
La letra ‘I’ se salta de acuerdo con las tradiciones de grabado. Si quiere incluir la letra ‘I’, use una de las instrucciones siguientes, según el estilo de marcas de ensayo que desee (sólo letras, letras dentro de un rectángulo o letras dentro de un círculo).

```
\set Score.markFormatter = #format-mark-alphabet
\set Score.markFormatter = #format-mark-box-alphabet
\set Score.markFormatter = #format-mark-circle-alphabet
\relative c'' {
```

```

\set Score.markFormatter = #format-mark-box-alphabet
c1 \mark \default
c1 \mark \default
c1 \mark #8
c1 \mark \default
c1 \mark \default
}

```

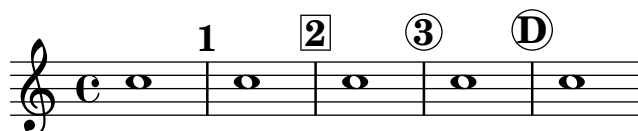


El estilo se define por medio de la propiedad `markFormatter`. Es una función que toma como argumentos la marca en curso (un número entero) y el contexto actual. Debe devolver un objeto de marcado. En el ejemplo siguiente, se establece `markFormatter` con el valor de un procedimiento enlatado. Después de algunos compases, se establece con el valor de una función que produce un número encerrado en una caja.

```

\relative c'' {
  \set Score.markFormatter = #format-mark-numbers
  c1 \mark \default
  c1 \mark \default
  \set Score.markFormatter = #format-mark-box-numbers
  c1 \mark \default
  \set Score.markFormatter = #format-mark-circle-numbers
  c1 \mark \default
  \set Score.markFormatter = #format-mark-circle-letters
  c1
}

```



El archivo `scm/translation-functions.scm` contiene las definiciones de `format-mark-letters` (el formato por omisión), `format-mark-box-letters`, `format-mark-numbers` y `format-mark-box-numbers`. Se pueden usar éstos como inspiración para otras funciones de formateo.

Podemos utilizar `format-mark-barnumbers`, `format-mark-box-barnumbers` y `format-mark-circle-barnumbers` para obtener números de compás en lugar de números o letras secuenciales.

Se pueden especificar otros estilos de letra de ensayo de forma manual:

```
\mark "A1"
```

Observe que `Score.markFormatter` no afecta a las marcas que se especifican de esta forma. Sin embargo, es posible aplicar un elemento `\markup` a la cadena.

```
\mark \markup { \box A1 }
```

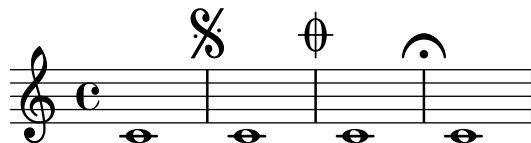
Los glifos musicales (como el segno) se pueden imprimir dentro de un elemento `\mark`

```

\relative c' {
  c1 \mark \markup { \musicglyph "scripts.segno" }
  c1 \mark \markup { \musicglyph "scripts.coda" }
}

```

```
c1 \mark \markup { \musicglyph "scripts.ufermata" }
c1
}
```



Consulte Sección A.8 [La tipografía Emmentaler], página 713, para ver una lista de los símbolos que se pueden imprimir con `\musicglyph`.

Para ver formas comunes de trucar la colocación de las letras de ensayo, consulte Sección 1.8.2 [Formatear el texto], página 264. Para un control más preciso, consulte `break-alignable-interface` en la sección Sección 5.5.1 [Alineación de objetos], página 678.

El archivo `scm/translation-functions.scm` contiene las definiciones de `format-mark-numbers` y de `format-mark-letters`. Se pueden utilizar como inspiración para escribir otras funciones de formateo.

## Véase también

Referencia de la notación: Sección A.8 [La tipografía Emmentaler], página 713, Sección 1.8.2 [Formatear el texto], página 264, Sección 5.5.1 [Alineación de objetos], página 678.

Archivos de inicio: `scm/translation-functions.scm`.

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “MarkEvent” en *Referencia de Funcionamiento Interno*, Sección “Mark-engraver” en *Referencia de Funcionamiento Interno*, Sección “RehearsalMark” en *Referencia de Funcionamiento Interno*.

## 1.2.6 Asuntos rítmicos especiales

### Notas de adorno

Los mordentes y notas de adorno son ornamentos musicales, impresos en un tipo más pequeño y no ocupan ningún tiempo lógico adicional en el compás.

```
\relative {
  c''4 \grace b16 a4(
    \grace { b16 c16 } a2)
}
```



Hay otros tres tipos posibles de notas de adorno; la *acciaccatura* o mordente de una nota (una nota de adorno sin medida que se indica mediante una nota ligada y con el corchete tachado) y la *appoggiatura*, que toma una fracción fija de la nota principal a la que se adjunta, y que se imprime sin tachar. Es posible escribir una nota de adorno con la plica tachada, como la *acciaccatura* pero sin la ligadura, como para colocarla entre notas que están ligadas entre sí, utilizando la función `\slashedGrace`.

```
\relative {
  \acciaccatura d''8 c4
  \appoggiatura e8 d4
  \acciaccatura { g16 f } e2
}
```

```

\slashedGrace a,8 g4
\slashedGrace b16 a4(
\slashedGrace b8 a2)
}

```



La colocación de notas de adorno se sincroniza entre los distintos pentagramas. En el siguiente ejemplo, hay dos semicorcheas de adorno por cada corchea de adorno:

```

<<
\new Staff \relative { e''2 \grace { c16 d e f } e2 }
\new Staff \relative { c''2 \grace { g8 b } c2 }
>>

```

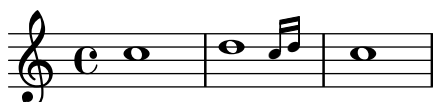


Si queremos terminar una nota con un adorno, usamos la instrucción `\afterGrace`. Toma dos argumentos: la nota principal, y las notas de adorno que siguen a la nota principal.

```

\relative { c''1 \afterGrace d1 { c16[ d] } c1 }

```



Esto sitúa las notas de adorno *después* del comienzo de la nota principal. El instante temporal en que se colocan las notas de adorno es una fracción dada de la duración de la nota principal. El ajuste predeterminado de

```

afterGraceFraction = 3/4

```

se puede redefinir en el nivel jerárquico superior. Las instrucciones `\afterGrace` individuales pueden, en vez de esto, llevar especificada la fracción directamente, después de la propia instrucción.

El siguiente ejemplo muestra el resultado de establecer el espacio a su valor predeterminado, establecerlo a 15/16, y finalmente a 1/2 de la nota principal.

```

<<
\new Staff \relative {
  c''1 \afterGrace d1 { c16[ d] } c1
}
\new Staff \relative {
  c''1 \afterGrace 15/16 d1 { c16[ d] } c1
}
\new Staff \relative {
  c''1 \afterGrace 1/2 d1 { c16[ d] } c1
}
>>

```

&gt;&gt;



El efecto de `\afterGrace` también se puede obtener usando espaciadores. El ejemplo siguiente sitúa la nota de adorno después de un espacio que dura 7/8 de la nota principal.

```
\new Voice \relative {
  <<
    { d''1^\trill_( }
    { s2 s4. \grace { c16 d } }
  >>
  c1)
}
```



Una expresión musical `\grace` introduce ajustes de tipografía especiales, por ejemplo para producir un tipo más pequeño y para fijar las direcciones. Por ello, cuando se introducen trucos para la presentación, deben ir dentro de la expresión de adorno. Las sobreescrituras se deben también revertir dentro de la expresión de adorno. Aquí, la dirección predeterminada de la plica de la nota de adorno se sobreescribe y luego se revierte.

```
\new Voice \relative {
  \acciaccatura {
    \stemDown
    f''16->
    \stemNeutral
  }
  g4 e c2
}
```



## Fragmentos de código seleccionados

### *Utilizar la barra que tacha las notas de adorno con notas normales*

Es posible aplicar la barrita que cruza la barra de las acciaccaturas, en otras situaciones.

```
\relative c'' {
  \override Flag.stroke-style = #"grace"
  c8( d2) e8( f4)
```

}



### *Trucar la disposición de las notas de adorno dentro de la música*

La disposición de las expresiones de adorno se puede cambiar a lo largo de toda la música usando las funciones `add-grace-property` y `remove-grace-property`. El ejemplo siguiente borra la definición de la dirección de la plica para esta nota de adorno, de manera que las plicas no siempre apuntan hacia arriba, y cambia la forma predeterminada de las cabezas a aspas.

```
\relative c'' {
  \new Staff {
    $(remove-grace-property 'Voice 'Stem 'direction)
    $(add-grace-property 'Voice 'NoteHead 'style 'cross)
    \new Voice {
      \acciaccatura { f16 } g4
      \grace { d16 e } f4
      \appoggiatura { f,32 g a } e2
    }
  }
}
```



### *Redefinición de los valores globales predeterminados para notas de adorno*

Los valores predeterminados para las notas de adorno están almacenados en los identificadores `startGraceMusic`, `stopGraceMusic`, `startAcciaccaturaMusic`, `stopAcciaccaturaMusic`, `startAppoggiaturaMusic` y `stopAppoggiaturaMusic`, que están definidos en el archivo `ly/grace-init.ly`. Redefiniéndolos se pueden obtener otros efectos.

```
startAcciaccaturaMusic = {
  <>(
    \override Flag.stroke-style = #"grace"
    \slurDashed
  )
}

stopAcciaccaturaMusic = {
  \revert Flag.stroke-style
  \slurSolid
  <>
}

\relative c'' {
  \acciaccatura d8 c1
}
```





## *Posicionamiento de las notas de adorno con espacio flotante*

Al establecer la propiedad '`strict-grace-spacing`' hacemos que las columnas musicales para las notas de adorno sean 'flotantes', es decir, desacopladas de las notas que no son de adorno: primero se aplica el espaciado de las notas normales, y luego se ponen las columnas musicales de las notas de adorno a la izquierda de las columnas musicales de las notas principales.

```
\relative c'' {
  <<
    \override Score.SpacingSpanner.strict-grace-spacing = ##t
    \new Staff \new Voice {
      \afterGrace c4 { c16[ c8 c16] }
      c8[ \grace { b16 d } c8]
      c4 r
    }
    \new Staff {
      c16 c c c c c c c c4 r
    }
  >>
}
```



## Véase también

Glosario musical: Sección “*grace notes*” en *Glosario Musical*, Sección “*acciaccatura*” en *Glosario Musical*, Sección “*appoggiatura*” en *Glosario Musical*.

Referencia de la notación: [Escarlar las duraciones], página 56, [Barras manuales], página 100.

Archivos de inicio: `ly/grace-init.ly`.

Fragmentos de código: Sección “*Rhythms*” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “*GraceMusic*” en *Referencia de Funcionamiento Interno*, Sección “*Grace\_beam\_engraver*” en *Referencia de Funcionamiento Interno*, Sección “*Grace\_auto\_beam\_engraver*” en *Referencia de Funcionamiento Interno*, Sección “*Grace\_engraver*” en *Referencia de Funcionamiento Interno*, Sección “*Grace\_spacing\_engraver*” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Una *acciaccatura* de varias notas con una barra se imprime sin tachar, y tiene exactamente la misma apariencia que una *appoggiatura* de varias notas con barra.

La sincronización de las notas de adorno también puede acarrear sorpresas. La notación de pentagramas, como armaduras, líneas divisorias, etc., se sincronizan también. Ponga cuidado cuando mezcle pentagramas con adornos y sin adornos, por ejemplo

```
<<
  \new Staff \relative { e''4 \bar ".|:" \grace c16 d2. }
  \new Staff \relative { c''4 \bar ".|:" d2. }
```

&gt;&gt;



Esto se puede remediar insertando desplazamientos de adorno de las duraciones correspondientes en los otros pentagramas. Para el ejemplo anterior

&lt;&lt;

```
\new Staff \relative { e''4 \bar ".|:" \grace c16 d2. }
\new Staff \relative { c''4 \bar ".|:" \grace s16 d2. }
```

&gt;&gt;



Es obligatorio usar la instrucción `\grace` para la parte de los desplazamientos, incluso si la parte visual usa `\acciaccatura` o `\appoggiatura` porque en caso contrario se imprime una fea ligadura que conecta a la nota de adorno invisible con la nota siguiente.

Las secciones de adorno sólo se deben usar dentro de expresiones de música secuenciales. No están contemplados el anidado ni la yuxtaposición de secciones de adorno, y podría producir caídas u otros errores.

Cada nota de adorno en la salida MIDI tiene una longitud que es 1/4 de su duración real. Si la duración combinada de las notas de adorno es mayor que la longitud de la nota precedente, se genera un error “Retrocediendo en el tiempo MIDI”. Tiene dos opciones: en primer lugar, puede hacer más corta la duración de las notas de adorno, por ejemplo:

```
c'8 \acciaccatura { c'8[ d' e' f' g'] }
```

se convierte en:

```
c'8 \acciaccatura { c'16[ d' e' f' g'] }
```

Otra opción es cambiar explícitamente la duración musical:

```
c'8 \acciaccatura { \scaleDurations 1/2 { c'8[ d' e' f' g'] } }
```

Véase [Escalar las duraciones], página 56.

## Alinear con una cadenza

En un contexto orquestal, las cadenzas presentan un problema especial: al construir una partitura que tiene una cadenza, todos los demás instrumentos deben saltar tantas notas como la longitud de la cadenza, pues en caso contrario empezarán demasiado pronto o demasiado tarde.

Una solución a este problema son las funciones `mmrest-of-length` y `skip-of-length`. Estas funciones de Scheme toman un fragmento de música como argumento y generan un `\skip` o silencio multicompa, de la longitud exacta del fragmento.

```
MyCadenza = \relative {
  c'4 d8 e f g g4
```

```

    f2 g4 g
  }

  \new GrandStaff <<
    \new Staff {
      \MyCadenza c'1
      \MyCadenza c'1
    }
    \new Staff {
      #(mmrest-of-length MyCadenza)
      c'1
      #(skip-of-length MyCadenza)
      c'1
    }
  >>

```



## Véase también

Glosario musical: Sección “cadenza” en *Glosario Musical*.

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

## Gestión del tiempo

El tiempo está administrado por el `Timing_translator`, que de forma predeterminada vive en el contexto de `Score`. Se añade un alias, `Timing`, al contexto en que se coloca el `Timing_translator`. Para asegurar que está disponible el alias `Timing`, quizá tenga que crear explícitamente una instancia del contexto contenedor (como `Voice` o `Staff`).

Se usan las siguientes propiedades de `Timing` para seguir la pista del tiempo dentro de la partitura.

### `currentBarNumber`

El número de compás en curso. Para ver un ejemplo que muestra el uso de esta propiedad, consulte [Numeración de compases], página 112.

### `measureLength`

La longitud de los compases dentro de la indicación actual de compás. Para un 4/4 esto es 1, y para el 6/8 es 3/4. Su valor determina cuándo se insertan las líneas divisorias y cómo se generan las barras automáticas.

### `measurePosition`

El punto en que nos encontramos dentro del compás. Esta cantidad se reinicia sustrayendo `measureLength` cada vez que se alcanza o se excede `measureLength`. Cuando eso ocurre, se incrementa `currentBarNumber`.

### `timing`

Si tiene un valor verdadero, las variables anteriores se actualizan a cada paso de tiempo. Cuando tiene un valor falso, el grabador se queda en el compás actual indefinidamente.

La cuenta del tiempo se puede cambiar estableciendo el valor de cualquiera de estas variables explícitamente. En el siguiente ejemplo, se imprime la indicación de compás predeterminada 4/4, pero `measureLength` tiene está ajustado a 5/4. En los 4/8 hasta el tercer compás, la posición `measurePosition` se adelanta en 1/8 hasta 5/8, acortando ese compás en 1/8. Entonces, la siguiente línea divisoria cae en 9/8 en vez de hacerlo en 5/4.

```
\new Voice \relative {
  \set Timing.measureLength = #(ly:make-moment 5/4)
  c'1 c4 |
  c1 c4 |
  c4 c
  \set Timing.measurePosition = #(ly:make-moment 5/8)
  b4 b b8 |
  c4 c1 |
}
```



Como lo ilustra el ejemplo, `ly:make-moment n/m` construye una duración de la fracción  $n/m$  de una redonda. Por ejemplo, `ly:make-moment 1/8` es una duración de una corchea y `ly:make-moment 7/16` es la duración de siete semicorcheas.

## Véase también

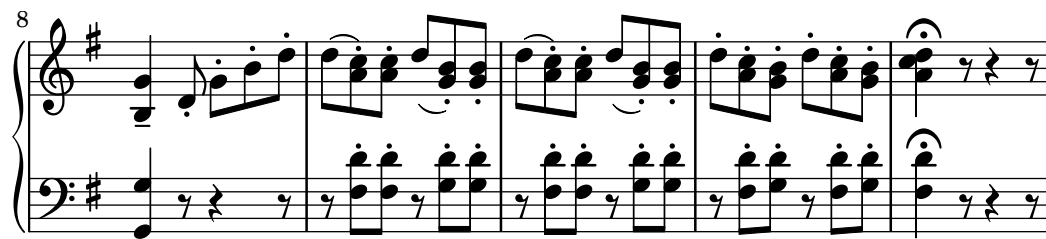
Referencia de la notación: [Numeración de compases], página 112, [Música sin compasear], página 79.

Fragmentos de código: Sección “Rhythms” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Timing\_translator” en *Referencia de Funcionamiento Interno*, Sección “Score” en *Referencia de Funcionamiento Interno*.

## 1.3 Expresiones

RONDO  
*Allegro*



Esta sección relaciona diversas marcas de expresión que se pueden crear en una partitura.

### 1.3.1 Expresiones adosadas a las notas

Esta sección explica cómo crear marcas expresivas que están aplicadas a notas: articulaciones, ornamentos y matices. También se tratan los métodos para crear las nuevas marcas dinámicas.

#### Articulaciones y ornamentos

Se pueden adjuntar a las notas un amplio abanico de símbolos para denotar articulaciones, adornos y otras indicaciones de ejecución, utilizando la sintaxis siguiente:

`nota\nombre`

La lista de los posibles valores de *nombre* está en Sección A.14 [Lista de articulaciones], página 800. Por ejemplo:

```
\relative {
  c''4\staccato c\mordent b2\turn
  c1\fermata
}
```



Algunas de estas articulaciones tienen abreviaturas que facilitan su escritura. Las abreviaturas se escriben detrás del nombre de la nota, y su sintaxis consiste en un guión - seguido de un símbolo que especifica la articulación. Existen abreviaturas predefinidas para el *marcato*, *stopped* (nota apagada), *tenuto*, *staccatissimo*, *accent* (acento), *staccato* (picado), y *portato*. La salida correspondiente a estas articulaciones aparece de la siguiente manera:

```
\relative {
  c''4-^ c-+ c-- c-!
  c4-> c-. c2-_
}
```



Las reglas para la colocación predeterminada de las articulaciones se encuentran definidas en el archivo `scm/script.scm`. Las articulaciones y ornamentos se pueden colocar manualmente por encima o por debajo del pentagrama; véase Sección 5.4.2 [Dirección y posición], página 661.

Las articulaciones son objetos `Script`. Sus propiedades se describen de forma más completa en Sección “Script” en *Referencia de Funcionamiento Interno*.

Se pueden aplicar articulaciones a los silencios normales y multi-compás, además de a las notas. La aplicación de una articulación a un silencio multi-compás crea un objeto `MultiMeasureRestScript`.

```
\override Script.color = #red
\override MultiMeasureRestScript.color = #blue
a'2\fermata r\fermata
```

R1\fermata



Además de articulaciones, se pueden adjuntar textos y elementos de marcado a las notas. Véase [Guiones de texto], página 257.

Para ver más información acerca de la ordenación de los elementos `Script` y `TextScript` que se adjuntan a las notas, consulte Sección “Colocación de los objetos” en *Manual de Aprendizaje*.

## Fragmentos de código seleccionados

### *Modificar los valores predeterminados para la notación abreviada de las articulaciones*

Las abreviaturas se encuentran definidas dentro del archivo `ly/script-init.ly`, donde las variables `dashHat`, `dashPlus`, `dashDash`, `dashBang`, `dashLarger`, `dashDot` y `dashUnderscore` reciben valores predeterminados. Se pueden modificar estos valores predeterminados para las abreviaturas. Por ejemplo, para asociar la abreviatura `--+` (`dashPlus`) con el símbolo del semitrino en lugar del símbolo predeterminado `+`, asigne el valor `trill` a la variable `dashPlus`:

```
\relative c'' { c1--+ }
```

```
dashPlus = "trill"
```

```
\relative c'' { c1--+ }
```



### *Controlar la ordenación vertical de las inscripciones*

El orden vertical que ocupan las inscripciones gráficas está controlado con la propiedad `'script-priority`. Cuanto más bajo es este número, más cerca de la nota se colocará. En este ejemplo, el `TextScript` (el sostenido) tiene primero la prioridad más baja, por lo que se sitúa en la posición más baja en el primer ejemplo. En el segundo, el `Script` (el semitrino) es el que la tiene más baja, por lo que se sitúa en la parte interior. Cuando dos objetos tienen la misma prioridad, el orden en que se introducen determina cuál será el que aparece en primer lugar.

```
\relative c''' {
  \once \override TextScript.script-priority = #-100
  a2^\prall^\markup { \sharp }
```

```

  \once \override Script.script-priority = #-100
  a2^\prall^\markup { \sharp }
```

}



### Crear un grupeto de anticipación

La creación de un grupeto circular de anticipación entre dos notas, donde la nota inferior del grupeto utiliza una alteración, requiere varias sobreescrituras de propiedades. La propiedad `outside-staff-priority` se debe establecer al valor `#f`, pues en caso contrario tendría prioridad sobre la propiedad `avoid-slur property`. Cambiando las fracciones  $2/3$  y  $1/3$  puede ajustarse la posición horizontal.

```
\relative c'' {
  c2*2/3 ( s2*1/3\turn d4) r
  <<
    { c4.( d8) }
    { s4 s\turn }
  >>
  \transpose c d \relative c'' <<
    { c4.( d8) }
    {
      s4
      \once \set suggestAccidentals = ##t
      \once \override AccidentalSuggestion.outside-staff-priority = ##f
      \once \override AccidentalSuggestion.avoid-slur = #'inside
      \once \override AccidentalSuggestion.font-size = -3
      \once \override AccidentalSuggestion.script-priority = -1
      \single \hideNotes
      b8-\turn \noBeam
      s8
    }
  >>
}
```



### Véase también

Glosario musical: Sección “tenuto” en *Glosario Musical*, Sección “accent” en *Glosario Musical*, Sección “staccato” en *Glosario Musical*, Sección “portato” en *Glosario Musical*.

Manual de aprendizaje: Sección “Colocación de los objetos” en *Manual de Aprendizaje*.

Referencia de la notación: [Guiones de texto], página 257, Sección 5.4.2 [Dirección y posición], página 661, Sección A.14 [Lista de articulaciones], página 800, [Trinos], página 155.

Archivos instalados: `scm/script.scm`.

Fragmentos de código: Sección “Expressive marks” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Script” en *Referencia de Funcionamiento Interno*, Sección “TextScript” en *Referencia de Funcionamiento Interno*.

## Matrices dinámicas

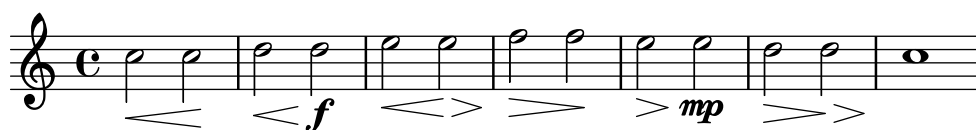
Las marcas dinámicas de matiz absoluto se especifican usando una instrucción después de una nota: `c4\ff`. Las marcas dinámicas disponibles son `\ppppp`, `\pppp`, `\ppp`, `\pp`, `\p`, `\mp`, `\mf`, `\f`, `\ff`, `\fff`, `\ffff`, `\fffff`, `\fp`, `\sf`, `\sff`, `\sp`, `\spp`, `\sfz`, `\rfz` y `\n`. Las indicaciones dinámicas se pueden colocar manualmente por encima o por debajo del pentagrama; para ver más detalles, consulte Sección 5.4.2 [Dirección y posición], página 661.

```
\relative c'' {
  c2\ppp c\mp
  c2\rfz c^\mf
  c2_\spp c^\ff
}
```



Una indicación de *crescendo* se comienza con `\<` y se termina con `\!`, con un matiz absoluto o con otra indicación de crescendo o de decrescendo. Una indicación de *decrescendo* comienza con `\>` y se termina también con `\!`, con un matiz dinámico absoluto o con otra indicación de crescendo o de decrescendo. Se pueden usar `\cr` y `\decr` en lugar de `\<` y `\>`. También es posible usar `\endcr` y `\enddecr` en lugar de `\!` para finalizar un crescendo o un decrescendo, respectivamente. De forma predeterminada, se tipografían reguladores en ángulo cuando se utiliza esta notación.

```
\relative c'' {
  c2\< c\!
  d2\< d\f
  e2\< e\>
  f2\> f\!
  e2\> e\mp
  d2\> d\>
  c1\!
}
```



Un regulador terminado mediante `\!` finaliza en el borde derecho de la nota que lleva el `\!` adosado. En el caso en que esté terminado con el comienzo de otra indicación *crescendo* o *decrescendo*, terminará en el centro de la nota que tiene adosada la siguiente indicación `\<` ó `\>`. El regulador siguiente empezará entonces en el borde derecho de la misma nota en lugar del borde izquierdo, como sería usual si hubiese terminado con `\!` previamente. Si un regulador termina en el inicio de un compás, finalizará sobre la línea divisoria precedente.

```
\relative {
  c''1\< | c4 a c\< a | c4 a c\! a\< | c4 a c a\!
}
```





Los reguladores que terminan con indicaciones dinámicas absolutas en lugar de con `\!` también se tipografían de manera parecida. Sin embargo, la longitud de la propia indicación dinámica puede alterar el punto en que finaliza el regulador anterior.

```
\relative {
  c'1\< | c4 a c\mf a | c1\< | c4 a c\ffff a
}
```



Se requieren silencios espaciadores para tipografiar varias indicaciones dinámicas sobre una sola nota. Esto es útil especialmente para añadir un *crescendo* y un *decrescendo* a la misma nota:

```
\relative {
  c'4\< c\! d\> e\!
  << f1 { s4 s4\< s4\> s4\! } >>
}
```



Se puede usar la instrucción `\espressivo` para indicar un crescendo y un decrescendo sobre la misma nota. Sin embargo, advierta que esta indicación está implementada como una articulación, no como una expresión dinámica.

```
\relative {
  c'2 b4 a
  g1\espressivo
}
```



Las indicaciones de crescendo textuales se inician con `\cresc.` Los decrescendos textuales se inician con `\decresc` o con `\dim.` Se trazan líneas extensoras cuando es necesario.

```
\relative {
  g'8\cresc a b c b c d e\mf |
  f8\decresc e d c e\> d c b |
  a1\dim ~ |
  a2. r4\! |
}
```



También pueden sustituirse los reguladores gráficos por cambios de dinámica textuales:

```
\relative c' {
```

```

\crescTextCresc
c4\< d e f\! |
\dimTextDecresc
g4\> e d c\! |
\dimTextDecr
e4\> d c b\! |
\dimTextDim
d4\> c b a\! |
\crescHairpin
\dimHairpin
c4\< d\! e\> d\! |
}

```



Para crear nuevas indicaciones de matiz absoluto o de texto que deba alinearse con los matices; véase [Indicaciones dinámicas nuevas], página 137.

La colocación vertical de las indicaciones de dinámica se maneja por parte de Sección “DynamicLineSpanner” en *Referencia de Funcionamiento Interno*.

Está disponible un contexto `Dynamics` para tipografiar las indicaciones de matiz dinámico en su propia línea horizontal. Utilice silencios de separación para indicar los tiempos (las notas que están dentro de un contexto `Dynamics` también ocupan tiempo musical, pero no se imprimen). El contexto `Dynamics` puede contener otros elementos como inscripciones textuales, objetos extensores de texto e indicaciones de pedal de piano.

```

<<
\new Staff \relative {
  c'2 d4 e |
  c4 e e,2 |
  g'4 a g a |
  c1 |
}
\new Dynamics {
  s1\< |
  s1\f |
  s2\dim s2-"rit." |
  s1\p |
}
>>

```



## Instrucciones predefinidas

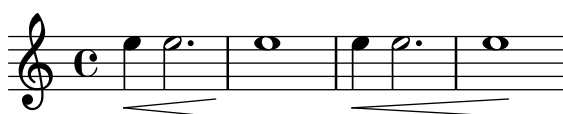
`\dynamicUp`, `\dynamicDown`, `\dynamicNeutral`, `\crescTextCresc`, `\dimTextDim`, `\dimTextDecr`, `\dimTextDecresc`, `\crescHairpin`, `\dimHairpin`.

## Fragmentos de código seleccionados

### *Establecer el comportamiento de los reguladores en las barras de compás*

Si la nota que da fin a un regulador cae sobre la primera parte de un compás, el regulador se detiene en la línea divisoria inmediatamente precedente. Se puede controlar este comportamiento sobrescribiendo la propiedad `'to-barline`.

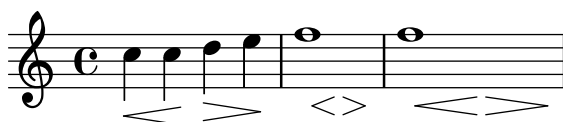
```
\relative c'' {
  e4\< e2.
  e1\!
  \override Hairpin.to-barline = ##f
  e4\< e2.
  e1\!
}
```



### *Ajustar la longitud mínima de los reguladores*

Si los reguladores son demasiado cortos, se pueden alargar modificando la propiedad `minimum-length` del objeto `Hairpin`.

```
\relative c'' {
  c4\< c\! d\> e\!
  << f1 { s4 s\< s\> s\! } >>
  \override Hairpin.minimum-length = #5
  << f1 { s4 s\< s\> s\! } >>
}
```

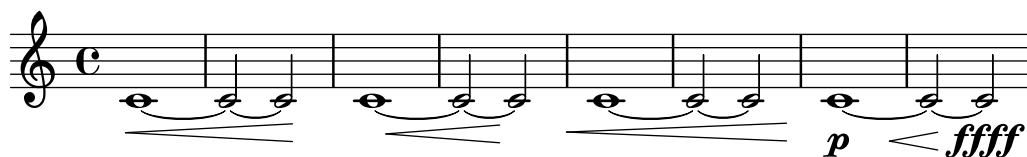


### *Moving the ends of hairpins*

The ends of hairpins may be offset by setting the `shorten-pair` property of the `Hairpin` object. Positive values move endpoints to the right, negative to the left. Unlike the `minimum-length` property, this property only affects the appearance of the hairpin; it does not adjust horizontal spacing (including the position of bounding dynamics). This method is thus suitable for fine-tuning a hairpin within its allotted space.

```
{
  c'1~\<
  c'2~ c'\!
  \once \override Hairpin.shorten-pair = #'(2 . 2)
  c'1~\<
  c'2~ c'\!
  \once \override Hairpin.shorten-pair = #'(-2 . -2)
  c'1~\<
  c'2~ c'\!
  c'1~\p-\tweak shorten-pair #'(2 . 0)\<
  c'2~ c'\ffff
}
```

}



### *Impresión de reguladores utilizando la notación «al niente»*

Se pueden imprimir reguladores con un círculo en la punta (notación «al niente») estableciendo la propiedad `circled-tip` del objeto `Hairpin` al valor `#t`.

```
\relative c'' {
  \override Hairpin.circled-tip = ##t
  c2\< c\!
  c4\> c\< c2\!
}
```



### *Imprimir reguladores en diversos estilos*

Los reguladores se pueden crear en una amplia variedad de estilos.

```
\relative c'' {
  \override Hairpin.stencil = #flared-hairpin
  a4\< a a a\f
  a4\p\< a a a\ff
  a4\sفز\< a a a\!
  \override Hairpin.stencil = #constante-hairpin
  a4\< a a a\f
  a4\p\< a a a\ff
  a4\sفز\< a a a\!
  \override Hairpin.stencil = #flared-hairpin
  a4\> a a a\f
  a4\p\> a a a\ff
  a4\sفز\> a a a\!
  \override Hairpin.stencil = #constante-hairpin
  a4\> a a a\f
  a4\p\> a a a\ff
  a4\sفز\> a a a\!
}
```



### *Indicaciones dinámicas y textuales alineadas verticalmente*

Todos los objetos `DynamicLineSpanner` (reguladores e indicaciones dinámicas de texto) se sitúan con su línea de referencia a una distancia de al menos `'staff-padding` del pentagrama, a no ser que otros elementos de notación los fuerce a colocarse a más distancia. Si se ajusta `'staff-padding` a un valor suficientemente grande, las indicaciones dinámicas quedarán alineadas.

Se usa una idea similar junto a `\textLengthOn` para alinear las inscripciones de texto a lo largo de su línea de base.

```
music = \relative c' {
  a'2\p b\f
  e4\p f\f\> g, b\p
  c2^\markup { \huge gorgeous } c^\markup { \huge fantastic }
}

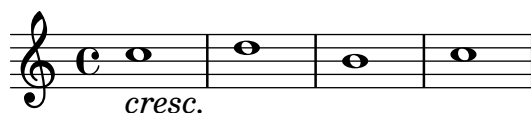
{
  \music
  \break
  \override DynamicLineSpanner.staff-padding = #3
  \textLengthOn
  \override TextScript.staff-padding = #1
  \music
}
```



### *Ocultar la línea de extensión de las expresiones textuales de dinámica*

Los cambios de dinámica con estilo de texto (como *cresc.* y *dim.*) se imprimen con una línea intermitente que muestra su alcance. Esta línea se puede suprimir de la siguiente manera:

```
\relative c'' {
  \override DynamicTextSpanner.style = #'none
  \crescTextCresc
  c1\< | d | b | c\!
}
```



## ***Cambiar el texto y los estilos de objeto de extensión para las indicaciones dinámicas textuales***

Se puede modificar el texto empleado para los crescendos y decrescendos modificando las propiedades de contexto `crescendoText` y `decrescendoText`.

El estilo de la línea de extensión se puede cambiar modificando la propiedad `'style` de `DynamicTextSpanner`. El valor predeterminado es `'dashed-line` (línea discontinua), y entre otros valores posibles se encuentran `'line` (línea), `'dotted-line` (línea de puntos) y `'none` (nada):

```
\relative c'' {
  \set crescendoText = \markup { \italic { cresc. poco } }
  \set crescendoSpanner = #'text
  \override DynamicTextSpanner.style = #'dotted-line
  a2\< a
  a2 a
  a2 a
  a2 a\mf
}
```



## **Véase también**

Glosario musical: Sección “al niente” en *Glosario Musical*, Sección “crescendo” en *Glosario Musical*, Sección “decrescendo” en *Glosario Musical*, Sección “hairpin” en *Glosario Musical*.

Manual de aprendizaje: Sección “Articulaciones y matices dinámicos” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 661, [Indicaciones dinámicas nuevas], página 137, Sección 3.5.9 [Enriquecimiento de la salida MIDI], página 564, Sección 3.5.4 [Control de las dinámicas del MIDI], página 555.

Fragmentos de código: Sección “Expressive marks” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “DynamicText” en *Referencia de Funcionamiento Interno*, Sección “Hairpin” en *Referencia de Funcionamiento Interno*, Sección “DynamicLineSpanner” en *Referencia de Funcionamiento Interno*, Sección “Dynamics” en *Referencia de Funcionamiento Interno*.

## **Indicaciones dinámicas nuevas**

La manera más fácil de crear indicaciones dinámicas es usar objetos de marcado (`\markup`).

```
moltoF = \markup { molto \dynamic f }

\relative {
  <d' e>16_\moltoF <d e>
  <d e>2..
}
```



En el modo de marcado se pueden crear indicaciones dinámicas editoriales (entre paréntesis o corchetes). La sintaxis del modo de marcado se describe en Sección 1.8.2 [Formatear el texto], página 264.

```
roundF = \markup {
  \center-align \concat { \bold { \italic ( }
    \dynamic f \bold { \italic ) } } }
boxF = \markup { \bracket { \dynamic f } }
\relative {
  c'1_\roundF
  c1_\boxF
}
```



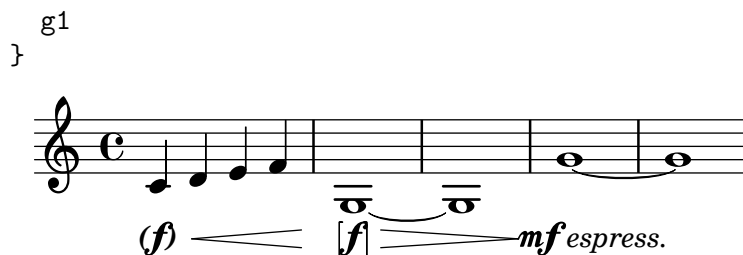
Las indicaciones dinámicas sencillas y centradas se crean fácilmente con la función `make-dynamic-script`.

```
sfzp = #(make-dynamic-script "sfzp")
\relative {
  c'4 c c\sfzp c
}
```



En general `make-dynamic-script` toma cualquier objeto de marcado como argumento. La fuente tipográfica de matices sólo contiene los caracteres **f**, **m**, **p**, **r**, **s** y **z**, por lo que si se desea obtener una indicación dinámica que incluya texto normal o signos de puntuación, es necesario utilizar instrucciones de marcado que devuelvan los ajustes de la familia de fuente tipográfica y su codificación a las del texto normal, por ejemplo `\normal-text`. El interés de la utilización de `make-dynamic-script` en lugar de un elemento de marcado corriente está en asegurar la alineación vertical de los objetos de marcado y reguladores que se aplican a la misma cabeza de nota.

```
roundF = \markup { \center-align \concat {
  \normal-text { \bold { \italic ( } }
  \dynamic f
  \normal-text { \bold { \italic ) } } } }
boxF = \markup { \bracket { \dynamic f } }
mfEspress = \markup { \center-align \line {
  \hspace #3.7 mf \normal-text \italic espress. } }
roundFdynamic = #(make-dynamic-script roundF)
boxFdynamic = #(make-dynamic-script boxF)
mfEspressDynamic = #(make-dynamic-script mfEspress)
\relative {
  c'4_\roundFdynamic\< d e f
  g,1~_\boxFdynamic\>
  g1
  g'1~\mfEspressDynamic
```



Se puede utilizar en su lugar la forma Scheme del modo de marcado. Su sintaxis se explica en Sección “Construcción de elementos de marcado en Scheme” en *Extender*.

```
moltoF = #(make-dynamic-script
            (markup #:normal-text "molto"
                  #:dynamic "f"))

\relative {
  <d' e>16 <d e>
  <d e>2..\moltoF
}
```



Para alinear el texto del matiz dinámico a la izquierda en lugar de centrarlo sobre una nota, utilice un `\tweak`:

```
moltoF = \tweak DynamicText.self-alignment-X #LEFT
          #(make-dynamic-script
            (markup #:normal-text "molto"
                  #:dynamic "f"))

\relative {
  <d' e>16 <d e>
  <d e>2..\moltoF <d e>1
}
```



Los ajustes para las fuentes tipográficas en el modo de marcado se describen en [Seleccionar la tipografía y su tamaño], página 266.

## Véase también

Referencia de la notación: Sección 1.8.2 [Formatear el texto], página 264, [Seleccionar la tipografía y su tamaño], página 266, Sección 3.5.9 [Enriquecimiento de la salida MIDI], página 564, Sección 3.5.4 [Control de las dinámicas del MIDI], página 555.

Extender LilyPond: Sección “Construcción de elementos de marcado en Scheme” en *Extender*.

Fragmentos de código: Sección “Expressive marks” en *Fragmentos de código*.

### 1.3.2 Expresiones como curvas

Esta sección explica cómo crear varias marcas expresivas de forma curva: ligaduras de expresión y de fraseo, respiraciones, caídas y elevaciones de tono.



## Ligaduras de expresión

Las *Ligaduras de expresión* se introducen utilizando paréntesis:

**Nota:** En música polifónica, las ligaduras de expresión deben terminar en la misma voz en la que empiezan.

```
\relative {
  f''4( g a) a8 b(
  a4 g2 f4)
  <c e>2( <b d>2)
}
```



Las ligaduras de expresión se pueden colocar manualmente por encima o por debajo de las notas; véase Sección 5.4.2 [Dirección y posición], página 661.

Debe tenerse cuidado con las ligaduras de expresión simultáneas o superpuestas. En casi todos los casos, las ligaduras de expresión externas indican, ciertamente, fraseo, y las ligaduras de fraseo pueden superponerse a las ligaduras de expresión normales, véase [Ligaduras de fraseo], página 143. Cuando se necesitan varias ligaduras de expresión normales dentro de una sola voz, los comienzos y finales correspondientes de las ligaduras deben etiquetarse precediéndolos por `\=` seguido de una clave identificativa (un símbolo o un entero no negativo).

```
\fixed c' {
  <c~ f\=1( g\=2( >2 <c e\=1) a\=2) >
}
```



Las ligaduras de expresión pueden ser continuas, discontinuas o de puntos. El estilo predefinido de las ligaduras de expresión es el continuo:

```
\relative {
  c'4( e g2)
  \slurDashed
  g4( e c2)
  \slurDotted
  c4( e g2)
  \slurSolid
  g4( e c2)
}
```



También se puede hacer que las ligaduras de expresión sean semi-discontinuas (la primera mitad discontinua y la segunda mitad continua) o semi-continuas (la primera mitad continua y la segunda discontinua):

```
\relative {
```

```

c'4( e g2)
\slurHalfDashed
g4( e c2)
\slurHalfSolid
c4( e g2)
\slurSolid
g4( e c2)
}

```



Se pueden definir patrones de discontinuidad para las ligaduras de expresión:

```

\relative {
  c'4( e g2)
  \slurDashPattern #0.7 #0.75
  g4( e c2)
  \slurDashPattern #0.5 #2.0
  c4( e g2)
  \slurSolid
  g4( e c2)
}

```



## Instrucciones predefinidas

`\slurUp`, `\slurDown`, `\slurNeutral`, `\slurDashed`, `\slurDotted`, `\slurHalfDashed`, `\slurHalfSolid`, `\slurDashPattern`, `\slurSolid`.

## Fragmentos de código seleccionados

### *Utilizar ligaduras dobles para acordes legato*

Algunos compositores escriben dos ligaduras cuando quieren acordes legato. Esto se puede conseguir estableciendo `doubleSlurs`.

```

\relative c' {
  \set doubleSlurs = ##t
  <c e>4( <d f> <c e> <d f>)
}

```



### *Situar los elementos de marcado de texto por dentro de las ligaduras*

Los elementos de marcado de texto deben tener la propiedad `outside-staff-priority` establecida al valor falso para que se impriman por dentro de las ligaduras de expresión.

```

\relative c'' {

```

```

\override TextScript.avoid-slur = #'inside
\override TextScript.outside-staff-priority = ##f
c2(^\markup { \halign #-10 \natural } d4.) c8
}

```



### *Hacer ligaduras de expresión con estructura compleja de discontinuidad*

Las ligaduras de expresión se pueden construir con patrones de discontinuidad complejos mediante la definición de la propiedad `dash-definition`. `dash-definition` es una lista de elementos de discontinuidad. Un elemento de discontinuidad es una lista de parámetros que definen el comportamiento de discontinuidad de un segmento de la ligadura de expresión.

La ligadura se define en términos del parámetro de bezier `t` cuyo rango va de 0 a la izquierda de la ligadura hasta 1 en el extremo derecho de la ligadura. Cada elemento de discontinuidad es una lista (`t-inicio t-final fracción-discontinuidad período-discontinuidad`). La región de la ligadura desde `t-inicio` hasta `t-final` tendrá una fracción `fracción-discontinuidad` de cada `período-discontinuidad` de color negro. `período-discontinuidad` se define en términos de espacios de pentagrama. `fracción-discontinuidad` se establece al valor de 1 para una ligadura continua.

```

\relative c' {
  \once \override
    Slur.dash-definition = #'((0 0.3 0.1 0.75)
                              (0.3 0.6 1 1)
                              (0.65 1.0 0.4 0.75))

  c4( d e f)
  \once \override
    Slur.dash-definition = #'((0 0.25 1 1)
                              (0.3 0.7 0.4 0.75)
                              (0.75 1.0 1 1))

  c4( d e f)
}

```



### Véase también

Glosario musical: Sección “slur” en *Glosario Musical*.

Manual de aprendizaje: Sección “Acerca de la no anidabilidad de llaves y ligaduras” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 661, [Ligaduras de fraseo], página 143.

Fragmentos de código: Sección “Expressive marks” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Slur” en *Referencia de Funcionamiento Interno*.

## Ligaduras de fraseo

Las *ligaduras de fraseo* (o marcas de fraseo) que indican una frase musical se escriben usando las instrucciones `\(` y `\)` respectivamente:

```
\relative {
  c'4\( d( e) f(
  e2) d\)
}
```



Tipográficamente, la ligadura de fraseo se comporta casi exactamente igual que una ligadura de expresión normal. Sin embargo, se tratan como objetos diferentes. Una `\slurUp` no tendrá ningún efecto sobre una ligadura de fraseo. El fraseo se puede situar manualmente por encima o por debajo de las notas; véase Sección 5.4.2 [Dirección y posición], página 661.

Las ligaduras de fraseo simultáneas o superpuestas se introducen usando `\=` igual que se hace con las ligaduras de expresión normales, véase [Ligaduras de expresión], página 140.

Las ligaduras de fraseo pueden ser continuas, de puntos o de rayas. El estilo predeterminado para las ligaduras de fraseo es el continuo:

```
\relative {
  c'4\( e g2\)
  \phrasingSlurDashed
  g4\( e c2\)
  \phrasingSlurDotted
  c4\( e g2\)
  \phrasingSlurSolid
  g4\( e c2\)
}
```



Se puede hacer también que las ligaduras de fraseo sean semi-discontinuas (la primera mitad discontinua y la segunda mitad continua) o semi-continuas (la primera mitad continua y la segunda mitad discontinua):

```
\relative {
  c'4\( e g2\)
  \phrasingSlurHalfDashed
  g4\( e c2\)
  \phrasingSlurHalfSolid
  c4\( e g2\)
  \phrasingSlurSolid
  g4\( e c2\)
}
```



Se pueden definir patrones de discontinuidad para las ligaduras de fraseo:

```
\relative {
  c'4\ ( e g2\ )
  \phrasingSlurDashPattern #0.7 #0.75
  g4\ ( e c2\ )
  \phrasingSlurDashPattern #0.5 #2.0
  c4\ ( e g2\ )
  \phrasingSlurSolid
  g4\ ( e c2\ )
}
```



Las definiciones de patrones de discontinuidad tienen la misma estructura que las definiciones de patrones de discontinuidad de las ligaduras de expresión. Para ver más información acerca de patrones de discontinuidad complejos, consulte los fragmentos de código bajo [Ligaduras de expresión], página 140.

## Instrucciones predefinidas

```
\phrasingSlurUp, \phrasingSlurDown, \phrasingSlurNeutral, \phrasingSlurDashed,
\phrasingSlurDotted, \phrasingSlurHalfDashed, \phrasingSlurHalfSolid,
\phrasingSlurDashPattern, \phrasingSlurSolid.
```

## Véase también

Manual de aprendizaje: Sección “Acerca de la no anidabilidad de llaves y ligaduras” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 661, [Ligaduras de expresión], página 140.

Fragmentos de código: Sección “Expressive marks” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “PhrasingSlur” en *Referencia de Funcionamiento Interno*.

## Marcas de respiración

Las respiraciones se introducen utilizando `\breathe`:

```
{ c''2. \breathe d''4 }
```



A diferencia de otras marcas de expresión, los símbolos de respiración no están asociados con la nota anterior, sino que son eventos independientes. Por ello, todas las marcas de expresión que están adjuntas a la nota anterior, cualquier corchete recto que indique barrado manual, y cualquier paréntesis que indique ligaduras de expresión y de fraseo, deben escribirse antes de `\breathe`.

Las marcas de respiración dan por terminadas las barras automáticas; para sobrescribir este comportamiento, consulte [Barras manuales], página 100.

```
\relative { c''8 \breathe d e f g2 }
```



Están contemplados los indicadores musicales para las marcas de respiración en la notación antigua, o divisiones. Para ver más detalles, consulte [Divisiones], página 478.

## Fragmentos de código seleccionados

### *Cambiar el símbolo de la marca de respiración*

El glifo de la marca de respiración se puede ajustar sobreescribiendo la propiedad de texto del objeto de presentación `BreathingSign`, con cualquier otro texto de marcado.

```
\relative c'' {
  c2
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.rvarcomma" }
  \breathe
  d2
}
```



### *Usar una raya corta como símbolo de respiración*

La música vocal y de viento usa con frecuencia una raya corta como signo de respiración. Esto indica una respiración que quita un poco de tiempo a la nota anterior en lugar de producir una corta pausa, lo que se indica con la marca de respiración en forma de coma. La marca se puede mover ligeramente hacia arriba para alejarla del pentagrama.

```
\relative c'' {
  c2
  \breathe
  d2
  \override BreathingSign.Y-offset = #2.6
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.tickmark" }
  c2
  \breathe
  d2
}
```



### *Insertar una cesura*

Las marcas de cesura se pueden crear sobreescribiendo la propiedad `'text` del objeto `BreathingSign`. También está disponible una marca de cesura curva.

```
\relative c'' {
```

```

\override BreathingSign.text = \markup {
  \musicglyph "scripts.caesura.straight"
}
c8 e4. \breathe g8. e16 c4

\override BreathingSign.text = \markup {
  \musicglyph "scripts.caesura.curved"
}
g8 e'4. \breathe g8. e16 c4
}

```



## Véase también

Glosario musical: Sección “caesura” en *Glosario Musical*.

Referencia de la notación: [Divisiones], página 478.

Fragmentos de código: Sección “Expressive marks” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “BreathingEvent” en *Referencia de Funcionamiento Interno*, Sección “BreathingSign” en *Referencia de Funcionamiento Interno*, Sección “Breathing\_sign\_engraver” en *Referencia de Funcionamiento Interno*.

## Caídas y elevaciones

Se pueden expresar *caídas* y *subidas* de tono (falls y doits) añadidas a las notas mediante la instrucción `\bendAfter`. La dirección de la caída o elevación se indica con un signo más o menos (arriba o abajo). El número indica el intervalo de alturas sobre el que se extiende la caída o elevación *partiendo de* la nota principal.

```

\relative c'' {
  c2\bendAfter #+4
  c2\bendAfter #-4
  c2\bendAfter #+6.5
  c2\bendAfter #-6.5
  c2\bendAfter #+8
  c2\bendAfter #-8
}

```



## Fragmentos de código seleccionados

### *Ajustar la forma de las subidas y caídas de tono*

Puede ser necesario trucar la propiedad `shortest-duration-space` para poder ajustar el tamaño de las caídas y subidas de tono («falls» y «doits»).

```

\relative c'' {
  \override Score.SpacingSpanner.shortest-duration-space = #4.0
  c2-\bendAfter #5
  c2-\bendAfter #-4.75
}

```

```

c2-\bendAfter #8.5
c2-\bendAfter #-6
}

```



## Véase también

Glosario musical: Sección “fall” en *Glosario Musical*, Sección “doit” en *Glosario Musical*.

Fragmentos de código: Sección “Expressive marks” en *Fragmentos de código*.

### 1.3.3 Expresiones como líneas

Esta sección explica cómo crear diversas marcas de expresión que siguen un camino lineal: glissandi, arpeggios y trinos.

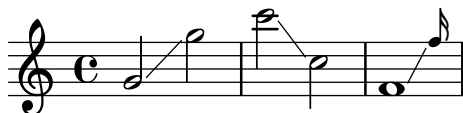
#### Glissando

Se crea un *glissando* añadiendo `\glissando` después de la nota:

```

\relative {
  g'2\glissando g'
  c2\glissando c,
  \afterGrace f,1\glissando f'16
}

```

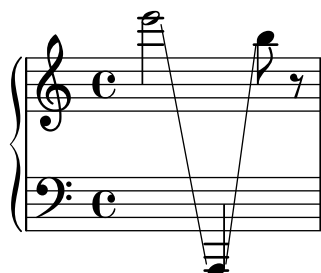


Un glissando puede conectar notas de un pentagrama a otro:

```

\new PianoStaff <<
  \new Staff = "right" {
    e''2\glissando
    \change Staff = "left"
    a,,4\glissando
    \change Staff = "right"
    b''8 r |
  }
  \new Staff = "left" {
    \clef bass
    s1
  }
>>

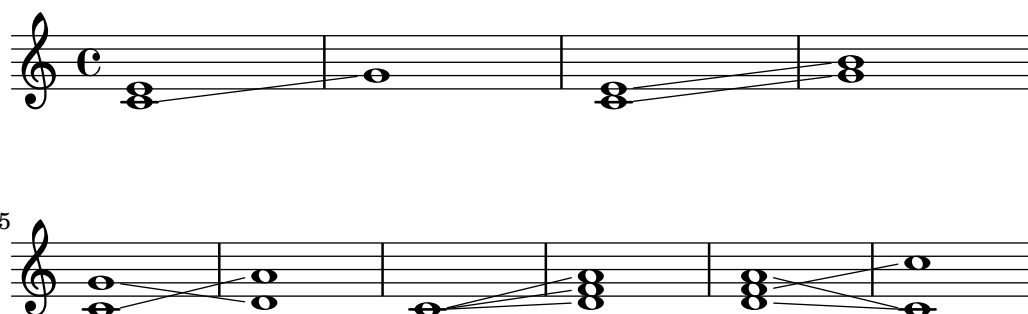
```





Un glissando puede conectar las notas dentro de un acorde. Si se necesita hacer algo que no sea una conexión de una nota con otra entre los dos acordes, las conexiones entre las notas vienen definidas por medio de `\glissandoMap`, donde las notas de un acorde se suponen numeradas, empezando en cero, en el orden en que aparecen en el código de entrada del archivo `.ly`.

```
\relative {
  <c' e>1\glissando g' |
  <c, e>1\glissando |
  <g' b> |
  \break
  \set glissandoMap = #'((0 . 1) (1 . 0))
  <c, g'>1\glissando |
  <d a'> |
  \set glissandoMap = #'((0 . 0) (0 . 1) (0 . 2))
  c1\glissando |
  <d f a> |
  \set glissandoMap = #'((2 . 0) (1 . 0) (0 . 1))
  <f d a'>1\glissando |
  <c c'> |
}
```



Se pueden crear distintos tipos de glissando. Para ver más detalles, consulte Sección 5.4.8 [Estilos de línea], página 676.

## Fragmentos de código seleccionados

### *Glissando contemporáneo*

Se puede tipografiar un glissando contemporáneo sin nota final utilizando una nota oculta y temporalización de cadenza.

```
\relative c'' {
  \time 3/4
  \override Glissando.style = #'zigzag
  c4 c
  \cadenzaOn
  c4\glissando
  \hideNotes
  c,,4
  \unHideNotes
  \cadenzaOff
  \bar ""|"
```

}



### *Añadir marcas de tiempo a glissandos largos*

Los pulsos que se saltan en glissandos muy largos se indican a veces mediante marcas de tiempo, que a menudo consisten en figuras sin cabeza. Estas plicas se pueden usar también para albergar indicaciones expresivas intermedias.

Si las plicas no quedan bien alineadas con el glissando, podría ser necesario recolocarlas ligeramente.

```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}
```

```
glissandoSkipOff = {
  \revert NoteColumn.glissando-skip
  \undo \hide NoteHead
  \revert NoteHead.no-ledgers
}
```

```
\relative c'' {
  r8 f8\glissando
  \glissandoSkipOn
  f4 g a a8\noBeam
  \glissandoSkipOff
  a8
```

```
  r8 f8\glissando
  \glissandoSkipOn
  g4 a8
  \glissandoSkipOff
  a8 |
```

```
  r4 f\glissando \<
  \glissandoSkipOn
  a4\f \>
  \glissandoSkipOff
  b8\! r |
}
```

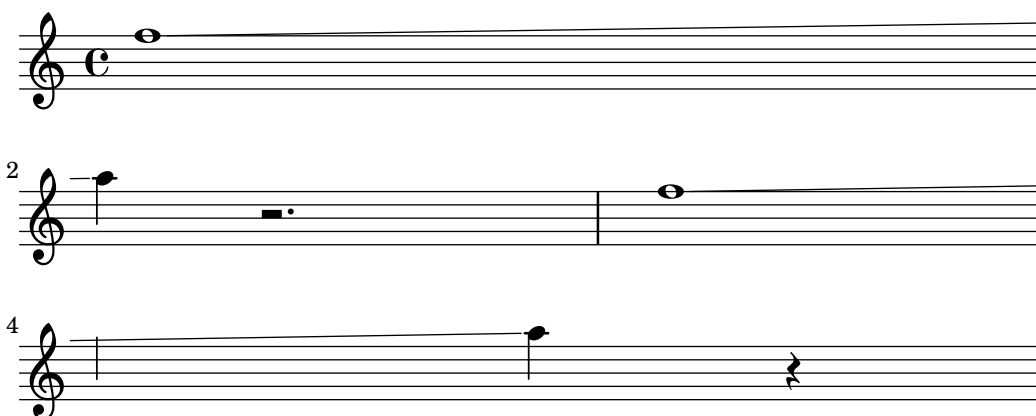


### *Hacer que los glissandos se puedan dividir en el salto de línea*

Si se ajusta la propiedad `breakable` al valor `#t` en combinación con `after-line-breaking`, podemos hacer que un glissando se divida en el salto de línea:

```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

\relative c'' {
  \override Glissando.breakable = ##t
  \override Glissando.after-line-breaking = ##t
  f1\glissando |
  \break
  a4 r2. |
  f1\glissando
  \once \glissandoSkipOn
  \break
  a2 a4 r4 |
}
```



### *Extender glissandos sobre repeticiones*

Se puede simular un glissando que se extiende hasta el interior de varios bloques `\alternative` de primera y segunda vez mediante la adición de una nota de adorno oculta con un glissando al comienzo de cada bloque `\alternative`. La nota de adorno debe estar a la misma altura que la nota que da inicio al primer glissando. Esto se implementa aquí con una función musical que toma como argumento la altura de la nota de adorno.

Observe que en música polifónica la nota de adorno debe coincidir con las notas de adorno correspondientes en todas las otras voces.

```
repeatGliss = #(define-music-function (grace)
  (ly:pitch?)
  #{
    % the next two lines ensure the glissando is long enough
    % to be visible
    \once \override Glissando.springs-and-rods
      = #ly:spanner::set-spacing-rods
    \once \override Glissando.minimum-length = #3.5
    \once \hideNotes
```

```

    \grace $grace \glissando
    #})

\score {
  \relative c'' {
    \repeat volta 3 { c4 d e f\glissando }
    \alternative {
      { g2 d }
      { \repeatGliss f g2 e }
      { \repeatGliss f e2 d }
    }
  }
}

music = \relative c' {
  \voiceOne
  \repeat volta 2 {
    g a b c\glissando
  }
  \alternative {
    { d1 }
    { \repeatGliss c \once \omit StringNumber e1\2 }
  }
}

\score {
  \new StaffGroup <<
    \new Staff <<
      \new Voice { \clef "G_8" \music }
    >>
    \new TabStaff <<
      \new TabVoice { \clef "moderntab" \music }
    >>
  >>
}

```

The image displays two musical staves. The top staff is a treble clef staff with a C-clef, showing a sequence of notes (C4, D4, E4, F4) followed by a glissando (indicated by a wavy line) to G4. This is repeated three times, with first, second, and third endings marked above the staff. The bottom staff is a guitar tablature staff, showing fret numbers (0, 2, 3, 5) and a glissando (indicated by a wavy line) to fret 8. This is repeated twice, with first and second endings marked above the staff.

## Véase también

Glosario musical: Sección “glissando” en *Glosario Musical*.

Referencia de la notación: Sección 5.4.8 [Estilos de línea], página 676.

Fragmentos de código: Sección “Expressive marks” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Glissando” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

La impresión de texto sobre la línea (como *gliss.*) no está contemplada.

## Arpeggio

Un signo de *acorde arpegiado* (conocido también como acorde quebrado) sobre un acorde, se denota adjuntando `\arpeggio` al acorde:

```
\relative { <c' e g c>1\arpeggio }
```



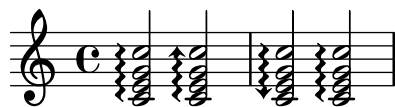
Se pueden escribir distintos tipos de acordes arpegiados. `\arpeggioNormal` produce la vuelta al arpeggio normal:

```
\relative {
  <c' e g c>2\arpeggio

  \arpeggioArrowUp
  <c e g c>2\arpeggio

  \arpeggioArrowDown
  <c e g c>2\arpeggio

  \arpeggioNormal
  <c e g c>2\arpeggio
}
```



Estas instrucciones predefinidas modifican internamente la propiedad '`arpeggio-direction`'; véanse sus definiciones completas en el archivo `ly/property-init.ly`.

Se pueden crear símbolos especiales de arpeggio *con corchete*:

```
\relative {
  <c' e g c>2

  \arpeggioBracket
  <c e g c>2\arpeggio

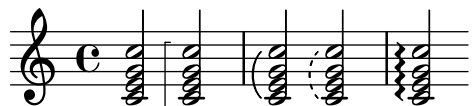
  \arpeggioParenthesis
  <c e g c>2\arpeggio

  \arpeggioParenthesisDashed
  <c e g c>2\arpeggio
```

```

\arpeggioNormal
<c e g c>2\arpeggio
}

```



Estas instrucciones predefinidas sobrescriben internamente la propiedad '**stencil**' del objeto **Arpeggio**, y puede también adaptar su '**X-extent**' (esto es, la dimensión horizontal que le supone no colisionar con otros objetos).

Las propiedades de discontinuidad del paréntesis del arpeggio se controlan mediante la propiedad '**dash-definition**', que se describe en [Ligaduras de expresión], página 140.

Los acordes arpegiados se pueden desarrollar explícitamente utilizando *ligaduras de unión*. Véase [Ligaduras de unión], página 58.

## Instrucciones predefinidas

`\arpeggio`, `\arpeggioArrowUp`, `\arpeggioArrowDown`, `\arpeggioNormal`, `\arpeggioBracket`, `\arpeggioParenthesis`, `\arpeggioParenthesisDashed`.

## Fragmentos de código seleccionados

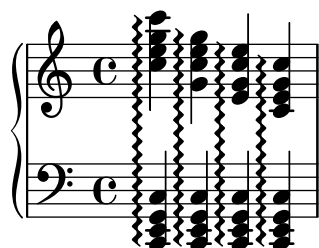
### *Crear arpeggios que se cruzan entre pentagramas dentro de un sistema de piano*

Dentro de un **PianoStaff**, es posible hacer que un arpeggio cruce entre los pentagramas ajustando la propiedad **PianoStaff.connectArpeggios**.

```

\new PianoStaff \relative c'' <<
  \set PianoStaff.connectArpeggios = ##t
  \new Staff {
    <c e g c>4\arpeggio
    <g c e g>4\arpeggio
    <e g c e>4\arpeggio
    <c e g c>4\arpeggio
  }
  \new Staff {
    \clef bass
    \repeat unfold 4 {
      <c,, e g c>4\arpeggio
    }
  }
}
>>

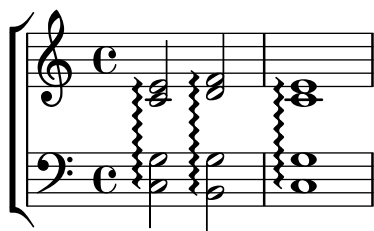
```



### *Creación de arpeggios que se cruzan entre pentagramas dentro de otros contextos*

Se pueden crear arpeggios que se cruzan entre pentagramas dentro de contextos distintos a `GrandStaff`, `PianoStaff` y `StaffGroup` si se incluye el grabador `Span_arpeggio_engraver` en el contexto de `Score`.

```
\score {
  \new ChoirStaff {
    \set Score.connectArpeggios = ##t
    <<
      \new Voice \relative c' {
        <c e>2\arpeggio
        <d f>2\arpeggio
        <c e>1\arpeggio
      }
      \new Voice \relative c {
        \clef bass
        <c g'>2\arpeggio
        <b g'>2\arpeggio
        <c g'>1\arpeggio
      }
    >>
  }
  \layout {
    \context {
      \Score
      \consists "Span_arpeggio_engraver"
    }
  }
}
```



### *Crear arpeggios entre notas de voces distintas*

Se puede trazar un símbolo de arpeggio entre notas de distintas voces que están sobre el mismo pentagrama si el grabador `Span_arpeggio_engraver` se traslada al contexto de `Staff`:

```
\new Staff \with {
  \consists "Span_arpeggio_engraver"
}
\relative c' {
  \set Staff.connectArpeggios = ##t
  <<
    { <e' g>4\arpeggio <d f> <d f>2 }
    \\\
    { <d, f>2\arpeggio <g b>2 }
  >>
```

}



## Véase también

Glosario musical: Sección “arpeggio” en *Glosario Musical*.

Referencia de la notación: [Ligaduras de expresión], página 140, [Ligaduras de unión], página 58.

Archivos instalados: `ly/property-init.ly`.

Fragmentos de código: Sección “Expressive marks” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Arpeggio” en *Referencia de Funcionamiento Interno*, Sección “Slur” en *Referencia de Funcionamiento Interno*, Sección “PianoStaff” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Las instrucciones predefinidas como `\arpeggioArrowUp` solamente se aplican al contexto en curso, y por ello no afectan a los arpeggios que abarcan varias voces o pentagramas. En tales casos, estas instrucciones requieren utilizarse en un bloque `\context` dentro de `\layout`, o en un bloque `\with`, como se explica en Sección 5.1.5 [Cambiar los valores por omisión de los contextos], página 633. De manera alternativa, en lugar de usar las abreviaturas predefinidas, se recomienda sebreescribir directamente las propiedades adecuadas para el objeto `Arpeggio` dentro del contexto correspondiente; por ejemplo:

```
\override Staff.Arpeggio.stencil = #ly:arpeggio::brew-chord-bracket
```

para imprimir corchetes de arpeggio que comprenden más de una voz en el nivel de `Staff`, o bien

```
\override PianoStaff.Arpeggio.arpeggio-direction = #UP
```

para imprimir arpeggios con punta de flecha (hacia arriba) que comprendan más de un pentagrama dentro de un contexto `PianoStaff`.

No es posible mezclar arpeggios conectados y no conectados en un `PianoStaff` en el mismo instante de tiempo.

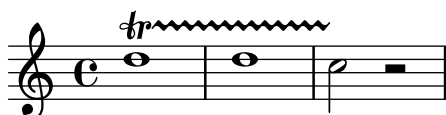
La forma sencilla de especificar el corchete en los arpeggios de paréntesis no funciona para los arpeggios de pentagrama cruzado; see [Líneas de cambio de pentagrama], página 359.

## Trinos

Los trinos cortos sin línea extensora se imprimen con `\trill`; véase [Articulaciones y ornamentos], página 128.

Los trinos largos mantenidos, con línea de extensión, se hacen con `\startTrillSpan` y `\stopTrillSpan`:

```
\relative {
  d'1\startTrillSpan
  d1
  c2\stopTrillSpan
  r2
}
```





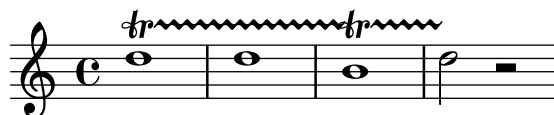
Un trino extendido que atraviesa un salto de línea recomienza exactamente encima de la primera nota de la nueva línea.

```
\relative {
  d''1\startTrillSpan
  \break
  d1
  c2\stopTrillSpan
  r2
}
```



Los trinos extendidos consecutivos funcionan sin necesidad de instrucciones `\stopTrillSpan` explícitas, porque cada trino se convertirá automáticamente en el borde derecho del trino anterior.

```
\relative {
  d''1\startTrillSpan
  d1
  b1\startTrillSpan
  d2\stopTrillSpan
  r2
}
```



También se pueden combinar los trinos con notas de adorno. La sintaxis de esta construcción y el método para colocar las notas de adorno con precisión están descritos en [Notas de adorno], página 120.

```
\relative {
  d''1~\afterGrace
  d1\startTrillSpan { c32[ d]\stopTrillSpan }
  c2 r2
}
```



Los trinos que se tienen que ejecutar sobre notas auxiliares explícitas se pueden tipografiar con la instrucción `pitchedTrill`. El primer argumento es la nota principal, y el segundo es la nota *trinada*, que se imprime como una cabeza de nota, sin plica y entre paréntesis.

```
\relative {
```

```

\pitchedTrill
d''2\startTrillSpan fis
d2
c2\stopTrillSpan
r2
}

```



La alteración del primer trino con altura definida dentro de un compás siempre se imprime, incluso si es un becuadro.

```

{
\key d \major
\pitchedTrill
d'2\startTrillSpan cis d\stopTrillSpan
\pitchedTrill
d2\startTrillSpan c d\stopTrillSpan
\pitchedTrill
d2\startTrillSpan e d\stopTrillSpan
}

```



Es necesario añadir manualmente las alteraciones subsiguientes (de la misma nota dentro del mismo compás).

```

\relative {
\pitchedTrill
eis''4\startTrillSpan fis
eis4\stopTrillSpan
\pitchedTrill
eis4\startTrillSpan cis
eis4\stopTrillSpan
\pitchedTrill
eis4\startTrillSpan fis
eis4\stopTrillSpan
\pitchedTrill
eis4\startTrillSpan fis!
eis4\stopTrillSpan
}

```



## Instrucciones predefinidas

\startTrillSpan, \stopTrillSpan.

## Véase también

Glosario musical: Sección “trill” en *Glosario Musical*.

Referencia de la notación: [Articulaciones y ornamentos], página 128, [Notas de adorno], página 120.

Fragmentos de código: Sección “Expressive marks” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TrillSpanner” en *Referencia de Funcionamiento Interno*.

## 1.4 Repeticiones



La repetición es un concepto fundamental en música, y existen varios tipos de notación para las repeticiones. LilyPond contempla los siguientes tipos de repetición:

### volta (primera y segunda vez)

La repetición de la música no se imprime de forma desarrollada, sino que se indica encerrándola entre barras de repetición. Si el salto de la repetición se encuentra al comienzo de una pieza, la barra de repetición sólo se imprime al final del fragmento. Se imprimen una serie de finales alternativos (volte) de izquierda a derecha indicados mediante corchetes. Ésta es la notación estándar para las repeticiones con finales alternativos.

### unfold (desplegada)

Las música repetida se escribe y se interpreta completamente tantas veces como especifique el valor `número_de_repeticiones`. Es útil cuando se está escribiendo música repetitiva.

### percent (porcentaje)

Hacer repeticiones de compases o parte de ellos. Tienen un aspecto semejante a un signo de porcentaje. Las repeticiones de porcentaje se deben declarar dentro de un contexto `Voice`.

### tremolo (trémolo)

Hacer barras de trémolo.

### 1.4.1 Repeticiones largas

Esta sección trata sobre la forma de introducir repeticiones largas, normalmente de varios compases. Las repeticiones adoptan dos formas: repeticiones encerradas entre signos de repetición, o repeticiones explícitas, que se usan para escribir música repetitiva. También se pueden controlar manualmente los signos de repetición.

#### Repeticiones normales

La sintaxis de una repetición normal es

```
\repeat volta número_de_repeticiones expresión_musical
```

donde *expresión\_musical* es una expresión musical.

Una repetición simple sin finales alternativos:

```
\relative {
  \repeat volta 2 { c''4 d e f }
  c2 d
  \repeat volta 2 { d4 e f g }
}
```



De forma predeterminada no se imprimen las dobles barras de apertura de repetición en el primer compás completo. Sin embargo es posible imprimirlas usando `\bar ".|:"` antes de la primera nota.

```
\relative {
  \repeat volta 2 { \bar ".|:" c''4 d e f }
  c2 d
  \repeat volta 2 { d4 e f g }
}
```



Los finales alternativos (casillas de primera y segunda vez) se pueden generar utilizando `\alternative`. Cada grupo de alternativas debe, a su vez, estar encerrado entre llaves curvas.

```
\repeat volta número_de_repeticiones expresión_musical
\alternative {
  { expresión_musical }
}
```

donde *expresión\_musical* es una expresión musical.

Si existen más repeticiones que finales alternativos, se asigna el primer final alternativo a las repeticiones más antiguas.

Repetición única con primera y segunda vez:

```
\relative {
  \repeat volta 2 { c''4 d e f | }
  \alternative {
    { c2 e | }
    { f2 g | }
  }
}
```

```
c1
}
```



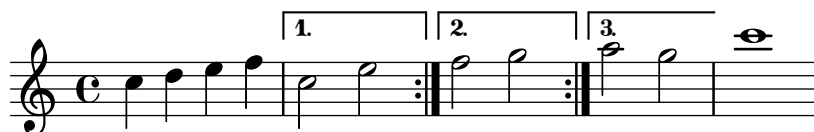
Varias repeticiones con primera y segunda vez:

```
\relative {
  \repeat volta 4 { c''4 d e f | }
  \alternative {
    { c2 e | }
    { f2 g | }
  }
  c1
}
```



Más de una repetición con más de un final alternativo:

```
\relative {
  \repeat volta 3 { c''4 d e f | }
  \alternative {
    { c2 e | }
    { f2 g | }
    { a2 g | }
  }
  c1
}
```



**Nota:** Si hay dos o más alternativas, no debe aparecer nada entre la llave de cierre de una y la de apertura de la otra dentro de un bloque `\alternative`, pues en caso contrario no obtendremos el número de finales esperado.

**Nota:** Si incluimos `\relative` dentro de un `\repeat` sin instanciar el contexto `Voice` explícitamente, aparecerán pentagramas adicionales no deseados. Véase Sección “Aparece un pentagrama de más” en *Utilización del Programa*.

Si una repetición sin casillas de primera y segunda vez comienza en medio de un compás, normalmente termina en el lugar correspondiente en mitad de otro compás posterior (de tal forma que los dos suman un compás completo). En este caso, los símbolos de repetición no son

‘verdaderas’ líneas divisorias, por lo que no deben escribirse en este sitio comprobaciones de compás ni instrucciones `\partial`:

```
c'4 e g
\repeat volta 4 {
  e4 |
  c2 e |
  g4 g g
}
g4 |
a2 a |
g1 |
```



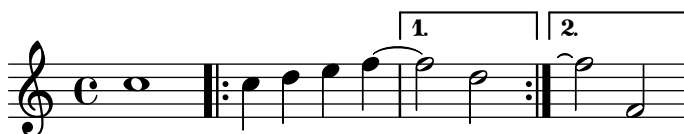
Si una repetición que no tiene casillas de primera y segunda vez comienza con un compás parcial, entonces se aplica el mismo principio, excepto que se requiere una instrucción `\partial` al comienzo del compás:

```
\partial 4
\repeat volta 4 {
  e'4 |
  c2 e |
  g4 g g
}
g4 |
a2 a |
g1 |
```



Se pueden añadir ligaduras de unión al segundo final:

```
\relative {
  c'1
  \repeat volta 2 { c4 d e f~ }
  \alternative {
    { f2 d }
    { f2\repeatTie f, }
  }
}
```



La instrucción `\inStaffSegno` se puede utilizar para generar una barra de compás combinada que incorpora el símbolo del segno junto con la barra de repetición, cuando se usa con la instrucción `\repeat volta`. Se selecciona automáticamente el tipo correcto de línea divisoria de repetición, ya sea de comienzo o fin de repetición, o de doble repetición. Observe que la indicación “D.S.” correspondiente se debe añadir manualmente.

Salida de una repetición:

```
\relative {
  e'1
  \inStaffSegno
  f2 g a b
  c1_"D.S." \bar "|."
}
```



Al comienzo de una repetición:

```
\relative {
  e'1
  \repeat volta 2 {
    \inStaffSegno % start repeat
    f2 g a b
  }
  c1_"D.S." \bar "|."
}
```



Al final de una repetición:

```
\relative {
  e'1
  \repeat volta 2 {
    f2 g a b
    \inStaffSegno % end repeat
  }
  f2 g a b
  c1_"D.S." \bar "|."
}
```



Entre dos repeticiones:

```
\relative {
  e'1
  \repeat volta 2 {
    f2 g a b
  }
  \inStaffSegno % double repeat
  \repeat volta 2 {
    f2 g a b
  }
}
```

```
c1_"D.S." \bar "|."
}
```



Se pueden obtener símbolos de línea divisoria alternativos mediante el establecimiento (en el contexto Score) de las propiedades `segnoType`, `startRepeatSegnoType`, `endRepeatSegnoType` o `doubleRepeatSegnoType` al tipo de línea requerido. Los tipos de línea divisoria alternativos deben seleccionarse a partir de los tipos predefinidos o de tipos definidos anteriormente con la instrucción `\defineBarLine` (véase [Barras de compás], página 104).

```
\defineBarLine ":|.S[" #'(":".S[" "]
\defineBarLine "]" #'("]" "" "")
\relative {
  e'1
  \repeat volta 2 {
    f2 g a b
    \once \set Score.endRepeatSegnoType = ":|.S["
    \inStaffSegno
  }
  f2 g \bar "]" a b
  c1_"D.S." \bar "|."
}
```



## Fragmentos de código seleccionados

### *Acortar los corchetes de primera y segunda vez*

De forma predeterminada, los corchetes de primera y segunda vez se trazan encima de los finales alternativos completos, pero es posible acortarlos estableciendo un valor cierto para `voltaSpannerDuration`. En el ejemplo siguiente, el corchete sólo dura un compás, que corresponde a una duración de 3/4.

```
\relative c'' {
  \time 3/4
  c4 c c
  \set Score.voltaSpannerDuration = #(ly:make-moment 3/4)
  \repeat volta 5 { d4 d d }
  \alternative {
    {
      e4 e e
      f4 f f
    }
    { g4 g g }
  }
}
```



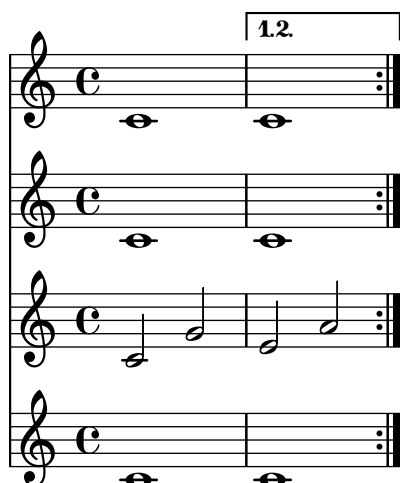
}



### *Añadir corchetes de primera y segunda vez a más pentagramas*

El grabador `Volta_engraver` reside de forma predeterminada dentro del contexto de `Score`, y los corchetes de la repetición se imprimen así normalmente sólo encima del pentagrama superior. Esto se puede ajustar añadiendo el grabador `Volta_engraver` al contexto de `Staff` en que deban aparecer los corchetes; véase también el fragmento de código “Volta multi staff”.

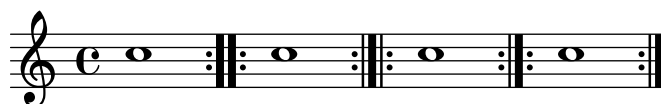
```
<<
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
  \new Staff \with { \consists "Volta_engraver" } { c'2 g' e' a' }
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
>>
```



### *Establecer el tipo de repetición doble predeterminado para la primera y segunda vez*

Existen tres estilos distintos de repeticiones dobles para la primera y segunda vez, que se pueden ajustar utilizando `doubleRepeatType`.

```
\relative c'' {
  \repeat volta 1 { c1 }
  \set Score.doubleRepeatType = #":...:"
  \repeat volta 1 { c1 }
  \set Score.doubleRepeatType = #":|.|:"
  \repeat volta 1 { c1 }
  \set Score.doubleRepeatType = #":|..:"
  \repeat volta 1 { c1 }
}
```



### *Numeración de compases alternativa*

Se pueden seleccionar dos métodos alternativos para la numeración de compases, especiales para cuando hay repeticiones.

```
\relative c'{
  \set Score.alternativeNumberingStyle = #'numbers
  \repeat volta 3 { c4 d e f | }
  \alternative {
    { c4 d e f | c2 d \break }
    { f4 g a b | f4 g a b | f2 a | \break }
    { c4 d e f | c2 d }
  }
  c1 \break
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \repeat volta 3 { c,4 d e f | }
  \alternative {
    { c4 d e f | c2 d \break }
    { f4 g a b | f4 g a b | f2 a | \break }
    { c4 d e f | c2 d }
  }
  c1
}
```

The image displays six staves of musical notation, each illustrating a different alternative numbering style for repeated measures. The notation is in treble clef with a common time signature (C). The first staff, labeled '1.', shows a first ending with a repeat sign and a first ending bracket. The second staff, labeled '2', shows a second ending with a repeat sign and a second ending bracket. The third staff, labeled '2', shows a third ending with a repeat sign and a third ending bracket. The fourth staff, labeled '5', shows a first ending with a repeat sign and a first ending bracket. The fifth staff, labeled '6b', shows a second ending with a repeat sign and a second ending bracket. The sixth staff, labeled '6c', shows a third ending with a repeat sign and a third ending bracket.

## Véase también

Glosario musical: Sección “repeat” en *Glosario Musical*, Sección “volta” en *Glosario Musical*.

Referencia de la notación: [Barras de compás], página 104, Sección 5.1.4 [Modificar los complementos (plug-ins) de contexto], página 631, [Modificación de ligaduras de unión y de expresión], página 684, [Gestión del tiempo], página 126.

Archivos instalados: `ly/engraver-init.ly`.

Fragmentos de código: Sección “Repeats” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “VoltaBracket” en *Referencia de Funcionamiento Interno*, Sección “RepeatedMusic” en *Referencia de Funcionamiento Interno*, Sección “VoltaRepeatedMusic” en *Referencia de Funcionamiento Interno*, Sección “UnfoldedRepeatedMusic” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Las ligaduras de expresión que abarcan desde un bloque `\repeat` hasta un bloque `\alternative` sólo funcionan para la casilla de primera vez. La apariencia visual de una ligadura de expresión que se continúa introduciéndose en otras casillas de repetición puede simularse con `\repeatTie` si la ligadura se extiende hasta una sola nota dentro de la casilla de repetición, aunque este método no funciona en `TabStaff`. Otros métodos que pueden venir bien para indicar ligaduras que se continúan por encima de varias notas en las casillas de repetición, y que también funcionan en contextos de tablatura `TabStaff`, están explicados en [Modificación de ligaduras de unión y de expresión], página 684.

Asimismo, no es posible hacer que las ligaduras de expresión se replieguen desde el final de una alternativa hasta el comienzo de la repetición.

Los glissandos que se extienden desde un bloque `\repeat` hasta un bloque `\alternative` solamente funcionan para la casilla de primera vez. Se puede indicar la apariencia visual de un glissando que se continúa hasta el interior de otras casillas de repetición mediante la codificación de un glissando que empieza en una nota de adorno oculta. Para ver un ejemplo, consulte “Extender los glissandos sobre las repeticiones” bajo el epígrafe Fragmentos de código seleccionados, en [Glissando], página 147.

Si una repetición que comienza con un compás incompleto tiene un bloque `\alternative` que contiene modificaciones a la propiedad `measureLength`, la utilización de `\unfoldRepeats` dará lugar a líneas divisorias erróneamente colocadas y advertencias de comprobación de compás.

Una repetición anidada como

```
\repeat ...
\repeat ...
\alternative
```

es ambigua porque no está claro a qué `\repeat` pertenece la `\alternative`. Esta ambigüedad se resuelve haciendo que la `\alternative` pertenezca siempre a la `\repeat` más interna. Para más claridad, se recomienda usar llaves en tales situaciones.

## Marcas de repetición manual

**Nota:** Estos métodos sólo se utilizan para realizar construcciones de repetición poco usuales, y pueden tener un comportamiento distinto al esperado. En casi todas las situaciones, se deben crear las repeticiones utilizando la instrucción estándar `\repeat` o imprimiendo las barras de compás correspondientes. Para ver más información, consulte [Barras de compás], página 104.

Se puede usar la propiedad `repeatCommands` para controlar la disposición de las repeticiones. Su valor es una lista de Scheme de instrucciones de repetición.

#### `start-repeat`

Imprimir una barra de compás .|:

```
\relative {
  c''1
  \set Score.repeatCommands = #'(start-repeat)
  d4 e f g
  c1
}
```



En la práctica habitual del grabado no se imprimen signos de repetición al principio de la pieza.

#### `end-repeat`

Imprimir una barra de compás :|.

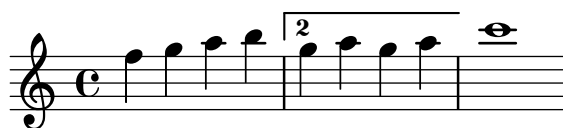
```
\relative {
  c''1
  d4 e f g
  \set Score.repeatCommands = #'(end-repeat)
  c1
}
```



#### `(volta number) ... (volta #f)`

Crear una nueva casilla de repetición con el número que se especifica. El corchete de vez se debe terminar de forma explícita, pues en caso contrario no se imprime.

```
\relative {
  f''4 g a b
  \set Score.repeatCommands = #'((volta "2"))
  g4 a g a
  \set Score.repeatCommands = #'((volta #f))
  c1
}
```



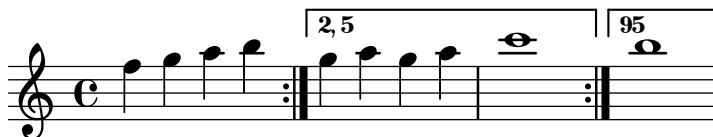
Se pueden producir varias instrucciones de repetición en el mismo punto:

```
\relative {
  f''4 g a b
  \set Score.repeatCommands = #'((volta "2, 5") end-repeat)
  g4 a g a
```

```

c1
\set Score.repeatCommands = #'((volta #f) (volta "95") end-repeat)
b1
\set Score.repeatCommands = #'((volta #f))
}

```



Se puede incluir texto dentro de la casilla de primera y segunda vez. El texto puede ser un número o números, o un elemento de marcado, véase Sección 1.8.2 [Formatear el texto], página 264. La forma más fácil de usar texto de marcado es definir el marcado previamente, y luego incluirlo dentro de una lista de Scheme.

```

voltaAdLib = \markup { 1. 2. 3... \text \italic { ad lib. } }
\relative {
  c''1
  \set Score.repeatCommands =
    #(list(list 'volta voltaAdLib) 'start-repeat)
  c4 b d e
  \set Score.repeatCommands = #'((volta #f) (volta "4.") end-repeat)
  f1
  \set Score.repeatCommands = #'((volta #f))
}

```



## Véase también

Referencia de la notación: [Barras de compás], página 104, Sección 1.8.2 [Formatear el texto], página 264.

Fragmentos de código: Sección “Repeats” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “VoltaBracket” en *Referencia de Funcionamiento Interno*, Sección “RepeatedMusic” en *Referencia de Funcionamiento Interno*, Sección “VoltaRepeatedMusic” en *Referencia de Funcionamiento Interno*.

## Repeticiones explícitas

Mediante la utilización de la instrucción `unfold` se pueden usar las repeticiones para simplificar la escritura desplegada de música repetitiva. La sintaxis es:

```
\repeat unfold número_de_repeticiones expresión_musical
```

donde *expresión\_musical* es una expresión musical y *número\_de\_repeticiones* es el número de veces que *expresión\_musical* se repite.

```

\relative {
  \repeat unfold 2 { c''4 d e f }
  c1
}

```



En ciertos casos, especialmente dentro de un contexto `\relative`, la función `\repeat unfold` no es exactamente igual que escribir la expresión musical varias veces. P. ej.:

```
\repeat unfold 2 { a'4 b c }
```

no equivale a

```
a'4 b c | a'4 b c
```

Se pueden hacer repeticiones desplegadas con finales alternativos.

```
\relative {
  \repeat unfold 2 { c''4 d e f }
  \alternative {
    { c2 g' }
    { c,2 b }
  }
  c1
}
```



Si hay más repeticiones que finales alternativos, el primer final alternativo se aplica las veces necesarias hasta que las alternativas restantes completan el número total de repeticiones.

```
\relative {
  \repeat unfold 4 { c''4 d e f }
  \alternative {
    { c2 g' }
    { c,2 b }
    { e2 d }
  }
  c1
}
```



Si existen más finales alternativos que repeticiones, se aplican solo los primeros finales alternativos. Las alternativas restantes se ignoran y no se imprimen.

```
\relative {
  \repeat unfold 2 { c''4 d e f }
  \alternative {
    { c2 g' }
    { c,2 b }
    { e2 d }
  }
  c1
}
```



Es posible también anidar varias funciones `unfold` (con finales alternativos o sin ellos).

```
\relative {
  \repeat unfold 2 {
    \repeat unfold 2 { c''4 d e f }
    \alternative {
      { c2 g' }
      { c,2 b }
    }
  }
}
c1
}
```



Las construcciones de acorde se pueden repetir mediante el símbolo de repetición de acordes q. Véase [Repetición de acordes], página 178.

**Nota:** Si pone `\relative` dentro de un bloque `\repeat` sin instanciar explícitamente el contexto `Voice`, aparecerán pentagramas no deseados. Véase Sección “Aparece un pentagrama de más” en *Utilización del Programa*.

## Véase también

Referencia de la notación: [Repetición de acordes], página 178.

Snippets: Fragmentos de código: Sección “Repeats” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “RepeatedMusic” en *Referencia de Funcionamiento Interno*, Sección “UnfoldedRepeatedMusic” en *Referencia de Funcionamiento Interno*.

### 1.4.2 Repeticiones cortas

Esta sección trata de cómo introducir repeticiones cortas. Las repeticiones cortas pueden adoptar dos formas: barras inclinadas o signos de porcentaje si estamos representando repeticiones de una sola nota, un compás o dos compases, y trémolos en caso contrario.

#### Repeticiones de compás o parte de ellos

Los patrones cortos que se repiten se imprimen una sola vez, y el patrón repetido se sustituye por un símbolo especial.

La sintaxis es:

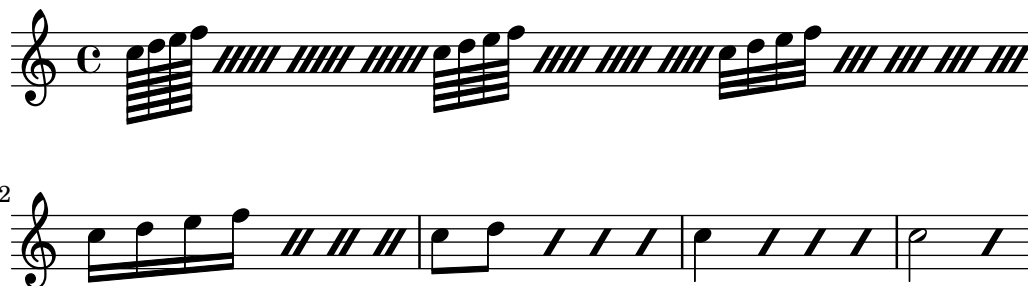
```
\repeat percent número expresión_musical
```

donde *expresión\_musical* es una expresión musical.

Los patrones más cortos que un compás se sustituyen por barras inclinadas.

```
\relative c'' {
  \repeat percent 4 { c128 d e f }
  \repeat percent 4 { c64 d e f }
  \repeat percent 5 { c32 d e f }
  \repeat percent 4 { c16 d e f }
  \repeat percent 4 { c8 d }
  \repeat percent 4 { c4 }
```

```
\repeat percent 2 { c2 }
}
```



Los patrones de repetición de uno o dos compases se sustituyen por signos parecidos al símbolo de porcentaje.

```
\relative c'' {
  \repeat percent 2 { c4 d e f }
  \repeat percent 2 { c2 d }
  \repeat percent 2 { c1 }
}
```

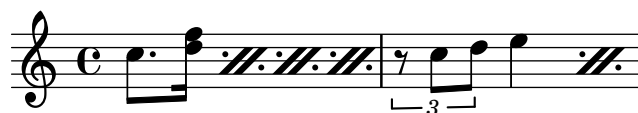


```
\relative {
  \repeat percent 3 { c''4 d e f | c2 g' }
}
```



Los patrones más cortos que un compás pero que contienen duraciones mezcladas utilizan un símbolo de porcentaje doble.

```
\relative {
  \repeat percent 4 { c''8. <d f>16 }
  \repeat percent 2 { \tuplet 3/2 { r8 c d } e4 }
}
```



## Fragmentos de código seleccionados

### *Contador de repeticiones de tipo porcentaje*

Las repeticiones de compases completos de más de dos repeticiones pueden llevar un contador si se activa la propiedad adecuada, como se ve en este ejemplo:

```
\relative c'' {
  \set countPercentRepeats = ##t
  \repeat percent 4 { c1 }
}
```



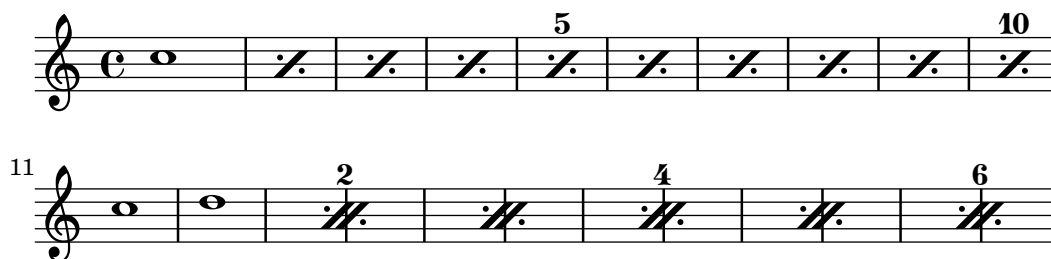
}



### *Visibilidad del contador de repeticiones de tipo porcentaje*

Se pueden mostrar los contadores de las repeticiones del tipo porcentaje a intervalos regulares mediante el establecimiento de la propiedad de contexto `repeatCountVisibility`.

```
\relative c'' {
  \set countPercentRepeats = ##t
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 5)
  \repeat percent 10 { c1 } \break
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 2)
  \repeat percent 6 { c1 d1 }
}
```



### *Símbolos de porcentaje sueltos*

También se pueden imprimir símbolos de porcentaje sueltos.

```
makePercent =
#(define-music-function (note) (ly:music?)
  "Make a percent repeat the same length as NOTE."
  (make-music 'PercentEvent
    'length (ly:music-length note)))

\relative c'' {
  \makePercent s1
}
```



### **Véase también**

Glosario musical: Sección “percent repeat” en *Glosario Musical*, Sección “simile” en *Glosario Musical*.

Fragmentos de código: Sección “Repeats” en *Fragmentos de código*.

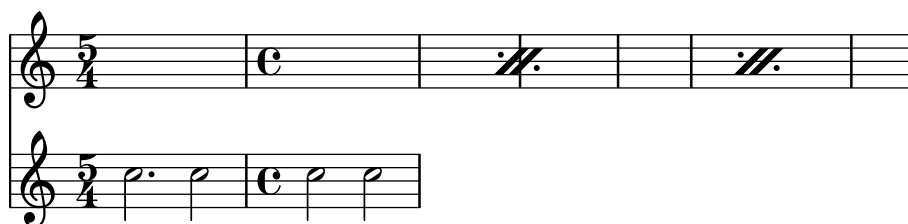
Referencia de funcionamiento interno: Sección “RepeatSlash” en *Referencia de Funcionamiento Interno*, Sección “RepeatSlashEvent” en *Referencia de Funcionamiento Interno*, Sección “DoubleRepeatSlash” en *Referencia de Funcionamiento Interno*, Sección “PercentRepeat” en *Referencia de Funcionamiento Interno*, Sección “PercentRepeatCounter” en *Referencia de Funcionamiento Interno*, Sección “PercentRepeatedMusic” en *Referencia de Funcionamiento Interno*, Sección “Percent\_repeat\_engraver” en *Referencia de Funcionamiento Interno*, Sección

“DoublePercentEvent” en *Referencia de Funcionamiento Interno*, Sección “DoublePercentRepeat” en *Referencia de Funcionamiento Interno*, Sección “DoublePercentRepeatCounter” en *Referencia de Funcionamiento Interno*, Sección “Double\_percent\_repeat\_engraver” en *Referencia de Funcionamiento Interno*, Sección “Slash\_repeat\_engraver” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Las repeticiones de porcentaje no contienen nada más aparte del propio signo de porcentaje; especialmente, los cambios de indicación de compás no se repiten.

```
\repeat percent 3 { \time 5/4 c2. 2 \time 4/4 2 2 }
```



Cualquier cambio de compás o instrucción `\partial` se tiene que producir en pasajes paralelos *fuera* de la repetición de porcentaje, p. ej. en una pista especial para la gestión del compás.

```
<<
\repeat percent 3 { c2. 2 2 2 }
\repeat unfold 3 { \time 5/4 s4*5 \time 4/4 s1 }
>>
```

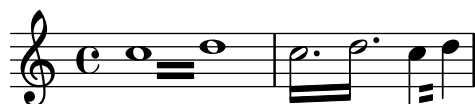


## Repeticiones de trémolo

Los trémolos pueden adoptar dos formas: alternancia entre dos acordes o dos notas, y repetición rápida de una sola nota o acorde. Los trémolos que consisten en una alternancia se indican por medio de la adición de barras entre las notas o acordes que se alternan, mientras que los trémolos que consisten en la repetición rápida de una sola nota se indican mediante la adición de barras cruzadas a una nota única.

Para colocar marcas de trémolo entre las notas, use `\repeat` con el estilo `tremolo` (trémolo):

```
\relative c'' {
\repeat tremolo 8 { c16 d }
\repeat tremolo 6 { c16 d }
\repeat tremolo 2 { c16 d }
}
```



La sintaxis de `\repeat tremolo` espera que se escriban exactamente dos notas dentro de las llaves, y el número de repeticiones debe corresponderse con un valor que se pueda expresar con

figuras normales o con puntillo. Así, `\repeat tremolo 7` es válido y produce una nota con doble puntillo, pero `\repeat tremolo 9` no es válido.

La duración del trémolo es igual a la duración de la expresión entre llaves, multiplicada por el número de repeticiones: `\repeat tremolo 8 { c16 d16 }` da como resultado un trémolo de redonda, escrito como dos redondas unidas por barras de trémolo.

Existen dos maneras de colocar marcas de trémolo sobre una única nota. Incluso aquí se puede utilizar la sintaxis `\repeat tremolo`, en cuyo caso la nota no debe ir encerrada entre llaves:

```
\repeat tremolo 4 c'16
```



El mismo resultado se puede obtener escribiendo `:N` después de la nota, donde  $N$  indica la duración de la subdivisión (debe ser 8 como mínimo). Si  $N$  es 8, se añade una barra de corchea a la plica de la nota. Si  $N$  se omite, se utiliza el último valor:

```
\relative {
  c''2:8 c:32
  c: c:
}
```



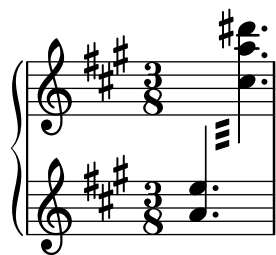
## Fragmentos de código seleccionados

### *Trémolos de pentagrama cruzado*

Dado que `\repeat tremolo` espera exactamente dos argumentos musicales para los trémolos de acorde, la nota o acorde que cambia de pentagrama en un trémolo que cruza el pentagrama se debe colocar dentro de llaves curvas junto a su instrucción `\change Staff`.

```
\new PianoStaff <<
  \new Staff = "up" \relative c'' {
    \key a \major
    \time 3/8
    s4.
  }
  \new Staff = "down" \relative c'' {
    \key a \major
    \time 3/8
    \voiceOne
    \repeat tremolo 6 {
      <a e'>32
      {
        \change Staff = "up"
        \voiceTwo
        <cis a' dis>32
      }
    }
  }
}
```

&gt;&gt;



## Véase también

Fragmentos de código: Sección “Repeats” en *Fragmentos de código*.

## 1.5 Notas simultáneas

The image displays three systems of musical notation for piano, illustrating simultaneous notes in both hands. The first system is in 9/16 time, featuring dynamics *f*, *p*, and *pp*. The second system, starting at measure 112, includes trills. The third system, starting at measure 116, shows a dynamic change from *p* to *f*.

La palabra ‘polifonía’ en música hace referencia a tener más de una voz en una pieza cualquiera de música. En LilyPond la polifonía hace referencia a tener más de una voz en el mismo pentagrama.

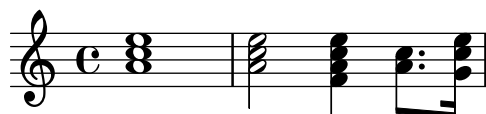
### 1.5.1 Una voz única

Esta sección trata de las notas simultáneas dentro de la misma voz.

#### Notas en acorde

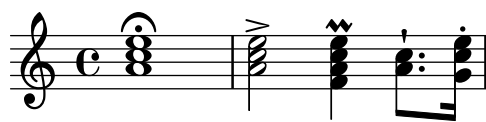
Un acorde se forma encerrando un conjunto de notas entre < y >. Un acorde puede ir seguido de una duración, como si fueran simples notas.

```
\relative {
  <a' c e>1 <a c e>2 <f a c e>4 <a c>8. <g c e>16
}
```



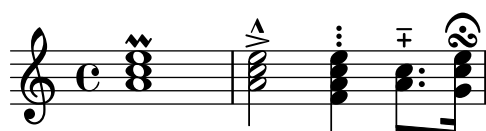
Los acordes también pueden ir seguidos de articulaciones, de nuevo como si fueran simples notas.

```
\relative {
  <a' c e>1\fermata <a c e>2-> <f a c e>4\prall <a c>8.^! <g c e>16-.
}
```



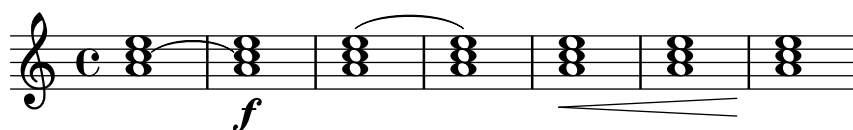
Las notas dentro del propio acorde también pueden ir seguidas de articulaciones y ornamentos.

```
\relative {
  <a' c\prall e>1 <a-> c-^ e>2 <f-. a c-. e-.>4
  <a-+ c-->8. <g\fermata c e\turn>16
}
```



Sin embargo, algunos elementos de notación tales como las expresiones de matices dinámicos y los reguladores se deben unir al acorde y no a las notas que integran el mismo, pues en caso contrario no se imprimirán. Otros elementos de notación como las digitaciones y las ligaduras de expresión se posicionan de forma muy diferente cuando se adjuntan a las notas de un acorde en lugar de hacerlo a redondas o notas sueltas.

```
\relative {
  <a'\f c( e>1 <a c) e>\f <a\< c e>( <a\! c e>)
  <a c e>\< <a c e> <a c e>\!
}
```



Un acorde funciona como un mero contenedor para las notas que lo componen y sus articulaciones y otros elementos adjuntos. Como consecuencia, un acorde sin ninguna nota en su interior

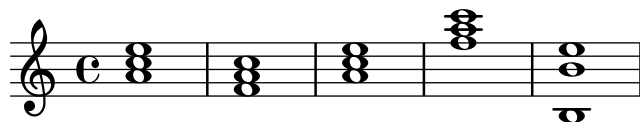
no tiene realmente ninguna duración. Cualquier articulación adjunta se producirá en el mismo momento musical que la nota o acorde siguiente y se puede combinar con ellos (para ver posibilidades más complejas de combinaciones de estos elementos, consulte [Expresiones simultáneas], página 179):

```
\relative {
  \grace { g'8( a b }
  <> ) \p \< -. -\markup \italic "sempre staccato"
  \repeat unfold 4 { c4 e } c1\f
}
```



Se puede usar el modo relativo para la altura de las notas de los acordes. La primera nota del acorde siempre es relativa a la primera nota del acorde anterior, o en caso de que el elemento precedente no sea un acorde, la altura de la última nota que vino antes del acorde. El resto de las notas del acorde son relativas a la nota anterior *dentro del mismo acorde*.

```
\relative {
  <a' c e>1 <f a c> <a c e> <f' a c> <b, e b,>
}
```



Para ver más información sobre los acordes, consulte Sección 2.7 [Notación de acordes], página 442.

## Véase también

Glosario musical: Sección “chord” en *Glosario Musical*.

Manual de aprendizaje: Sección “Combinar notas para formar acordes” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 2.7 [Notación de acordes], página 442, [Articulaciones y ornamentos], página 128, [Escritura de octava relativa], página 2, Sección 1.5.2 [Varias voces], página 181.

Fragmentos de código: Sección “Simultaneous notes” en *Fragmentos de código*.

## Advertencias y problemas conocidos

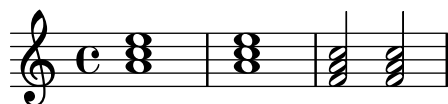
Los acordes que contienen más de dos alturas en el mismo espacio de pentagrama, como ‘<e f! fis!>’, crean notas cuyas cabezas se superponen. Dependiendo de la situación, una mejor representación puede requerir:

- el uso temporal de varias voces, véase Sección 1.5.2 [Varias voces], página 181, ‘<< f! \\  
<e fis!>>>’,
- la transcripción enarmónica de una o más notas, ‘<e f ges>’, or
- Culsters o racimos; véase [Racimos (clusters)], página 181.

## Repetición de acordes

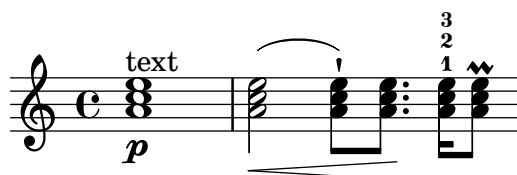
Para reducir el tecleo, se puede usar una abreviatura de repetición del acorde anterior. El símbolo de repetición de acordes es q:

```
\relative {
  <a' c e>1 q <f a c>2 q
}
```



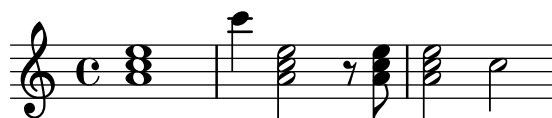
Como en los acordes normales, el símbolo de repetición de acordes puede usarse con duraciones, articulaciones, elementos de marcado, ligaduras de expresión, barras, etc., pues sólo se duplican las notas del acorde precedente.

```
\relative {
  <a' c e>1 \p^"text" q2 \<( q8) [-! q8.] \! q16-1-2-3 q8 \prall
}
```



El símbolo de repetición de acordes siempre recuerda la última ocurrencia de un acorde, por lo que es posible repetir el acorde más reciente incluso si se han escrito en medio otras notas que no están en un acorde, o silencios.

```
\relative {
  <a' c e>1 c'4 q2 r8 q8 |
  q2 c, |
}
```



Sin embargo, el símbolo de repetición de acordes no retiene los matices dinámicos, las articulaciones ni los ornamentos que contiene o que están adosados al acorde anterior.

```
\relative {
  <a'-. c \prall e>1 \sfz c'4 q2 r8 q8 |
  q2 c, |
}
```



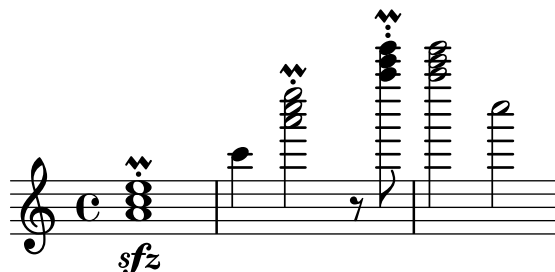
Para poder retener algunos de ellos, se puede llamar explícitamente a la función `\chordRepeats` con un argumento adicional que especifique una lista de *tipos de evento* que conservar, a no ser que los eventos de ese tipo ya estén presentes en el propio acorde q.

```
\relative {
```

```

\chordRepeats #'(articulation-event)
{ <a'-. c\prall e>1\sfz c'4 q2 r8 q8-. } |
q2 c, |
}

```



Aquí, el uso de `\chordRepeats` dentro de una construcción `\relative` produce un resultado inesperado: una vez se han expandido los eventos de acorde, no pueden distinguirse de aquellos introducidos como acordes normales, haciendo que `\relative` asigne una octava basada en su contexto actual.

Dado que las instancias anidadas de `\relative` no se afectan mutuamente, se puede usar otra instrucción `\relative` dentro de `\chordRepeats` para establecer las relaciones de octava antes de expandir los acordes repetidos. En tal caso, todo el contenido de la instrucción `\relative` interior no afecta a la exterior; de aquí la distinta escritura de octava en la última nota de este ejemplo.

```

\relative {
  \chordRepeats #'(articulation-event)
  \relative
  { <a'-. c\prall e>1\sfz c'4 q2 r8 q8-. } |
  q2 c'' |
}

```



Las interacciones con `\relative` se producen solamente con las llamadas explícitas de `\chordRepeats`: la expansión implícita al comienzo del proceso de tipografiado se hace en un momento en que todas las instancias de `\relative` ya se han procesado.

## Véase también

Referencia de la notación: Sección 2.7 [Notación de acordes], página 442, [Articulaciones y ornamentos], página 128.

Archivos de inicio: `ly/chord-repetition-init.ly`.

## Expresiones simultáneas

Una o más expresiones musicales encerradas entre ángulos dobles se entienden como simultáneas. Si la primera expresión comienza con una sola nota o si toda la expresión simultánea aparece explícitamente dentro de una sola voz, la expresión completa se sitúa sobre un solo pentagrama; en caso contrario los elementos de la expresión simultánea se sitúan en pentagramas distintos.

Los ejemplos siguientes muestran expresiones simultáneas sobre un solo pentagrama:

```

\new Voice { % explicit single voice
  << \relative { a'4 b g2 }

```



```
\relative { d'4 g c,2 } >>
}
```



```
\relative {
  % single first note
  a' << \relative { a'4 b g }
    \relative { d'4 g c, } >>
}
```



Esto puede ser de utilidad si las secciones simultáneas tienen idénticas duraciones, pero se producirán errores si se intentan poner notas de distinta duración sobre la misma plica. Las notas, articulaciones y cambios de propiedades que están dentro de *un solo* contexto ‘Voice’ se recolectan y se representan en el orden musical:

```
\relative {
  <a' c>4-. <>-. << c a >> << { c-. <c a> } { a s-. } >>
}
```



Tratar de poner más de una plica o barra de corchea, o distintas duraciones o propiedades en el mismo momento musical, requiere el uso de más de una voz.

El ejemplo siguiente muestra cómo las expresiones simultáneas pueden generar varios pentagramas de forma implícita:

```
% no single first note
<< \relative { a'4 b g2 }
  \relative { d'4 g2 c,4 } >>
```



Aquí no hay problema en tener distintas duraciones porque se interpretan en voces distintas.

## Advertencias y problemas conocidos

Si hay notas que proceden de dos o más voces, sin especificar ningún desplazamiento horizontal, y tienen las plicas en la misma dirección, aparece el mensaje

Advertencia: esta voz requiere un ajuste de `\voiceXx` o `\shiftXx` durante la compilación. Este mensaje se puede suprimir mediante `\override NoteColumn.ignore-collision = ##t`

Sin embargo, esto no solamente suprime la advertencia sino que puede evitar cualquier resolución de colisiones y puede tener otros efectos no deseados (consulte también *Problemas conocidos* en [Resolución de las colisiones], página 186).

## Racimos (clusters)

Un «cluster» o racimo indica que se deben tocar simultáneamente un conjunto de notas consecutivas. Se escriben aplicando la función `\makeClusters` a una secuencia de acordes, p.ej.:

```
\relative \makeClusters { <g' b>2 <c g'> }
```



Se pueden mezclar en el mismo pentagrama notas normales y clusters, incluso al mismo tiempo. En tal caso, no se hace ningún intento de evitar automáticamente las colisiones entre clusters y notas normales.

## Véase también

Glosario musical: Sección “cluster” en *Glosario Musical*.

Fragmentos de código: Sección “Simultaneous notes” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “ClusterSpanner” en *Referencia de Funcionamiento Interno*, Sección “ClusterSpannerBeacon” en *Referencia de Funcionamiento Interno*, Sección “Cluster-spanner-engraver” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Los clusters sólo tienen un buen aspecto cuando abarcan un mínimo de dos acordes. En caso contrario aparecerán excesivamente estrechos.

Los clusters no llevan plica y por sí mismos no pueden indicar las duraciones, pero la longitud del cluster que se imprime viene determinada por la duración de los acordes que lo definen. Los racimos separados necesitan silencios de separación entre ellos.

Los clusters no producen ninguna salida MIDI.

### 1.5.2 Varias voces

Esta sección trata las notas simultáneas en varias voces o varios pentagramas.

## Polifonía en un solo pentagrama

### *Instanciar las voces explícitamente*

La estructura básica necesaria para obtener varias voces independientes en un mismo pentagrama se ilustra en el ejemplo siguiente:

```
\new Staff <<
  \new Voice = "first"
    \relative { \voiceOne r8 r16 g' e8. f16 g8[ c,] f e16 d }
  \new Voice= "second"
    \relative { \voiceTwo d'16 c d8~ 16 b c8~ 16 b c8~ 16 b8. }
>>
```



Aquí se crean explícitamente instancias de voces, cada una de las cuales recibe un nombre. Las instrucciones `\voiceOne ... \voiceFour` (voz uno hasta voz cuatro) preparan las voces de manera que la primera y segunda voces llevan las plicas hacia arriba, las voces segunda y cuarta llevan las plicas hacia abajo, las cabezas de las notas en las voces tercera y cuarta se desplazan horizontalmente, y los silencios de las voces respectivas se desplazan también automáticamente para evitar las colisiones. La instrucción `\oneVoice` (una voz) devuelve todos los ajustes de las voces al estado neutro predeterminado.

### *Pasajes polifónicos temporales*

Se puede crear un pasaje polifónico temporal con la construcción siguiente:

```
<< { \voiceOne ... }
    \new Voice { \voiceTwo ... }
>> \oneVoice
```

Aquí, la primera expresión dentro de un pasaje polifónico temporal se coloca en el contexto `Voice` que estaba en uso inmediatamente antes del pasaje polifónico, y ese mismo contexto `Voice` continua después de la sección temporal. Otras expresiones dentro de los ángulos se asignan a distintas voces temporales. Esto permite asignar la letra de forma continua a una voz antes, durante y después de la sección polifónica:

```
\relative <<
  \new Voice = "melody" {
    a'4
    <<
      {
        \voiceOne
        g f
      }
      \new Voice {
        \voiceTwo
        d2
      }
    >>
    \oneVoice
    e4
  }
  \new Lyrics \lyricsto "melody" {
    This is my song.
  }
>>
```



Aquí, las instrucciones `\voiceOne` y `\voiceTwo` son necesarias para definir los ajustes de cada voz.

### *La construcción de la doble barra invertida*

La construcción `<< { ... } \ { ... } >>`, en que las dos expresiones (o más) están separadas por doble barra invertida, se comporta de forma distinta a la construcción similar sin las dobles

barras: *todas* las expresiones dentro de esta construcción se asignan a contextos **Voice** nuevos. Estos contextos **Voice** nuevos se crean implícitamente y reciben los nombres fijos "1", "2", etc.

El primer ejemplo podría haberse tipografiado de la manera siguiente:

```
<<
  \relative { r8 r16 g'' e8. f16 g8[ c,] f e16 d }
  \\
  \relative { d''16 c d8~ 16 b c8~ 16 b c8~ 16 b8. }
>>
```



Esta sintaxis se puede usar siempre que no nos importe que se creen nuevas voces temporales que después serán descartadas. Estas voces creadas implícitamente reciben ajustes equivalentes al efecto de las instrucciones `\voiceOne ... \voiceFour`, en el orden en que aparecen en el código.

En el siguiente ejemplo, la voz intermedia lleva las plicas hacia arriba, de manera que la introducimos en tercer lugar para que pase a ser la voz tres, que tiene las plicas hacia arriba tal y como queremos. Se usan silencios de separación para evitar la aparición de silencios duplicados.

```
<<
  \relative { r8 g'' g g f16 ees f8 d }
  \\
  \relative { ees'8 r ees r d r d r }
  \\
  \relative { d''8 s c s bes s a s }
>>
```



En todas las partituras excepto las más simples, se recomienda crear contextos **Voice** explícitos como aparece explicado en Sección “Contextos y grabadores” en *Manual de Aprendizaje* y en Sección “Voces explícitas” en *Manual de Aprendizaje*.

### Orden de las voces

Al escribir varias voces en el archivo de entrada, utilice el orden siguiente:

```
Voz 1: las más aguda
Voz 2: la más grave
Voz 3: la segunda más aguda
Voz 4: la segunda más grave
Voz 5: la tercera más aguda
Voz 6: la tercera más grave
etc.
```

A pesar de que esto puede parecer contrario a la intuición, simplifica el proceso de disposición automática de las figuras. Observe que las voces de numeración impar reciben plicas hacia arriba, y las de numeración par reciben plicas hacia abajo:

```
\new Staff <<
```

```

\time 2/4
{ f''2 } % 1: highest
\\
{ c'2 } % 2: lowest
\\
{ d''2 } % 3: second-highest
\\
{ e'2 } % 4: second-lowest
\\
{ b'2 } % 5: third-highest
\\
{ g'2 } % 6: third-lowest
>>

```



Si se desea introducir las voces en un orden distinto, puede ser de utilidad la instrucción `\voices`:

```

\new Staff \voices 1,3,5,6,4,2 <<
\time 2/4
{ f''2 } % 1: highest
\\
{ d''2 } % 3: second-highest
\\
{ b'2 } % 5: third-highest
\\
{ g'2 } % 6: third-lowest
\\
{ e'2 } % 4: second-lowest
\\
{ c'2 } % 2: lowest
>>

```



**Nota:** No se pueden crear letras ni elementos de extensión (como ligaduras, reguladores, etc.) que se crucen de una voz a otra.

### *Duraciones idénticas*

En el caso especial en que queremos tipografiar fragmentos de música que discurre en paralelo y con las mismas duraciones, se pueden combinar en un solo contexto de voz, formando así acordes. Para conseguirlo, las incorporamos dentro de una construcción de música simultánea, dentro de una voz creada explícitamente:

```

\new Voice <<
\relative { e''4 f8 d e16 f g8 d4 }

```

```
\relative { c''4 d8 b c16 d e8 b4 }
>>
```



Este método conduce a barrados extraños y advertencias si los fragmentos de música no tienen las mismas duraciones exactas.

## Instrucciones predefinidas

`\voiceOne`, `\voiceTwo`, `\voiceThree`, `\voiceFour`, `\oneVoice`.

## Véase también

Manual de aprendizaje: Sección “Las voces contienen música” en *Manual de Aprendizaje*, Sección “Voces explícitas” en *Manual de Aprendizaje*.

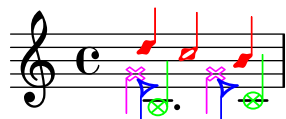
Referencia de la notación: [Pautas de percusión], página 422, [Silencios invisibles], página 64, [Plicas], página 245.

Fragmentos de código: Sección “Simultaneous notes” en *Fragmentos de código*.

## Estilos de voz

Se pueden aplicar colores y formas distintos a las voces para permitir identificarlas fácilmente:

```
<<
\relative { \voiceOneStyle d''4 c2 b4 }
\\
\relative { \voiceTwoStyle e'2 e }
\\
\relative { \voiceThreeStyle b2. c4 }
\\
\relative { \voiceFourStyle g'2 g }
>>
```



Para recuperar la presentación normal se utiliza la instrucción `\voiceNeutralStyle`.

## Instrucciones predefinidas

`\voiceOneStyle`, `\voiceTwoStyle`, `\voiceThreeStyle`, `\voiceFourStyle`, `\voiceNeutralStyle`.

## Véase también

Manual de aprendizaje: Sección “Oigo voces” en *Manual de Aprendizaje*, Sección “Otras fuentes de información” en *Manual de Aprendizaje*.

Fragmentos de código: Sección “Simultaneous notes” en *Fragmentos de código*.

## Resolución de las colisiones

Las cabezas de notas que están en diferentes voces y tienen la misma altura, la misma forma de cabeza, y dirección opuesta de la plica, se combinan automáticamente, pero las que tienen cabezas distintas o la misma dirección de la plica no se combinan. Los silencios que se encuentran en el lado opuesto de una plica en otra voz se desplazan verticalmente. El ejemplo siguiente muestra tres circunstancias distintas, sobre los pulsos 1 y 3 en el primer compás y sobre el pulso 1 del segundo compás, donde la combinación automática falla.

```
<<
\relative {
  c''8 d e d c d c4
  g'2 fis
} \\\
\relative {
  c''2 c8. b16 c4
  e,2 r
} \\\
\relative {
  \oneVoice
  s1
  e'8 a b c d2
}
>>
```



Las cabezas de notas diferentes se pueden combinar como se muestra más abajo. En este ejemplo las cabezas del pulso 1 del primer compás sí se combinan:

```
<<
\relative {
  \mergeDifferentlyHeadedOn
  c''8 d e d c d c4
  g'2 fis
} \\\
\relative {
  c''2 c8. b16 c4
  e,2 r
} \\\
\relative {
  \oneVoice
  s1
  e'8 a b c d2
}
>>
```



Las figuras negras y blancas no se combinan de esta manera, pues sería difícil poder distinguirlas.

También se pueden combinar cabezas con puntillos diferentes, como se muestra en el tercer pulso del primer compás:

```
<<
  \relative {
    \mergeDifferentlyHeadedOn
    \mergeDifferentlyDottedOn
    c''8 d e d c d c4
    g'2 fis
  } \\\
  \relative {
    c''2 c8. b16 c4
    e,2 r
  } \\\
  \relative {
    \oneVoice
    s1
    e'8 a b c d2
  }
>>
```



La blanca y la corchea en el comienzo del segundo compás no están correctamente combinadas porque la combinación automática no es capaz de completarse satisfactoriamente cuando se encuentran tres o más notas alineadas en la misma columna de notas, y en este caso la cabeza combinada es incorrecta. Para conseguir que la combinación seleccione la cabeza correcta, se debe aplicar un `\shiftOn` o desplazamiento a la nota que no se debe combinar. Aquí, se aplica `\shiftOn` para mover el sol agudo fuera de la columna, y entonces `\mergeDifferentlyHeadedOn` hace su trabajo correctamente.

```
<<
  \relative {
    \mergeDifferentlyHeadedOn
    \mergeDifferentlyDottedOn
    c''8 d e d c d c4
    \shiftOn
    g'2 fis
  } \\\
  \relative {
    c''2 c8. b16 c4
    e,2 r
  } \\\
  \relative {
    \oneVoice
    s1
    e'8 a b c d2
  }
>>
```



&gt;&gt;



La instrucción `\shiftOn` permite (aunque no fuerza) que las notas de una voz se puedan desplazar. Cuando `\shiftOn` se aplica a una voz, una nota o acorde de esta voz se desplaza solamente si su plica chocase en caso contrario con una plica de otra voz, y sólo si las plicas en colisión apuntan en la misma dirección. La instrucción `\shiftOff` evita la posibilidad de este tipo de desplazamiento.

De forma predeterminada, las voces externas (normalmente las voces uno y dos) tienen `\shiftOff` especificado (desplazamiento desactivado), mientras que las voces interiores (tres y siguientes) tienen especificado `\shiftOn` (desplazamiento activado). Cuando se aplica un desplazamiento, las voces con plicas hacia arriba (voces de numeración impar) se desplazan hacia la derecha, y las voces con las plicas hacia abajo (voces con numeración par) se desplazan a la izquierda.

He aquí un ejemplo que le ayudará a visualizar la forma en que se expandiría internamente una expresión polifónica.

**Nota:** Observe que con tres o más voces, el orden vertical de las mismas dentro de su archivo de entrada ¡no sería el mismo que el orden vertical de las voces en el pentagrama!

```
\new Staff \relative {
  %% abbreviated entry
  <<
    { f''2 } % 1: highest
    \\\
    { g,2 } % 2: lowest
    \\\
    { d'2 } % 3: upper middle
    \\\
    { b2 } % 4: lower middle
  >>
  %% internal expansion of the above
  <<
    \new Voice = "1" { \voiceOne \shiftOff f'2 }
    \new Voice = "2" { \voiceTwo \shiftOff g,2 }
    \new Voice = "3" { \voiceThree \shiftOn d'2 } % shifts right
    \new Voice = "4" { \voiceFour \shiftOn b2 } % shifts left
  >>
}
```



Dos instrucciones adicionales, `\shiftOnn` y `\shiftOnnn`, ofrecen niveles de desplazamiento mayores que se pueden especificar temporalmente para resolver colisiones en situaciones complejas; véase Sección “Ejemplos reales de música” en *Manual de Aprendizaje*.

Sólo se combinan las notas si tienen la plica en direcciones opuestas (como la tienen, por ejemplo, en las voces uno y dos de forma predeterminada o cuando las plicas se establecen explícitamente en direcciones opuestas).

## Instrucciones predefinidas

```
\mergeDifferentlyDottedOn, \mergeDifferentlyDottedOff, \mergeDifferentlyHeadedOn,
\mergeDifferentlyHeadedOff.
```

```
\shiftOn, \shiftOnn, \shiftOnnn, \shiftOff.
```

## Fragmentos de código seleccionados

### *Voces adicionales para evitar colisiones*

En ciertos casos de polifonía compleja, se necesitan voces adicionales para evitar colisiones entre las notas. Si se necesitan más de cuatro voces paralelas, las voces adicionales se añaden definiendo una variable que utiliza la función de Scheme `context-spec-music`.

```
voiceFive = #(context-spec-music (make-voice-props-set 4) 'Voice)

\relative c'' {
  \time 3/4
  \key d \minor
  \partial 2
  <<
    \new Voice {
      \voiceOne
      a4. a8
      e'4 e4. e8
      f4 d4. c8
    }
    \new Voice {
      \voiceTwo
      d,2
      d4 cis2
      d4 bes2
    }
    \new Voice {
      \voiceThree
      f'2
      bes4 a2
      a4 s2
    }
    \new Voice {
      \voiceFive
      s2
      g4 g2
      f4 f2
    }
  >>
```

}



### *Desplazar las notas con puntillo en polifonía*

Cuando se puede una nota en la voz superior para evitar la colisión con una nota de otra voz, el comportamiento predeterminado es desplazar la nota superior a la derecha. Se puede cambiar usando la propiedad `prefer-dotted-right` de `NoteCollision`.

```
\new Staff \relative c' <<
{
  f2. f4
  \override Staff.NoteCollision.prefer-dotted-right = ##f
  f2. f4
  \override Staff.NoteCollision.prefer-dotted-right = ##t
  f2. f4
}
\\
{ e4 e e e e e e e e e e }
>>
```



### *Forzar el desplazamiento horizontal de las notas*

Cuando el motor de tipografiado no es capaz de todo, se puede usar la sintaxis siguiente para sobrescribir las decisiones de tipografía. Las unidades de medida que se usan aquí son espacios de pentagrama.

```
\relative c' <<
{
  <d g>2 <d g>
}
\\
{
  <b f'>2
  \once \override NoteColumn.force-hshift = #1.7
  <b f'>2
}
>>
```



### **Véase también**

Glosario musical: Sección “polyphony” en *Glosario Musical*.

Manual de aprendizaje: Sección “Varias notas a la vez” en *Manual de Aprendizaje*, Sección “Las voces contienen música” en *Manual de Aprendizaje*, Sección “Ejemplos reales de música” en *Manual de Aprendizaje*.

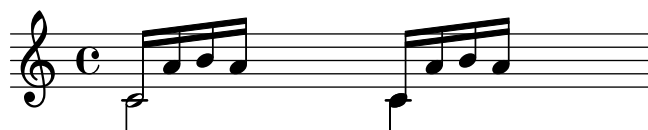
Fragmentos de código: Sección “Simultaneous notes” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “NoteColumn” en *Referencia de Funcionamiento Interno*, Sección “NoteCollision” en *Referencia de Funcionamiento Interno*, Sección “RestCollision” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

El uso de `\override NoteColumn.ignore-collision = ##t` hace que las notas con distinta cabeza en voces diferentes se mezclen incorrectamente.

```
\mergeDifferentlyHeadedOn
<< \relative { c'16 a' b a } \\ \relative { c'2 } >>
\override NoteColumn.ignore-collision = ##t
<< \relative { c'16 a' b a } \\ \relative { c'2 } >>
```



## Combinación de silencios

Al usar más de una voz, es frecuente combinar los silencios que aparecen en las dos partes. Esto puede hacerse mediante el uso del grabador `Merge_rests_engraver`.

```
voiceA = \relative { d''4 r d2 | R1 | }
voiceB = \relative { fis'4 r g2 | R1 | }
\score {
  <<
    \new Staff \with {
      instrumentName = "unmerged"
    }
    <<
      \new Voice { \voiceOne \voiceA }
      \new Voice { \voiceTwo \voiceB }
    >>
    \new Staff \with {
      instrumentName = "merged"
      \consists "Merge_rests_engraver"
    }
    <<
      \new Voice { \voiceOne \voiceA }
      \new Voice { \voiceTwo \voiceB }
    >>
  >>
```

}



Al establecer el valor de la propiedad de contexto `suspendRestMerging` a `##t` podemos desactivar temporalmente la combinación de silencios.

## Combinación automática de las partes

La combinación automática de particellas se usa para mezclar dos partes musicales distintas sobre un pentagrama. Esto puede ser de gran ayuda especialmente al tipografiar partituras orquestales. Se imprime una sola voz cuando la música de las dos voces es la misma, pero en aquellos lugares en que difieren, se imprime una segunda voz. Las direcciones de las plicas se establecen hacia arriba o hacia abajo según proceda, al tiempo que se identifican y quedan marcadas las partes de solista y de dúo.

La sintaxis para la combinación automática de las partes es:

```
\partCombine expresión_musical_1 expresión_musical_2
```

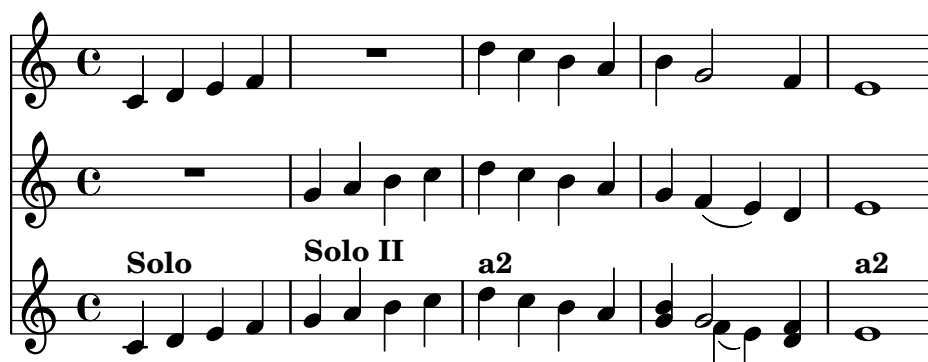
El ejemplo siguiente ejemplifica la funcionalidad básica, poniendo las partes en un solo pentagrama en forma polifónica, y estableciendo las direcciones de las plicas de forma adecuada. Se utilizan las mismas variables para las partes independientes y el pentagrama combinado.

```
instrumentOne = \relative {
  c'4 d e f |
  R1 |
  d'4 c b a |
  b4 g2 f4 |
  e1 |
}

instrumentTwo = \relative {
  R1 |
  g'4 a b c |
  d4 c b a |
  g4 f( e) d |
  e1 |
}

<<
  \new Staff \instrumentOne
  \new Staff \instrumentTwo
  \new Staff \partCombine \instrumentOne \instrumentTwo
```

&gt;&gt;



Las dos partes tienen notas idénticas en el tercer compás, por lo que aparecen solamente una vez. Las direcciones de las plicas y ligaduras se establecen de forma automática, según se trate de un solo o de un unísono. Cuando se necesita en situaciones de polifonía, la primera parte (que recibe el nombre de contexto *one*) recibe las plicas hacia arriba, mientras que la segunda (llamada *two*) siempre recibe las plicas hacia abajo. En los fragmentos de solo, las partes se marcan con “Solo” y “Solo II”, respectivamente. Las partes en unísono (*a due*) se marcan con el texto “a2”.

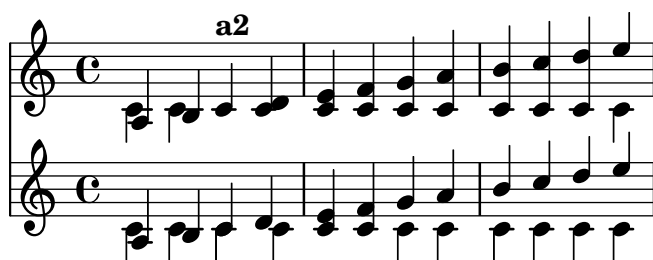
De forma predeterminada, el combinador de partes mezcla dos notas de la misma altura como una nota *a due*, combina las notas que tengan la misma duración y estén a menos de una novena de distancia como acordes, y dispone en voces separadas las notas que estén a más de una novena de distancia (o cuando hay cruce de voces). Este comportamiento se puede sobrescribir con un argumento opcional que consiste en una pareja de números después de la instrucción `\partCombine`: el primero especifica el intervalo donde las notas comienzan a estar combinadas (el valor predeterminado es cero) y el segundo donde las notas se dividen en voces distintas. El ajuste del segundo argumento al valor cero significa que el combinador de partes separa las notas que estén dentro de un intervalo de segunda o más, establecerlo al valor uno separa las notas de una tercera o más, y así sucesivamente.

```
instrumentOne = \relative {
  a4 b c d |
  e f g a |
  b c d e |
}

instrumentTwo = \relative {
  c'4 c c c |
  c c c c |
  c c c c |
}

<<
\new Staff \partCombine \instrumentOne \instrumentTwo
\new Staff \partCombine #'(2 . 3) \instrumentOne \instrumentTwo
```

&gt;&gt;



Los dos argumentos de `\partCombine` se interpretan como contextos de Sección “Voice” en *Referencia de Funcionamiento Interno* separados, por lo que si la música está escrita en modo relativo, entonces *las dos* partes han de incluir la función `\relative`, es decir:

```
\partCombine
  \relative ... expresión_musical_1
  \relative ... expresión_musical_2
```

Una sección `\relative` que se encuentra fuera de `\partCombine` no tiene ningún efecto sobre las notas de `expresión_musical_1` y `expresión_musical_2`.

En las partituras profesionales, las voces con frecuencia se mantienen separadas entre sí durante pasajes prolongados incluso si algunas notas coinciden entre ambas, y podrían escribirse fácilmente como un unísono. Por ello, la combinación de las notas en un acorde, o mostrar una voz como solo, no son soluciones ideales porque la función `\partCombine` considera cada nota de forma individual. En este caso, el resultado de la función `\partCombine` se puede alterar o corregir con una de las instrucciones siguientes. Todas las instrucciones se pueden anteceder por `\once` para que se apliquen solamente a la nota siguiente dentro de la expresión musical.

- `\partCombineApart` mantiene las notas como dos voces separadas incluso si se pueden combinar en un acorde o unísono.
- `\partCombineChords` combina las notas en un acorde.
- `\partCombineUnisono` combina las dos voces como unísono y marca el resultado como “unison”.
- `\partCombineSoloI` muestra solo la voz uno y la marca como “Solo”.
- `\partCombineSoloII` imprime solo la voz dos y la marca como “Solo”.
- `\partCombineAutomatic` termina el efecto de las instrucciones anteriores y retorna a la funcionalidad estándar de `\partCombine`.

```
instrumentOne = \relative c' {
  \partCombineApart c2^"apart" e |
  \partCombineAutomatic e2^"auto" e |
  \partCombineChords e'2^"chord" e |
  \partCombineAutomatic c2^"auto" c |
  \partCombineApart c2^"apart" \once \partCombineChords e^"chord once" |
  c2 c |
}
instrumentTwo = \relative {
  c'2 c |
  e2 e |
  a,2 c |
  c2 c' |
  c2 c |
  c2 c |
}
```

```
<<
  \new Staff { \instrumentOne }
  \new Staff { \instrumentTwo }
  \new Staff { \partCombine \instrumentOne \instrumentTwo }
>>
```

The image shows a musical score with three staves. The top staff has notes with labels 'apart', 'auto', 'chord', 'auto', 'apart', and 'chord once' above them. The middle staff has notes with labels 'apart', 'auto', 'chord', 'auto', 'apart', and 'chord once' above them. The bottom staff has notes with labels 'apart', 'a2', 'chord', 'a2', 'apart', and 'chord once' above them. The notes are in a 4/4 time signature and are written in a simple, clean style.

### *Uso de \partCombine con letra*

La instrucción `\partCombine` no está diseñada para funcionar con la letra de las canciones; si una de las voces recibe un nombre explícito para poder adjuntar una letra, el combinador de partes no funciona. Sin embargo, se puede conseguir este efecto usando un contexto `NullVoice`. Véase [Polifonía con letras compartidas], página 317.

## Fragmentos de código seleccionados

### *Combinar dos partes sobre el mismo pentagrama*

La herramienta de combinación de partes (instrucción `\partCombine`) permite la combinación de varias partes diferentes sobre el mismo pentagrama. Las indicaciones textuales tales como “solo” o “a2” se añaden de forma predeterminada; para quitarlas, sencillamente establezca la propiedad `printPartCombineTexts` al valor “falso”. Para partituras vocales (como himnos), no hay necesidad de añadir los textos “solo” o “a2”, por lo que se deben desactivar. Sin embargo, podría ser mejor no usarlo si hay solos, porque éstos no se indicarán. En tales casos podría ser preferible la notación polifónica estándar.

Este fragmento de código presenta las tres formas en que se pueden imprimir dos partes sobre un solo pentagrama: polifonía estándar, `\partCombine` sin textos, y `\partCombine` con textos.

```
%% Combining pedal notes with clef changes
```

```
musicUp = \relative c'' {
  \time 4/4
  a4 c4.( g8) a4 |
  g4 e' g,( a8 b) |
  c b a2.
}
```

```
musicDown = \relative c'' {
  g4 e4.( d8) c4 |
  r2 g'4( f8 e) |
  d2 \stemDown a
```



```

}

\score {
  <<
    \new Staff \with { instrumentName = "Standard polyphony" }

    << \musicUp \\\ \musicDown >>

    \new Staff \with {
      instrumentName = "PartCombine without text"
      printPartCombineTexts = ##f
    }

    \partCombine \musicUp \musicDown

    \new Staff \with { instrumentName = "PartCombine with text" }
    \partCombine \musicUp \musicDown
  >>
  \layout {
    indent = 6.0\cm
    \context {
      \Score
      \override SystemStartBar.collapse-height = #30
    }
  }
}

```

Standard polyphony	
PartCombine without text	
PartCombine with text	

### *Cambiar los textos de partCombine*

Al utilizar la posibilidad de combinación automática de partes, se puede modificar el texto que se imprime para las secciones de solo y de unísono:

```

\new Staff <<
  \set Staff.soloText = #"girl"
  \set Staff.soloIIIText = #"boy"
  \set Staff.aDueText = #"together"
  \partCombine
  \relative c'' {
    g4 g r r
    a2 g
  }

```

```

\relative c'' {
  r4 r a( b)
  a2 g
}
>>

```



## Véase también

Glosario musical: Sección “a due” en *Glosario Musical*, Sección “part” en *Glosario Musical*.

Referencia de la notación: Sección 1.6.3 [Escritura de las particellas], página 220.

Fragmentos de código: Sección “Simultaneous notes” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “PartCombineMusic” en *Referencia de Funcionamiento Interno*, Sección “Voice” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Todas las funciones `\partCombine...` admiten exclusivamente dos voces.

Las funciones `\partCombine...` no se pueden escribir dentro de los bloques `\tuplet` ni `\relative`.

Si `\printPartCombineTexts` está establecido y las dos voces tocan y terminan las mismas notas en el mismo compás, el combinador de partes puede tipografiar `a2` más de una vez en ese compás.

`\partCombine` solo tiene en cuenta el momento de inicio de las notas dentro de una voz `Voice`; por ejemplo, no puede recordar si una nota dentro de una voz ya ha comenzado cuando las notas que se combinan se han iniciado justo ahora en la otra voz. Esto puede conducir a cierto número de problemas inesperados entre los que se incluye la impresión incorrecta de marcas de “Solo” o de “Unison”.

`\partCombine` conserva todos los elementos extensos (ligaduras, reguladores, etc.) dentro del mismo contexto `Voice` de forma que si uno cualquiera de estos elementos extensos inicia o termina en un contexto `Voice` diferente, puede no imprimirse o hacerlo incorrectamente.

Si la función `\partCombine` no puede combinar las dos expresiones musicales (es decir, cuando las dos voces tienen duraciones distintas), otorgará internamente sus propios nombres a las voces: `one` y `two` respectivamente. Esto significa que si se produce un cambio a un contexto `Voice` que tenga un nombre distinto, se ignorarán los eventos dentro del contexto `Voice` que tiene el nombre distinto.

Consulte también el apartado *Advertencias y problemas conocidos* en la sección [Tablaturas predeterminadas], página 371, al utilizar `\partCombine` con tablaturas, y la *Nota* de [Barras automáticas], página 88, al utilizar barrado automático.

## Escribir música en paralelo

La música para varias partes se puede intercalar dentro del código de entrada. La función `\parallelMusic` admite una lista que contiene los nombres de las variables que se van a crear, y una expresión musical. El contenido de los compases alternativos extraídos de la expresión se convierten en el valor de las variables respectivas, de manera que podemos utilizarlas más tarde para imprimir la música.

**Nota:** Es obligatorio utilizar comprobaciones de compás |, y los compases deben tener la misma longitud.

```
\parallelMusic voiceA,voiceB,voiceC {
  % Bar 1
  r8 g'16 c'' e'' g' c'' e'' r8 g'16 c'' e'' g' c'' e'' |
  r16 e'8.~ 4 r16 e'8.~ 4 |
  c'2 c'2 |

  % Bar 2
  r8 a'16 d'' f'' a' d'' f'' r8 a'16 d'' f'' a' d'' f'' |
  r16 d'8.~ 4 r16 d'8.~ 4 |
  c'2 c'2 |

}
\new StaffGroup <<
  \new Staff << \voiceA \ \ \voiceB >>
  \new Staff { \clef bass \voiceC }
>>
```



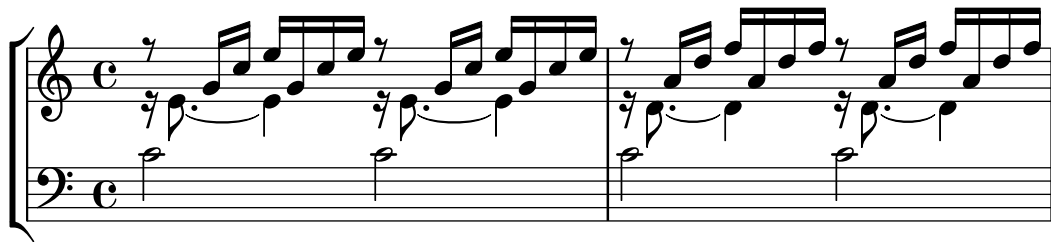
Se puede usar el modo relativo. Observe que la instrucción `\relative` no se utiliza dentro del propio bloque `\parallelMusic`. Las notas guardan relación con la nota anterior en la misma voz, no con la nota anterior dentro del código de entrada (dicho de otra manera, las notas relativas de `vozA` ignoran a las notas que hay en `vozB`).

```
\parallelMusic voiceA,voiceB,voiceC {
  % Bar 1
  r8 g16 c e g, c e r8 g,16 c e g, c e |
  r16 e8.~ 4 r16 e8.~ 4 |
  c2 c |

  % Bar 2
  r8 a,16 d f a, d f r8 a,16 d f a, d f |
  r16 d8.~ 4 r16 d8.~ 4 |
  c2 c |

}
\new StaffGroup <<
  \new Staff << \relative c' \voiceA \ \relative c' \voiceB >>
  \new Staff \relative c' { \clef bass \voiceC }
>>
```

&gt;&gt;



Esto funciona aceptablemente bien para música de piano. El siguiente ejemplo asigna cada cuatro compases consecutivos a cuatro variables:

```
global = {
  \key g \major
  \time 2/4
}

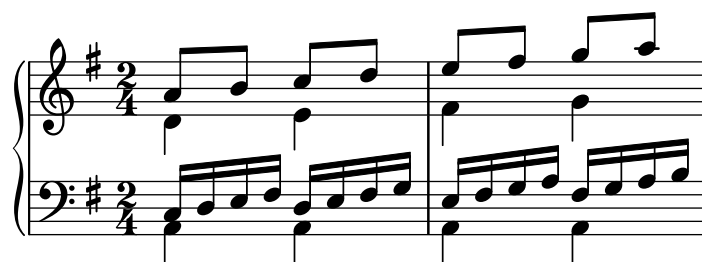
\parallelMusic voiceA,voiceB,voiceC,voiceD {
  % Bar 1
  a8    b    c    d    |
  d4          e    |
  c16 d e fis d e fis g |
  a4          a    |

  % Bar 2
  e8    fis g    a    |
  fis4          g    |
  e16 fis g a fis g a b |
  a4          a    |

  % Bar 3 ...
}

\score {
  \new PianoStaff <<
    \new Staff {
      \global
      <<
        \relative c'' \voiceA
        \\
        \relative c' \voiceB
      >>
    }
    \new Staff {
      \global \clef bass
      <<
        \relative c \voiceC
        \\
        \relative c \voiceD
      >>
    }
  }
}>>
```

}



## Véase también

Manual de aprendizaje: Sección “Organizar las piezas mediante variables” en *Manual de Aprendizaje*

Fragmentos de código: Sección “Simultaneous notes” en *Fragmentos de código*.

## 1.6 Notación de los pentagramas

Esta sección explica cómo influir sobre la apariencia de los pentagramas, cómo imprimir partituras con más de un pentagrama y cómo añadir indicaciones de tempo y notas guía a los pentagramas.

### 1.6.1 Impresión de los pentagramas

Esta sección describe los distintos métodos de creación de pentagramas y grupos de ellos.

## Crear instancias de pentagramas nuevos

Las *pautas* y los *pentagramas* o pautas de cinco líneas se crean con las instrucciones `\new` o `\context`. Para ver más detalles, consulte Sección 5.1.2 [Crear y referenciar contextos], página 624.

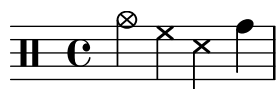
El contexto básico de pentagrama es `Staff`:

```
\new Staff \relative { c''4 d e f }
```



El contexto `DrumStaff` crea una pauta de cinco líneas preparada para un conjunto de batería típico. Cada instrumento se presenta con un símbolo distinto. Los instrumentos se escriben en el modo de percusión que sigue a una instrucción `\drummode`, con cada instrumento identificado por un nombre. Para ver más detalles, consulte [Pautas de percusión], página 422.

```
\new DrumStaff {
  \drummode { cymc hh ss tomh }
}
```



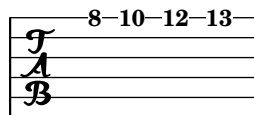
`RhythmicStaff` crea una pauta de una sola línea que sólo muestra las duraciones de la entrada. Se preservan las duraciones reales. Para ver más detalles, consulte [Mostrar los ritmos de la melodía], página 86.

```
\new RhythmicStaff { c4 d e f }
```



`TabStaff` crea una tablatura con seis cuerdas en la afinación estándar de guitarra. Para ver más detalles, consulte [Tablaturas predeterminadas], página 371.

```
\new TabStaff \relative { c''4 d e f }
```



Existen dos contextos de pauta específicos para la notación de música antigua: `MensuralStaff` y `VaticanaStaff`. Se describen en [Contextos predefinidos], página 465.

`GregorianTranscriptionStaff` crea una pauta para la notación moderna de canto gregoriano. No muestra líneas divisorias.

```
\new GregorianTranscriptionStaff \relative { c''4 d e f e d }
```



Se pueden definir contextos nuevos de pentagrama único. Para ver más detalles, consulte Sección 5.1.6 [Definir contextos nuevos], página 638.

## Véase también

Glosario musical: Sección “staff” en *Glosario Musical*, Sección “staves” en *Glosario Musical*.

Referencia de la notación: Sección 5.1.2 [Crear y referenciar contextos], página 624, [Pautas de percusión], página 422, [Mostrar los ritmos de la melodía], página 86, [Tablaturas predeterminadas], página 371, [Contextos predefinidos], página 465, [El símbolo del pentagrama], página 208, [Contextos del canto gregoriano], página 476, [Contextos de la música mensural], página 468, Sección 5.1.6 [Definir contextos nuevos], página 638.

Fragmentos de código: Sección “Staff notation” en *Fragmentos de código*.

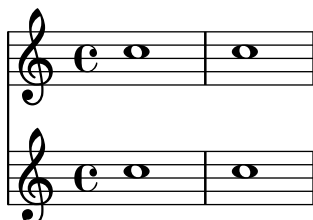
Referencia de funcionamiento interno: Sección “Staff” en *Referencia de Funcionamiento Interno*, Sección “DrumStaff” en *Referencia de Funcionamiento Interno*, Sección “GregorianTranscriptionStaff” en *Referencia de Funcionamiento Interno*, Sección “RhythmicStaff” en *Referencia de Funcionamiento Interno*, Sección “TabStaff” en *Referencia de Funcionamiento Interno*, Sección “MensuralStaff” en *Referencia de Funcionamiento Interno*, Sección “VaticanaStaff” en *Referencia de Funcionamiento Interno*, Sección “StaffSymbol” en *Referencia de Funcionamiento Interno*.

## Agrupar pentagramas

Existen varios contextos para agrupar pentagramas individuales formando sistemas. Cada contexto de agrupación establece el estilo del delimitador de comienzo del sistema y el comportamiento de las barras de compás.

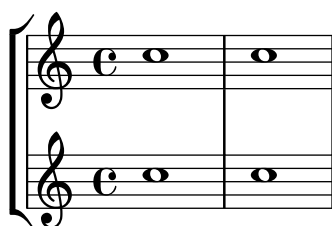
Si no se especifica ningún contexto, se usan las propiedades predeterminadas: el grupo comienza con una línea vertical y las barras de compás no están conectadas.

```
<<
  \new Staff \relative { c''1 c }
  \new Staff \relative { c''1 c }
>>
```



En el contexto `StaffGroup`, el grupo se inicia con un corchete y las barras de compás se dibujan atravesando todos los pentagramas.

```
\new StaffGroup <<
  \new Staff \relative { c''1 c }
  \new Staff \relative { c''1 c }
>>
```



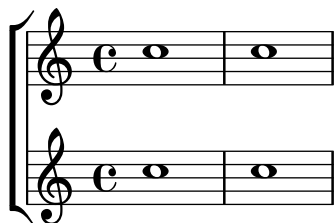
En un `ChoirStaff` (sistema de coro), el grupo se inicia con un corchete, pero las barras de compás no están conectadas.

```
\new ChoirStaff <<
```

```

\new Staff \relative { c''1 c }
\new Staff \relative { c''1 c }
>>

```

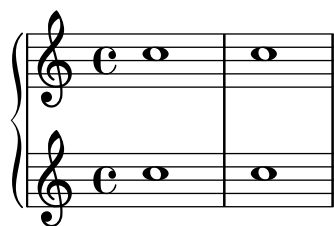


En un **GrandStaff** (sistema de piano), el grupo se inicia con una llave y las barras de compás se conectan entre los pentagramas.

```

\new GrandStaff <<
  \new Staff \relative { c''1 c }
  \new Staff \relative { c''1 c }
>>

```



El **PianoStaff** (sistema de piano) es idéntico a **GrandStaff**, excepto que contempla directamente la impresión del nombre del instrumento. Para ver más detalles, consulte [Nombres de instrumentos], página 220.

```

\new PianoStaff \with { instrumentName = "Piano" }
<<
  \new Staff \relative { c''1 c }
  \new Staff \relative { \clef bass c1 c }
>>

```



Cada contexto de grupo de pentagramas fija la propiedad del delimitador de inicio **systemStartDelimiter** a uno de los siguientes valores: **SystemStartBar** (línea), **SystemStartBrace** (llave) o **SystemStartBracket** (corchete). También está disponible un cuarto delimitador, **SystemStartSquare** (corchete en ángulo recto), pero se debe especificar explícitamente.

Se pueden definir contextos nuevos de grupo de pentagramas. Para ver más detalles, consulte Sección 5.1.6 [Definir contextos nuevos], página 638.

## Fragmentos de código seleccionados



### *Uso del corchete recto al comienzo de un grupo de pentagramas*

Se puede usar el delimitador de comienzo de un sistema `SystemStartSquare` estableciéndolo explícitamente dentro de un contexto `StaffGroup` o `ChoirStaffGroup`.

```
\score {
  \new StaffGroup { <<
    \set StaffGroup.systemStartDelimiter = #'SystemStartSquare
    \new Staff { c'4 d' e' f' }
    \new Staff { c'4 d' e' f' }
  >> }
}
```



### *Mostrar corchete o llave en grupos de un solo pentagrama*

Si hay un solo pentagrama en un de los tipos de sistema `ChoirStaff` o `StaffGroup`, el comportamiento predeterminado es que no se imprima el corchete en la barra inicial. Esto se puede cambiar sobrescribiendo `collapse-height` para fijar su valor de manera que sea menor que el número de líneas en la pauta.

Observe que en contextos como `PianoStaff` y `GrandStaff` en que los sistemas empiezan con una llave en lugar de un corchete, se debe establecer el valor de una propiedad distinta, como se ve en el segundo sistema del ejemplo.

```
\score {
  \new StaffGroup <<
    % Must be lower than the actual number of staff lines
    \override StaffGroup.SystemStartBracket.collapse-height = #4
    \override Score.SystemStartBar.collapse-height = #4
    \new Staff {
      c'1
    }
  >>
}
\score {
  \new PianoStaff <<
    \override PianoStaff.SystemStartBrace.collapse-height = #4
    \override Score.SystemStartBar.collapse-height = #4
    \new Staff {
      c'1
    }
  >>
}
```



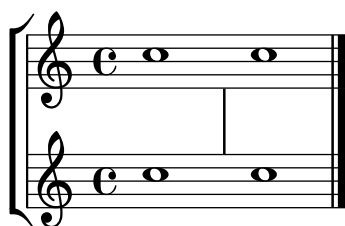


### *Disposición Mensurstriche (líneas divisorias entre pentagramas)*

La disposición «mensurstriche» en que las líneas divisorias no están dibujadas sobre los pentagramas, sino entre ellos, se puede conseguir con un `StaffGroup` en vez de un `ChoirStaff`. La línea divisoria sobre los pentagramas se borra usando `\hide`.

```
global = {
  \hide Staff.BarLine
  s1 s
  % the final bar line is not interrupted
  \undo \hide Staff.BarLine
  \bar "|"
}

\new StaffGroup \relative c' {
  <<
    \new Staff { << \global { c1 c } >> }
    \new Staff { << \global { c c } >> }
  >>
}
```



### Véase también

Glosario musical: Sección “brace” en *Glosario Musical*, Sección “bracket” en *Glosario Musical*, Sección “grand staff” en *Glosario Musical*.

Referencia de la notación: [Nombres de instrumentos], página 220, Sección 5.1.6 [Definir contextos nuevos], página 638.

Fragmentos de código: Sección “Staff notation” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Staff” en *Referencia de Funcionamiento Interno*, Sección “StaffGroup” en *Referencia de Funcionamiento Interno*, Sección “ChoirStaff” en *Referencia de Funcionamiento Interno*, Sección “GrandStaff” en *Referencia de Funcionamiento Interno*, Sección “PianoStaff” en *Referencia de Funcionamiento Interno*, Sección “SystemStartBar” en *Referencia de Funcionamiento Interno*, Sección “SystemStartBrace” en *Referencia de Funcionamiento Interno*, Sección “SystemStartBracket” en *Referencia de Funcionamiento Interno*, Sección “SystemStartSquare” en *Referencia de Funcionamiento Interno*.

### Grupos de pentagramas anidados

Los contextos de grupos de pentagramas se pueden anidar hasta una profundidad arbitraria. En este caso, cada contexto descendiente crea un corchete nuevo adyacente al corchete de su grupo padre.

```
\new StaffGroup <<
  \new Staff \relative { c'2 c | c2 c }
  \new StaffGroup <<
    \new Staff \relative { g'2 g | g2 g }
```

```

\new StaffGroup \with {
  systemStartDelimiter = #'SystemStartSquare
}
<<
  \new Staff \relative { e'2 e | e2 e }
  \new Staff \relative { c'2 c | c2 c }
>>
>>
>>

```



Se pueden definir nuevos contextos de grupos de pentagramas anidados. Para ver más detalles, consulte Sección 5.1.6 [Definir contextos nuevos], página 638.

## Fragmentos de código seleccionados

### *Anidado de grupos de pentagramas*

Se puede utilizar la propiedad `systemStartDelimiterHierarchy` para crear grupos de pentagramas anidados de forma más compleja. La instrucción `\set StaffGroup.systemStartDelimiterHierarchy` toma una lista alfabética del número de pentagramas producidos. Se puede proporcionar antes de cada pentagrama un delimitador de comienzo de sistema. Se debe encerrar entre corchetes y admite tantos pentagramas como encierran las llaves. Se pueden omitir los elementos de la lista, pero el primer corchete siempre abarca todos los pentagramas. Las posibilidades son `SystemStartBar`, `SystemStartBracket`, `SystemStartBrace` y `SystemStartSquare`.

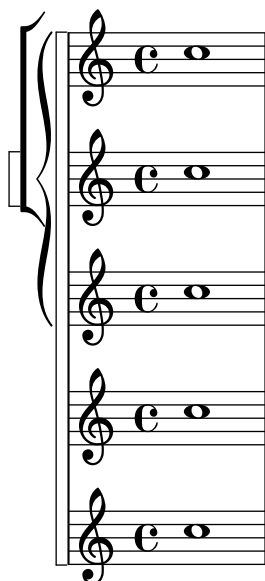
```

\new StaffGroup
\relative c'' <<
  \override StaffGroup.SystemStartSquare.collapse-height = #4
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare (SystemStartBrace (SystemStartBracket a
                                              (SystemStartSquare b) ) c ) d)

  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }

```

&gt;&gt;



## Véase también

Referencia de la notación: [Agrupar pentagramas], página 202, [Nombres de instrumentos], página 220, Sección 5.1.6 [Definir contextos nuevos], página 638.

Fragmentos de código: Sección “Staff notation” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “StaffGroup” en *Referencia de Funcionamiento Interno*, Sección “ChoirStaff” en *Referencia de Funcionamiento Interno*, Sección “SystemStartBar” en *Referencia de Funcionamiento Interno*, Sección “SystemStartBrace” en *Referencia de Funcionamiento Interno*, Sección “SystemStartBracket” en *Referencia de Funcionamiento Interno*, Sección “SystemStartSquare” en *Referencia de Funcionamiento Interno*.

## Separación de sistemas

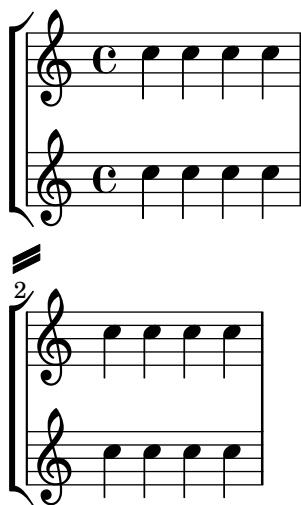
Si el número de sistemas por página varía de una página a otra, es costumbre separar los sistemas colocando una marca separadora entre ellos. De forma predeterminada, el separador de sistemas es nulo, pero se puede activar con una opción de `\paper`.

```
\book {
  \score {
    \new StaffGroup <<
      \new Staff {
        \relative {
          c''4 c c c
          \break
          c4 c c c
        }
      }
      \new Staff {
        \relative {
          c''4 c c c
          \break
          c4 c c c
        }
      }
    }
  }
```

```

>>
}
\paper {
  system-separator-markup = \slashSeparator
  % following commands are needed only to format this documentation
  paper-width = 100\mm
  paper-height = 100\mm
  tagline = ##f
}
}

```



## Véase también

Referencia de la notación: Sección 4.1 [Disposición de la página], página 567.

Fragmentos de código: Sección “Notación de los pentagramas” en *Fragmentos de código*.

### 1.6.2 Modificación de pentagramas sueltos

Esta sección explica cómo cambiar los atributos específicos de un pentagrama: por ejemplo, cambiar el número de líneas o el tamaño de la pauta. También se describen los métodos para comenzar y terminar los pentagramas, y establecer secciones de ossia.

#### El símbolo del pentagrama

Se pueden usar las instrucciones `\stopStaff` y `\startStaff` para detener y (re)iniciar, respectivamente, la impresión de las líneas de la pauta en cualquier punto de la partitura.

```

\relative {
  \stopStaff f''4 d \startStaff g, e
  f'4 d \stopStaff g, e
  f'4 d \startStaff g, e
}

```



#### Instrucciones predefinidas

`\startStaff`, `\stopStaff`.

Las líneas de la pauta pertenecen al grob `StaffSymbol` (incluidas las líneas adicionales) y se pueden modificar usando las propiedades de `StaffSymbol`, pero esas modificaciones deben hacerse antes de que el pentagrama se (re)inicie.

Se puede cambiar el número de líneas de la pauta:

```
\relative {
  f''4 d \stopStaff
  \override Staff.StaffSymbol.line-count = #2
  \startStaff g, e |

  f'4 d \stopStaff
  \revert Staff.StaffSymbol.line-count
  \startStaff g, e |
}
```



La posición de cada una de las líneas de la pauta también puede cambiarse. Una lista de números fija la posición de cada línea. 0 corresponde a la línea central normal, y las posiciones normales de las líneas son (-4 -2 0 2 4). Se imprime una sola línea de pauta por cada valor que se introduce, de manera que el número de líneas así como su posición en la pauta pueden cambiarse con una sola operación de sobreescritura.

```
\relative {
  f''4 d \stopStaff
  \override Staff.StaffSymbol.line-positions = #'(1 3 5 -1 -3)
  \startStaff g, e |
  f'4 d \stopStaff
  \override Staff.StaffSymbol.line-positions = #'(8 6.5 -6 -8 -0.5)
  \startStaff g, e |
}
```



Para preservar las direcciones típicas de las plicas (hacia arriba para la mitad inferior del pentagrama y hacia abajo para la mitad superior), debemos alinear la línea (o espacio) central de la pauta personalizada con la posición de la línea central normal (0). Puede ser necesario un ajuste de la posición de la clave y del Do central para que se correspondan a las líneas nuevas. Véase [Clave], página 18.

Puede cambiarse el grosor de las líneas de la pauta. También resultan afectadas, de forma predeterminada, las líneas adicionales y no la plica de las figuras.

```
\new Staff \with {
  \override StaffSymbol.thickness = #3
} \relative {
  f''4 d g, e
}
```



También es posible fijar el grosor de las líneas adicionales de forma independiente del de las líneas de la pauta.

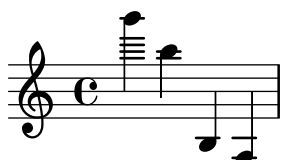
```
\new Staff \with {
  \override StaffSymbol.thickness = #2
  \override StaffSymbol.ledger-line-thickness = #'(0.5 . 0.4)
} \relative {
  f'''4 a, a,, f
}
```



El primer valor se multiplica por el grosor de las líneas del pentagrama, el segundo por el ancho de un espacio del pentagrama, y después los dos valores se suman para obtener el grosor de las líneas adicionales.

Se pueden alterar las posiciones de las líneas adicionales:

```
\new Staff \with {
  \override StaffSymbol.ledger-positions = #'(-3 -2 -1 2 5 6)
} \relative {
  f'''4 a, a,, f
}
```



Puede hacerse que las líneas adicionales añadidas aparezcan por encima o por debajo de la cabeza de las figuras, dependiendo de la posición actual relativa a otras cabezas de figura que tienen asimismo sus propias líneas adicionales.

```
\new Staff \with {
  \override StaffSymbol.ledger-extra = #4
} \relative {
  f'''4 a, d, f,
}
```



También puede hacerse que las líneas adicionales aparezcan dentro del pentagrama allí donde se requieren líneas de pauta personalizadas. El ejemplo muestra la posición predeterminada de las líneas adicionales cuando el valor de posición explícito `ledger-position` se ha fijado o no. La instrucción `\stopStaff` se necesita en el ejemplo para revertir la instrucción de sobreescritura `\override` para todo el `StaffSymbol` (el símbolo de pauta).

```
\relative d' {
  \override Staff.StaffSymbol.line-positions = #'(-8 0 2 4)
  d4 e f g
```

```

\stopStaff
\startStaff
\override Staff.StaffSymbol.ledger-positions = #'(-8 -6 (-4 -2) 0)
d4 e f g
}

```



Puede alterarse la distancia entre líneas de la pauta. Esto afecta también al espaciado de las líneas adicionales.

```

\new Staff \with {
  \override StaffSymbol.staff-space = #1.5
} \relative {
  f''4 d, g, e,
}

```



## Fragmentos de código seleccionados

### *Hacer unas líneas del pentagrama más gruesas que las otras*

Se puede engrosar una línea del pentagrama con fines pedagógicos (p.ej. la tercera línea o la de la clave de Sol). Esto se puede conseguir añadiendo más líneas muy cerca de la línea que se quiere destacar, utilizando la propiedad `line-positions` del objeto `StaffSymbol`.

```

{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}

```



## Véase también

Glosario musical: Sección “line” en *Glosario Musical*, Sección “ledger line” en *Glosario Musical*, Sección “staff” en *Glosario Musical*.

Referencia de la notación: [Clave], página 18.

Fragmentos de código: Sección “Staff notation” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “StaffSymbol” en *Referencia de Funcionamiento Interno*, Sección “staff-symbol-interface” en *Referencia de Funcionamiento Interno*.



## Pentagramas de Ossia

Los pentagramas *Ossia* se pueden preparar mediante la creación de un pentagrama simultáneo nuevo en la posición adecuada:

```
\new Staff \relative {
  c' '4 b d c
  <<
    { c4 b d c }
    \new Staff { e4 d f e }
  >>
  c4 b c2
}
```

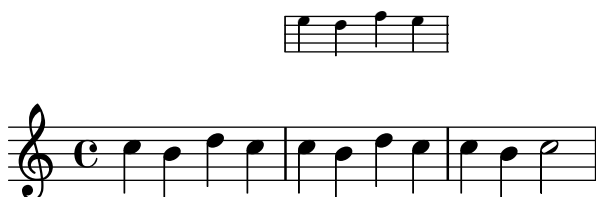


Sin embargo, el ejemplo anterior no es lo que normalmente se desea. Para crear pentagramas de ossia que estén encima del pentagrama original, que no tengan compás ni clave, y que tengan un tamaño menor de tipografía se deben usar varios trucos. El Manual de aprendizaje describe una técnica específica para llegar a este objetivo, empezando por Sección “Anidado de expresiones musicales” en *Manual de Aprendizaje*.

El ejemplo siguiente utiliza la propiedad `alignAboveContext` para alinear el pentagrama de ossia. Este método es muy conveniente cuando se necesitan sólo algunos pentagramas de ossia.

```
\new Staff = "main" \relative {
  c' '4 b d c
  <<
    { c4 b d c }

    \new Staff \with {
      \remove "Time_signature_engraver"
      alignAboveContext = "main"
      \magnifyStaff #2/3
      firstClef = ##f
    }
    { e4 d f e }
  >>
  c4 b c2
}
```



Si se requieren muchos pentagramas de ossia aislados, puede ser más conveniente la creación de un contexto `Staff` vacío con un *identificador de contexto* específico; después se pueden crear los pentagramas de ossia llamando a este contexto y usando `\startStaff` y `\stopStaff` en los

puntos deseados. Las ventajas de este método son más patentes si la pieza es más larga que en ejemplo siguiente.

```
<<
\new Staff = "ossia" \with {
  \remove "Time_signature_engraver"
  \hide Clef
  \magnifyStaff #2/3
}
{ \stopStaff s1*6 }

\new Staff \relative {
  c'4 b c2
  <<
    { e4 f e2 }
    \context Staff = "ossia" {
      \startStaff e4 g8 f e2 \stopStaff
    }
  >>
  g4 a g2 \break
  c4 b c2
  <<
    { g4 a g2 }
    \context Staff = "ossia" {
      \startStaff g4 e8 f g2 \stopStaff
    }
  >>
  e4 d c2
}
>>
```



4



Como alternativa, se puede usar la instrucción `\RemoveAllEmptyStaves` para crear pentagramas de ossia. Este método es muy conveniente cuando los pentagramas de ossia aparecen inmediatamente después de un salto de línea. Para ver más información sobre `\RemoveAllEmptyStaves`, consulte [Ocultar pentagramas], página 215.

```
<<
\new Staff = "ossia" \with {
  \remove "Time_signature_engraver"
  \hide Clef
```

```

\magnifyStaff #2/3
\RemoveAllEmptyStaves
} \relative {
  R1*3
  c' '4 e8 d c2
}
\new Staff \relative {
  c'4 b c2
  e4 f e2
  g4 a g2 \break
  c4 b c2
  g4 a g2
  e4 d c2
}
>>

```



## Fragmentos de código seleccionados

### *Alineación vertical de la letra y los compases de ossia*

Este fragmento de código muestra el uso de las propiedades de contexto `alignBelowContext` y `alignAboveContext` para controlar la posición de la letra y los compases de ossia.

```

\paper {
  ragged-right = ##t
}

\relative c' <<
  \new Staff = "1" { c4 c s2 }
  \new Staff = "2" { c4 c s2 }
  \new Staff = "3" { c4 c s2 }
  { \skip 2
    <<
      \lyrics {
        \set alignBelowContext = #"1"
        lyrics4 below
      }
      \new Staff \with {
        alignAboveContext = #"3"
        fontSize = #-2
        \override StaffSymbol.staff-space = #(magstep -2)
        \remove "Time_signature_engraver"
      }
    }
  }

```

```

    } {
      \tuplet 6/4 {
        \override TextScript.padding = #3
        c8[^"ossia above" d e d e f]
      }
    }
  >>
}
>>

```



## Véase también

Glosario musical: Sección “ossia” en *Glosario Musical*, Sección “staff” en *Glosario Musical*, Sección “Frenched staff” en *Glosario Musical*.

Manual de aprendizaje: Sección “Anidado de expresiones musicales” en *Manual de Aprendizaje*, Sección “Tamaño de los objetos” en *Manual de Aprendizaje*, Sección “Longitud y grosor de los objetos” en *Manual de Aprendizaje*.

Referencia de la notación: [Ocultar pentagramas], página 215.

Fragmentos de código: Sección “Staff notation” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “StaffSymbol” en *Referencia de Funcionamiento Interno*.

## Ocultar pentagramas

Se pueden ocultar las líneas del pentagrama quitando el grabador `Staff_symbol_engraver` del contexto de `Staff`. Como alternativa se puede utilizar la instrucción `\stopStaff`.

```

\new Staff \with {
  \remove "Staff_symbol_engraver"
}
\relative { a'8 f e16 d c b a2 }

```

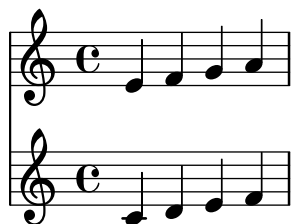


Se pueden ocultar los pentagramas vacíos (para hacer la que se conoce como ‘partitura a la francesa’) aplicando la instrucción `\RemoveEmptyStaves` sobre un contexto, lo cual se puede hacer globalmente (dentro del bloque `\layout`) así como para pautas específicas solamente (dentro de un bloque `\with`). Esta instrucción elimina todas las pautas vacías de la partitura

excepto las del primer sistema. Si queremos ocultar también las del primer sistema, usamos `\RemoveAllEmptyStaves`.

```
\layout {
  \context {
    \Staff
    \RemoveEmptyStaves
  }
}

\relative <<
  \new Staff {
    e'4 f g a \break
    b1 \break
    a4 b c2
  }
  \new Staff {
    c,4 d e f \break
    R1 \break
    f4 g c,2
  }
}>>
```



Se considera que un pentagrama está vacío cuando contiene solamente silencios multicompas, silencios normales, silencios oscultos o saltos, o una combinación de todos los elementos anteriores. *Cualquier otro* objeto musical (que hace que un pentagrama no se considere como vacío) está en la lista de la propiedad de contexto `keepAliveInterfaces`, como se establece inicialmente en el archivo `ly/engraver-init.ly`.

Tanto `\RemoveEmptyStaves` como `\RemoveAllEmptyStaves` son atajos predefinidos que establecen propiedades tales como `remove-empty` y `remove-first` para el objeto `VerticalAxisGroup`, como aparece explicado en Sección A.20 [Identificadores de modificación de contextos], página 856.

El grabador `Keep_alive_together_engraver` solo permite que grupos de pentagramas se puedan eliminar juntos y no individualmente. Por defecto, forma parte del contexto `PianoStaff`: una parte de piano solo se oculta cuando sus dos pentagramas están vacíos. De forma semejante, es una práctica común del tipografiado de las partituras orquestales eliminar grupos de pentagramas vacíos en lugar de pautas individuales; esto es posible conseguirlo añadiendo el grabador `Keep_alive_together_engraver` al contexto de agrupamiento de pentagramas adecuado, como aparece explicado en Sección 5.1.4 [Modificar los complementos (plug-ins) de contexto], página 631, (consulte [Agrupar pentagramas], página 202, para ver los nombres de los contextos).

```
\layout {
  \context {
    \StaffGroup
    \RemoveEmptyStaves
    \consists "Keep_alive_together_engraver"
  }
}
```

En el ejemplo siguiente, los pentagramas de instrumentos de viento son eliminados en el segundo sistema; sin embargo, el contrabajo no se elimina porque forma parte del grupo mayor de cuerda frotada, que está tocando en ese momento.

The image displays a musical score for an orchestra, illustrating the effect of the `Keep_alive_together_engraver` plugin. The score is written in 3/4 time with a key signature of two flats (B-flat and E-flat). The instruments listed on the left are Flute, Oboe, Basson, Violin I, Violin II, Alto, Cello, and Double bass. The score is divided into three systems. In the first system, all instruments have musical notation. In the second system, the staves for Flute, Oboe, and Basson are empty, while the string instruments (Violin I, Violin II, Alto, Cello, and Double bass) continue to play. In the third system, all instruments have musical notation again. This demonstrates how the plugin groups empty staves of instruments that are part of the same larger group (in this case, the wind section) for removal, while preserving staves that are part of a different group (the string section) that is still active.

El grabador `Keep_alive_together_engraver` usa internamente la propiedad `remove-layer` del `VerticalAxisGroup` de un pentagrama para decidir si imprimirlo o no, cuando se considera vacío. Dicha propiedad puede también establecerse directamente, en cuyo caso actúa como un índice de prioridad: los valores más cercanos a cero tienen precedencia frente a los números más altos, y de esta forma los pentagramas cuyo `remove-layer` es más alto quedarán enmascarados en favor de los que tienen un número más bajo.

Esto es especialmente útil para los pentagramas en ‘divisi’, donde ciertas partes individuales (véase Sección 1.6.3 [Escritura de las particellas], página 220) a veces tienen que expandirse a más de un pentagrama. En el ejemplo siguiente, dos partes resultan dirigidas a un total de *tres* pentagramas; sin embargo, los tres pentagramas nunca se imprimen todos al mismo tiempo:

- en los primeros sistemas, solo uno de ellos se imprime, porque la propiedad `keepAliveInterfaces` se ha establecido a una lista vacía; por tanto, los otros dos pentagramas se considera que están vacíos y por ello ocultos, sin que importe su contenido;
- cuando la propiedad resulta des-establecida (volviendo así a su valor predeterminado), ya no impide que los otros dos pentagramas se impriman; sin embargo, como el valor de `remove-layer` de éstos es menor que el del pentagrama suelto, estos dos pentagramas se imprimen ahora en su lugar.

Tales sustituciones se aplican no solamente a notas, acordes y otros eventos musicales que tienen lugar inmediatamente después de cambiar el valor, sino a todo el sistema dentro del cual ocurre el cambio.

```
\layout {
  short-indent = 2\cm
  indent = 3\cm
  \context {
    \Staff
    keepAliveInterfaces = #'()
  }
}

violI = {
  \repeat unfold 24 { d'4 }
  \once \unset Staff.keepAliveInterfaces
  <d' g''>2
  \repeat unfold 14 { d'4 }
  \bar "|."
}
```

```

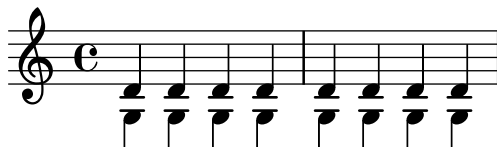
}

violIII = {
  \repeat unfold 24 { g4 }
  <g d'>2
  \repeat unfold 14 { g4 }
  \bar "|."
}

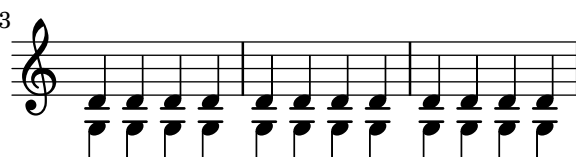
\new StaffGroup \with { \consists "Keep_alive_together_engraver" } <<
  \new Staff \with {
    instrumentName = "Violins"
    shortInstrumentName = "V I & II"
    \override VerticalAxisGroup.remove-layer = 2
  } << \violI \\\violIII >>
  \new Staff \with {
    instrumentName = "Violin I"
    shortInstrumentName = "V I"
    \RemoveAllEmptyStaves
    \override VerticalAxisGroup.remove-layer = 1
  } \violI
  \new Staff \with {
    instrumentName = "Violin II"
    shortInstrumentName = "V II"
    \RemoveAllEmptyStaves
    \override VerticalAxisGroup.remove-layer = 1
  } \violII
>>

```


Violins



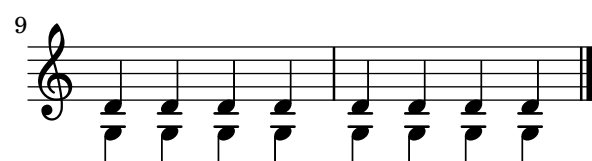
V I & II



V I



V II





También se puede usar `\RemoveAllEmptyStaves` para crear secciones de ossia para un pentagrama. Para ver más detalles, consulte [Pentagramas de Ossia], página 212.

## Instrucciones predefinidas

`\RemoveEmptyStaves`, `\RemoveAllEmptyStaves`.

## Véase también

Glosario musical: Sección “Frenched staff” en *Glosario Musical*.

Manual de aprendizaje: Sección “Visibilidad y color de los objetos” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.1.5 [Cambiar los valores por omisión de los contextos], página 633, [El símbolo del pentagrama], página 208, [Pentagramas de Ossia], página 212, [Notas ocultas], página 242, [Silencios invisibles], página 64, Sección 5.4.7 [Visibilidad de los objetos], página 670, Sección A.20 [Identificadores de modificación de contextos], página 856, [Agrupar pentagramas], página 202, Sección 5.1.4 [Modificar los complementos (plug-ins) de contexto], página 631.

Archivos instalados: `ly/engraver-init.ly`.

Fragmentos de código: Sección “Staff notation” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “ChordNames” en *Referencia de Funcionamiento Interno*, Sección “FiguredBass” en *Referencia de Funcionamiento Interno*, Sección “Lyrics” en *Referencia de Funcionamiento Interno*, Sección “Staff” en *Referencia de Funcionamiento Interno*, Sección “VerticalAxisGroup” en *Referencia de Funcionamiento Interno*, Sección “Staff-symbol-engraver” en *Referencia de Funcionamiento Interno*, Sección “Axis-group-engraver” en *Referencia de Funcionamiento Interno*, Sección “Keep-alive-together-engraver” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

La eliminación del grabador `Staff_symbol_engraver` también oculta las barras de compás. Si se fuerza la visibilidad de la barra de compás, pueden ocurrir errores de formato visual. En este caso, utilice las siguientes sobreescrituras de valores en vez de quitar el grabador:

```
\omit StaffSymbol
\override NoteHead.no-ledgers = ##t
```

Para ver los fallos y problemas conocidos, así como las advertencias asociadas con `\RemoveEmptyStaves`, consulte Sección 5.1.5 [Cambiar los valores por omisión de los contextos], página 633.

### 1.6.3 Escritura de las particellas

Esta sección explica cómo preparar el papel de atril, o particellas, para música orquestal o de conjunto, lo que frecuentemente requiere insertar nombres de instrumentos en la partitura. También se describen métodos para citar otras voces y dar formato a las notas guía, así como una manera de contraer varios compases vacíos consecutivos dentro de las partes individuales.

Además, puede verse un método para la impresión de los pentagramas de *divisi*, que se usan a veces en las particellas, en [Ocultar pentagramas], página 215.

## Nombres de instrumentos

Se pueden imprimir los nombres de los instrumentos en el lado izquierdo de los pentagramas dentro de los contextos `Staff`, `PianoStaff`, `StaffGroup`, `GrandStaff` y `ChoirStaff`. El valor de `instrumentName` se usa para el primer pentagrama, y el valor de `shortInstrumentName` se usa para todos los pentagramas siguientes.

```
\new Staff \with {
```

```

instrumentName = "Violin "
shortInstrumentName = "Vln. "
} \relative {
  c'4.. g'16 c4.. g'16 \break | c1 |
}

```



También podemos usar `\markup` para construir nombres de instrumento más complicados:

```

\new Staff \with {
  instrumentName = \markup {
    \column { "Clarinetti"
      \line { "in B" \smaller \flat }
    }
  }
} \relative {
  c''4 c,16 d e f g2
}

```



Cuando se agrupan dos o más contextos de pentagrama, los nombres de instrumento y los nombres cortos aparecen centrados de forma predeterminada. Para centrar nombres de instrumento de varias líneas, se debe utilizar `\center-column`:

```

<<
  \new Staff \with {
    instrumentName = "Flute"
  } \relative {
    f''2 g4 f
  }
  \new Staff \with {
    instrumentName = \markup {
      \center-column { "Clarinet"
        \line { "in B" \smaller \flat }
      }
    }
  } \relative { c''4 b c2 }
>>

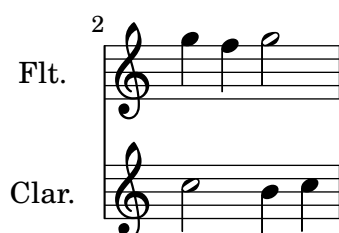
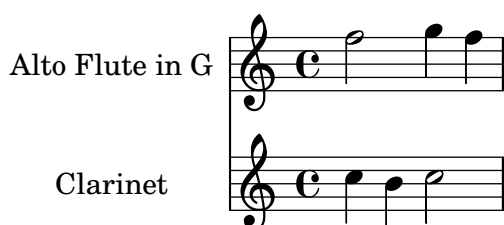
```



Sin embargo, si los nombres de instrumento son más largos, éstos no se centran para un grupo de pentagramas a no ser que se aumenten los valores del sangrado, `indent`, y del sangrado corto, `short-indent`. Para ver más detalles sobre estos ajustes, consulte [Variables de `\paper` para desplazamientos y sangrados], página 575.

```
<<
  \new Staff \with {
    instrumentName = "Alto Flute in G"
    shortInstrumentName = "Flt."
  } \relative {
    f''2 g4 f \break
    g4 f g2
  }
  \new Staff \with {
    instrumentName = "Clarinet"
    shortInstrumentName = "Clar."
  } \relative {
    c''4 b c2 \break
    c2 b4 c
  }
>>

\layout {
  indent = 3.0\cm
  short-indent = 1.5\cm
}
```



Para añadir nombres de instrumento a otros contextos (como `ChordNames` o `FiguredBass`), debemos añadir el grabador `Instrument_name_engraver` a dicho contexto. Para ver más detalles, consulte Sección 5.1.4 [Modificar los complementos (plug-ins) de contexto], página 631.

El nombre de instrumento corto `shortInstrumentName` se puede cambiar en mitad de una pieza, igual que otros ajustes que pueden ser necesarios para el instrumento nuevo; sin embargo, sólo el primer `instrumentName` se imprime y el resto de ellos se ignoran:

```
prepPiccolo = <>^\markup \italic { muta in Piccolo }

prepFlute = <>^\markup \italic { muta in Flauto }
```

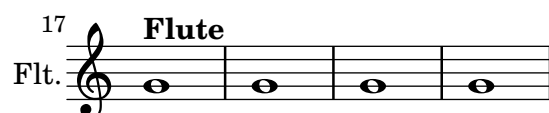
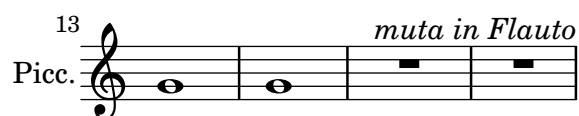
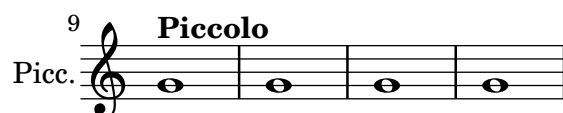
```

setPiccolo = {
  <>^\markup \bold { Piccolo }
  \transposition c''
}

setFlute = {
  <>^\markup \bold { Flute }
  \transposition c'
}

\new Staff \with {
  instrumentName = "Flute"
  shortInstrumentName = "Flt."
}
\relative {
  g'1 g g g \break
  g1 g \prepPiccolo R R \break
  \set Staff.instrumentName = "Piccolo"
  \set Staff.shortInstrumentName = "Picc."
  \setPiccolo
  g1 g g g \break
  g1 g \prepFlute R R \break
  \set Staff.instrumentName = "Flute"
  \set Staff.shortInstrumentName = "Flt."
  \setFlute
  g1 g g g
}

```



## Véase también

Referencia de la notación: [Variables de `\paper` para desplazamientos y sangrados], página 575, Sección 5.1.4 [Modificar los complementos (plug-ins) de contexto], página 631.

Fragmentos de código: Sección “Staff notation” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “InstrumentName” en *Referencia de Funcionamiento Interno*, Sección “PianoStaff” en *Referencia de Funcionamiento Interno*, Sección “Staff” en *Referencia de Funcionamiento Interno*.

## Citar otras voces

Es muy frecuente que una voz use las mismas notas que otra voz. Por ejemplo, los violines primero y segundo tocando la misma frase durante un determinado pasaje musical. Esto se hace dejando que una voz *cite* a la otra, sin tener que volver a introducir la música para la segunda voz.

La instrucción `\addQuote`, utilizada en el ámbito del nivel sintáctico superior, define un flujo de música desde el que es posible citar fragmentos.

La instrucción `\quoteDuring` se usa para indicar el punto en que comienza la cita. Va seguida por dos argumentos: el nombre de la voz citada, tal y como se definió con `\addQuote`, y una expresión musical que indica la duración de la cita.

```
fluteNotes = \relative {
  a'4 gis g gis | b4^"quoted" r8 ais\p a4( f)
}

oboeNotes = \relative {
  c'4 cis c b \quoteDuring "flute" { s1 }
}

\addQuote "flute" { \fluteNotes }

\score {
  <<
    \new Staff \with { instrumentName = "Flute" } \fluteNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}
```



Si la expresión musical que se usa para `\quoteDuring` contiene cualquier cosa que no sea un silencio de separación o un silencio multicompa, se produce una situación de polifonía, lo que no suele ser deseable:

```
fluteNotes = \relative {
  a'4 gis g gis | b4^"quoted" r8 ais\p a4( f)
}
```

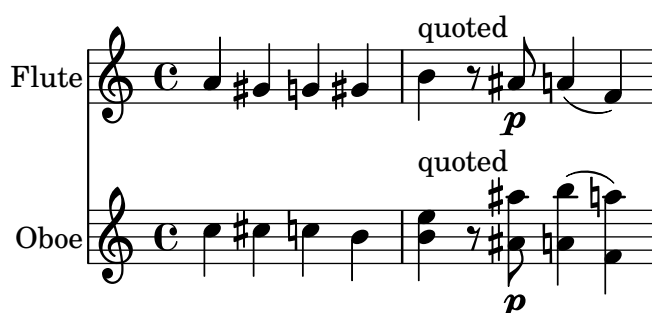
```

oboeNotes = \relative {
  c' '4 cis c b \quoteDuring "flute" { e4 r8 ais b4 a }
}

\addQuote "flute" { \fluteNotes }

\score {
  <<
    \new Staff \with { instrumentName = "Flute" } \fluteNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}

```



Si una instrucción `\unfoldRepeats` dentro de una expresión musical se necesita imprimir al usar `\quoteDuring`, entonces debe contener también su propia instrucción `\unfoldRepeats`;

```

fluteNotes = \relative {
  \repeat volta 2 { a'4 gis g gis }
}

oboeNotesDW = \relative {
  \repeat volta 2 \quoteDuring "incorrect" { s1 }
}

oboeNotesW = \relative {
  \repeat volta 2 \quoteDuring "correct" { s1 }
}

\addQuote "incorrect" { \fluteNotes }

\addQuote "correct" { \unfoldRepeats \fluteNotes }

\score {
  \unfoldRepeats
  <<
    \new Staff \with { instrumentName = "Flute" }
    \fluteNotes
    \new Staff \with { instrumentName = "Oboe (incorrect)" }
    \oboeNotesDW
    \new Staff \with { instrumentName = "Oboe (correct)" }
    \oboeNotesW
  >>
}

```

}

Flute

Oboe (incorrect)

Oboe (correct)

La instrucción `\quoteDuring` usa los ajustes de `\transposition` tanto de la parte citada como de la que cita, para producir notas que tienen la misma altura de sonido en la parte que cita y en las de la parte citada.

```
clarinetNotes = \relative c'' {
  \transposition bes
  \key d \major
  b4 ais a ais | cis4^"quoted" r8 bis\p b4( f)
}

oboeNotes = \relative {
  c''4 cis c b \quoteDuring "clarinet" { s1 }
}

\addQuote "clarinet" { \clarinetNotes }

\score {
  <<
    \new Staff \with { instrumentName = "Clarinet" } \clarinetNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}
```

Clarinet

Oboe

De forma predeterminada, la música citada incluye todas las articulaciones, matices dinámicos, elementos de marcado, etc. de la expresión citada. Es posible elegir cuáles de estos objetos de la música que se cita se imprimen, mediante la propiedad de contexto `quotedEventTypes`.

```
fluteNotes = \relative {
  a'2 g2 |
  b4\<^"quoted" r8 ais a4\f( c->)
```

```

}

oboeNotes = \relative {
  c''2. b4 |
  \quoteDuring "flute" { s1 }
}

\addQuote "flute" { \fluteNotes }

\score {
  <<
    \set Score.quotedEventTypes = #'(note-event articulation-event
                                     crescendo-event rest-event
                                     slur-event dynamic-event)
    \new Staff \with { instrumentName = "Flute" } \fluteNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}

```



Las citas también se pueden etiquetar, véase [Uso de etiquetas], página 541.

## Véase también

Referencia de la notación: [Transposición de los instrumentos], página 28, [Uso de etiquetas], página 541.

Archivos de inicio: `scm/define-event-classes.scm`.

Fragmentos de código: Sección “Staff notation” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Music classes” en *Referencia de Funcionamiento Interno*, Sección “QuoteMusic” en *Referencia de Funcionamiento Interno*, Sección “Voice” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Sólo el contenido de la primera voz de una instrucción `\addQuote` se tiene en cuenta para la cita, de manera que si la expresión musical contiene enunciados `\new` o `\context Voice`, sus contenidos no serían citados. La cita de notas de adorno no está contemplada y puede hacer que LilyPond termine de forma abrupta; la cita de unos tresillos dentro de otros puede dar como resultado una notación de pobre calidad.

## Formateo de las notas guía

La manera más sencilla de dar formato a las notas guía es crear explícitamente un contexto `CueVoice` dentro de la parte.

```

\relative {
  R1
  <<

```



```

{ e'2\rest r4. e8 }
\new CueVoice {
  \stemUp d'8^"flute" c d e fis2
}
>>
d,,4 r a r
}

```



La instrucción `\cueClef` puede usarse también con un contexto `CueVoice` explícito si se requiere un cambio de clave, e imprime una clave del tamaño adecuado para las notas guía. Después puede utilizarse la instrucción `\cueClefUnset` para volver a la clave original, de nuevo con un signo de clave del tamaño adecuado.

```

\relative {
  \clef "bass"
  R1
  <<
    { e'2\rest r4. \cueClefUnset e,8 }
    \new CueVoice {
      \cueClef "treble" \stemUp d''8^"flute" c d e fis2
    }
  >>
  d,,4 r a r
}

```



Las instrucciones `\cueClef` y `\cueClefUnset` se pueden usar también sin un `CueVoice` si es necesario.

```

\relative {
  \clef "bass"
  R1
  \cueClef "treble"
  d''8^"flute" c d e fis2
  \cueClefUnset
  d,,4 r a r
}

```



En casos de colocación más compleja de notas guía, por ejemplo con transposición o insertando notas guía procedentes de más de una fuente, pueden usarse las instrucciones `\cueDuring` o `\cueDuringWithClef`. Son una forma más especializada de `\quoteDuring`, véase [Citar otras voces], página 224, en la sección anterior.



```
\cueDuring "oboe" #UP { R1 }
g2 c,
}
```



Se pueden usar elementos de marcado para mostrar el nombre del instrumento citado. Si las notas guía requieren un cambio de clave, puede hacerse manualmente pero la clave original se debe restaurar al final de las notas guía.

```
fluteNotes = \relative {
  r2. c' '4 d8 c d e fis2 g2 d2
}

bassoonNotes = \relative c {
  \clef bass
  R1
  \clef treble
  <>^\markup \tiny { flute }
  \cueDuring "flute" #UP { R1 }
  \clef bass
  g4. b8 d2
}

\addQuote "flute" { \fluteNotes }

\new Staff {
  \bassoonNotes
}
```



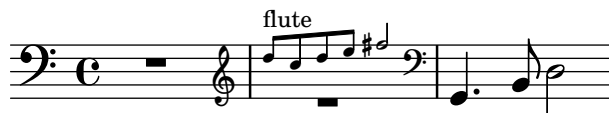
De forma alternativa, puede usarse la función `\cueDuringWithClef` en su lugar. Esta instrucción admite un argumento adicional para especificar el cambio de clave que se necesita imprimir para las notas guía, pero después imprime automáticamente la clave original una vez que ha finalizado la serie de notas guía.

```
fluteNotes = \relative {
  r2. c' '4 d8 c d e fis2 g2 d2
}

bassoonNotes = \relative c {
  \clef bass
  R1
  <>^\markup { \tiny "flute" }
  \cueDuringWithClef "flute" #UP "treble" { R1 }
  g4. b8 d2
}
```

```
\addQuote "flute" { \fluteNotes }
```

```
\new Staff {
  \bassoonNotes
}
```



Como `\quoteDuring`, `\cueDuring` tiene en cuenta las transposiciones instrumentales. Las notas guía se producen en las alturas en que se escribirían para el instrumento que recibe la cita, para así producir las alturas de sonido del instrumento original.

Para transportar las notas guía de forma diferente, use `\transposedCueDuring`. Esta instrucción acepta un argumento adicional para especificar (en modo absoluto) la altura impresa con que queremos representar el sonido de un Do central de concierto. Esto es útil para extraer citas de un instrumento que está en un registro completamente diferente.

```
piccoloNotes = \relative {
  \clef "treble^8"
  R1
  c''^8 c c e g2
  c4 g g2
}
```

```
bassClarinetNotes = \relative c' {
  \key d \major
  \transposition bes,
  d4 r a r
  \transposedCueDuring "piccolo" #UP d { R1 }
  d4 r a r
}
```

```
\addQuote "piccolo" { \piccoloNotes }
```

```
<<
  \new Staff \piccoloNotes
  \new Staff \bassClarinetNotes
>>
```



La instrucción `\killCues` elimina las notas guía de una expresión musical, de forma que la misma expresión musical pueda utilizarse para producir la particella instrumental (con notas guía) y la partitura de conjunto. La instrucción `\killCues` elimina solamente las notas y eventos que se han citado mediante `\cueDuring`. Otros elementos de marcado asociados con las guías,

como los cambios de clave o una etiqueta identificativa del instrumento fuente, pueden marcarse para su inclusión selectiva dentro de la partitura; véase [Uso de etiquetas], página 541.

```
fluteNotes = \relative {
  r2. c''4 d8 c d e fis2 g2 d2
}

bassoonNotes = \relative c {
  \clef bass
  R1
  \tag #'part {
    \clef treble
    <>\markup \tiny { flute }
  }
  \cueDuring "flute" #UP { R1 }
  \tag #'part \clef bass
  g4. b8 d2
}

\addQuote "flute" { \fluteNotes }

\new Staff {
  \bassoonNotes
}

\new StaffGroup <<
  \new Staff {
    \fluteNotes
  }
  \new Staff {
    \removeWithTag #'part { \killCues { \bassoonNotes } }
  }
>>
```



## Véase también

Referencia de la notación: [Citar otras voces], página 224, [Transposición de los instrumentos], página 28, [Nombres de instrumentos], página 220, [Clave], página 18, [Guías musicales], página 334, [Uso de etiquetas], página 541.

Fragmentos de código: Sección “Staff notation” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “CueVoice” en *Referencia de Funcionamiento Interno*, Sección “Voice” en *Referencia de Funcionamiento Interno*.

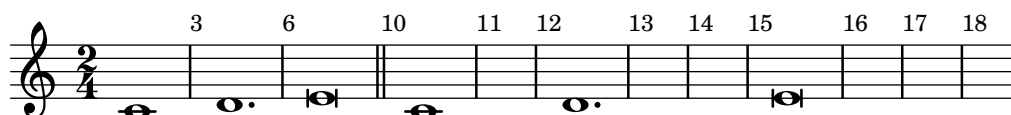
## Advertencias y problemas conocidos

Pueden ocurrir colisiones con los silencios si se usa `\cueDuring`, entre los contextos de `Voice` y de `CueVoice`. Al usar `\cueDuringWithClef` o `\transposedCueDuring` el argumento adicional requerido para cada caso debe ir después de la cita y de la dirección.

## Compresión de los compases vacíos

De forma predeterminada, todos los compases se imprimen aunque estén vacíos (cosa que puede ocurrir si un evento rítmico, como notas, silencios o saltos invisibles, es tan largo que abarca varios compases. Este comportamiento se puede cambiar contrayendo todos los compases vacíos en uno solo, como se ilustra aquí (la segunda parte del ejemplo, con compases expandidos, en realidad vuelve de nuevo al comportamiento predeterminado):

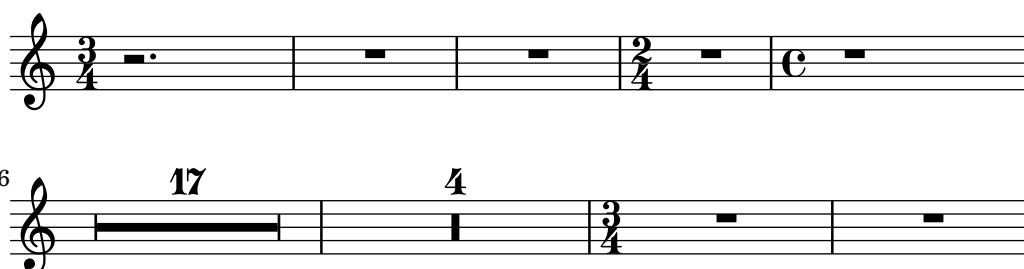
```
\override Score.BarNumber.break-visibility = ##(f #t #t)
\time 2/4
\compressEmptyMeasures
c'1 d'1. e'\breve
\bar "||"
\expandEmptyMeasures
c'1 d'1. e'\breve
```



Aunque esta notación es correcta sintácticamente, puede dar lugar a confusión desde el punto de vista musical, como queda ilustrado en el ejemplo anterior; de ahí la necesidad de imprimir explícitamente los números de compás, usando la sintaxis que se describe en [Uso de `break-visibility` (visibilidad en el salto)], página 671.

Donde esta notación puede ser más útil es cuando se aplica a los [Silencios de compás completo], página 65. Un silencio multicompaás se imprimirá entonces como un solo compás que contiene un símbolo de silencio multicompaás, con el número de compases de duración de este silencio impreso encima del compás:

```
% Default behavior
\time 3/4 r2. | R2.*2 |
\time 2/4 R2 |
\time 4/4
% Rest measures contracted to single measure
\compressEmptyMeasures
r1 | R1*17 | R1*4 |
\expandEmptyMeasures
% Rest measures expanded again
\time 3/4
R2.*2 |
```



A diferencia de `\compressEmptyMeasures`, la función musical `\compressMMRests` solo se aplica a los silencios, dejando sin comprimir todos los demás eventos. Dado que es una función, y no un ajuste del valor de una propiedad, su sintaxis difiere ligeramente en que debe ir seguida de una expresión musical:

```
\compressMMRests {
  % Rests are compressed...
  R1*7
  % ... but notes can still span multiple measures.
  g'1 a'1*2 d'1
  R1*2
}
```



Todas las instrucciones descritas en esta sección, descansan en realidad en la propiedad interna `skipBars`, que se establece dentro del contexto de `Score` como se explica en Sección 5.3.2 [La instrucción `\set`], página 646.

## Instrucciones predefinidas

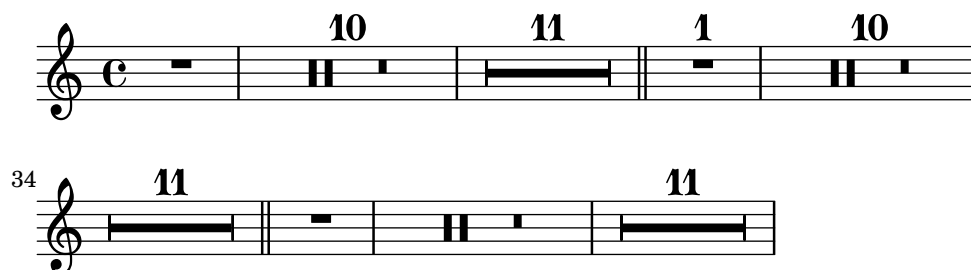
`\compressEmptyMeasures`, `\expandEmptyMeasures`, `\compressMMRests`.

## Fragmentos de código seleccionados

### *Numbering single measure rests*

Multi measure rests show their length by a number except for single measures. This can be changed by setting `restNumberThreshold`.

```
{
  \compressEmptyMeasures
  R1 R1*10 R1*11 \bar "||"
  \set restNumberThreshold = 0
  R1 R1*10 R1*11 \bar "||"
  \set restNumberThreshold = 10
  R1 R1*10 R1*11
}
```



### *Cambiar la forma de los silencios multicompás*

Si hay diez compases de silencio o menos, se imprime en el pentagrama una serie de silencios de breve y longa (conocidos en alemán como “Kirchenpausen”, «silencios eclesiásticos»); en caso contrario se muestra una barra normal. Este número predeterminado de diez se puede cambiar sobrescribiendo la propiedad `expand-limit`:

```
\relative c'' {
```

```

\compressMMRests {
  R1*2 | R1*5 | R1*9
  \override MultiMeasureRest.expand-limit = #3
  R1*2 | R1*5 | R1*9
}

```

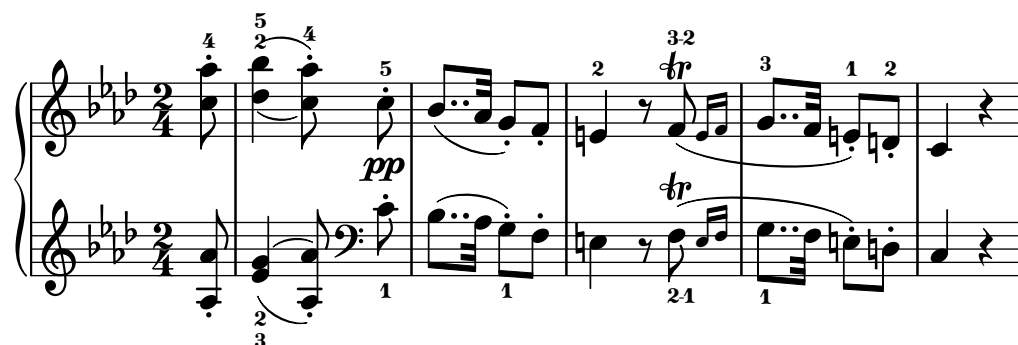


## Véase también

Referencia de la notación: [Uso de break-visibility (visibilidad en el salto)], página 671, [Silencios de compás completo], página 65, Sección 5.3.2 [La instrucción `\set`], página 646.

Referencia de funcionamiento interno: Sección “MultiMeasureRest” en *Referencia de Funcionamiento Interno*, Sección “MultiMeasureRestNumber” en *Referencia de Funcionamiento Interno*, Sección “MultiMeasureRestScript” en *Referencia de Funcionamiento Interno*, Sección “MultiMeasureRestText” en *Referencia de Funcionamiento Interno*.

## 1.7 Anotaciones editoriales



Esta sección trata de las diversas maneras de modificar el aspecto de las notas y de aplicar énfasis analítico o educativo.

### 1.7.1 Dentro del pentagrama

Esta sección trata sobre cómo aplicar énfasis a los elementos situados dentro del pentagrama.

## Seleccionar el tamaño de la tipografía para la notación



**Nota:**

Para el tamaño de la tipografía del texto, consulte [Seleccionar la tipografía y su tamaño], página 266.

Para el tamaño del pentagrama, consulte Sección 4.2.2 [Establecer el tamaño del pentagrama], página 581.

Para las notas guía, consulte [Formateo de las notas guía], página 227.

Para los pentagramas de Ossia, consulte [Pentagramas de Ossia], página 212.

Para modificar el tamaño de la notación sin cambiar el tamaño del pentagrama, especifique un factor de ampliación con la instrucción `\magnifyMusic`:

```
\new Staff <<
  \new Voice \relative {
    \voiceOne
    <e' e'>4 <f f'>8. <g g'>16 <f f'>8 <e e'>4 r8
  }
  \new Voice \relative {
    \voiceTwo
    \magnifyMusic 0.63 {
      \override Score.SpacingSpanner.spacing-increment = #(* 1.2 0.63)
      r32 c'' a c a c a c r c a c a c a c
      r c a c a c a c a c a c a c a c
    }
  }
>>
```

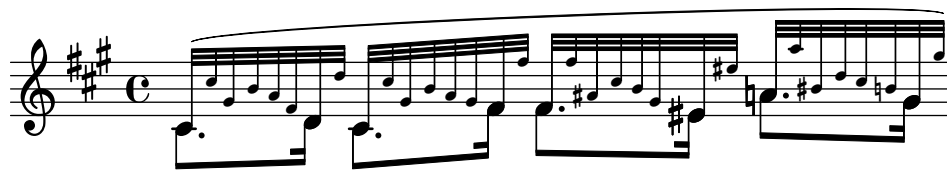


El `\override` del ejemplo anterior es una solución provisional para solventar un fallo del programa. Consulte el apartado de “Problemas y advertencias conocidos” al final de esta sección.

Si una figura con un tamaño normal de cabeza se mezcla con otra más pequeña, el tamaño de la menor podría necesitar reiniciarse (con `\once \normalsize`) para que las plicas y las alteraciones accidentales mantengan una buena alineación:

```
\new Staff <<
  \key fis \minor
  \mergeDifferentlyDottedOn
  \new Voice \relative {
    \voiceOne
    \magnifyMusic 0.63 {
      \override Score.SpacingSpanner.spacing-increment = #(* 1.2 0.63)
      \once \normalsize cis'32( cis' gis b a fis \once \normalsize d d'
      \once \normalsize cis, cis' gis b a gis \once \normalsize fis fis'
      \once \normalsize fis, fis' ais, cis b gis \once \normalsize eis eis'
      \once \normalsize a, a' bis, d cis b \once \normalsize gis gis')
    }
  }
  \new Voice \relative {
    \voiceTwo
    cis'8. d16 cis8. fis16 fis8. eis16 a8. gis16
  }
>>
```

&gt;&gt;



La instrucción `\magnifyMusic` no está pensada para las notas guía, de adorno, o para los pentagramas de Ossia; existen métodos más apropiados para la introducción de cada uno de esos elementos. En vez de ello, es útil cuando el tamaño de la notación cambia dentro de una sola parte instrumental sobre un pentagrama, y cuando las notas de adorno no son adecuadas, como en pasajes de tipo cadencial o en casos como los de los ejemplos anteriores. Establecer el valor de `\magnifyMusic` a 0.63 duplica las dimensiones del contexto `CueVoice`.

**Nota:** La instrucción `\magnifyMusic` *no* debe utilizarse si se está cambiando el tamaño del pentagrama al mismo tiempo. Consulte Sección 4.2.2 [Establecer el tamaño del pentagrama], página 581.

### *Cambiar el tamaño de los objetos individuales*

Un objeto de presentación individual se puede cambiar de tamaño usando las instrucciones `\tweak` u `\override` para ajustar su propiedad `font-size`:

```
\relative {
  % resize a note head
  <f' \tweak font-size -4 b e>-5
  % resize a fingering
  bes-\tweak font-size 0 -3
  % resize an accidental
  \once \override Accidental.font-size = -4 bes!-^
  % resize an articulation
  \once \override Script.font-size = 4 bes!-^
}
```



El valor predeterminado de `font-size` para cada objeto de presentación está relacionado en el Manual de Referencia de funcionamiento interno. La propiedad `font-size` solamente se puede fijar para los objetos de presentación que contemplan la interfaz de presentación `font-interface`. Si no está especificado `font-size` en la lista de ‘Ajustes estándar’ del objeto, su valor es 0. Véase Sección “All layout objects” en *Referencia de Funcionamiento Interno*.

### *Comprensión de la propiedad `fontSize`*

La propiedad de contexto `fontSize` ajusta el tamaño relativo de todos los elementos de notación basados en glifos dentro de un contexto:

```
\relative {
  \time 3/4
  d'4---5 c8( b a g) |
  \set fontSize = -6
  e'4-- c!8-4( b a g) |
}
```

```
\set fontSize = 0
fis4---3 e8( d) fis4 |
g2.
}
```



El valor de `fontSize` es un número que indica el tamaño relativo al tamaño estándar para la altura actual del pentagrama. El valor predeterminado de `fontSize` es 0; la adición de 6 a cualquier valor de `fontSize` duplica el tamaño impreso de los glifos, y la sustracción de 6 reduce el tamaño a la mitad. Cada paso aumenta el tamaño en un 12% aproximadamente.

La función de Scheme `magnification->font-size` se provee por conveniencia, pues las unidades logarítmicas de la propiedad `font-size` no son totalmente intuitivas. Por ejemplo, para ajustar la notación musical al 75% del tamaño predeterminado, usamos:

```
\set fontSize = #(magnification->font-size 0.75)
```

La función de Scheme `magstep` tiene el efecto opuesto: convierte un valor de `font-size` en un factor de ampliación.

La propiedad `fontSize` no afecta solamente a los elementos notacionales que se dibujan con glifos, como la cabeza de las figuras, las alteraciones accidentales, las inscripciones textuales, etc. No afecta al tamaño de la propia pauta, ni cambia la escala de las plicas, barras, o espaciado horizontal. Para cambiar la escala de las plicas, barras y el espaciado horizontal junto con el tamaño de la notación (sin cambiar el tamaño de la pauta), use la instrucción `\magnifyMusic` que se ha explicado anteriormente. Para cambiar la escala del tamaño de todo, incluida la pauta, véase Sección 4.2.2 [Establecer el tamaño del pentagrama], página 581.

Siempre que se establece el valor de la *propiedad de contexto* `fontSize`, su valor se añade al valor de la *propiedad de objeto gráfico* `font-size` para los objetos de presentación individuales, antes de que se imprima cualquiera de los glifos. Esto puede causar confusión cuando se fija el valor de las propiedades `font-size` individuales mientras `fontSize` ya está fijado:

```
% the default font-size for NoteHead is 0
% the default font-size for Fingering is -5
c''4-3

\set fontSize = -3
% the effective font size for NoteHead is now -3
% the effective font size for Fingering is now -8
c''4-3

\override Fingering.font-size = 0
% the effective font size for Fingering is now -3
c''4-3
```



Las siguientes instrucciones de atajo también están disponibles:

Instrucción	Equivalente a	Tamaño relativo
<code>\teeny</code>	<code>\set fontSize = -3</code>	71%
<code>\tiny</code>	<code>\set fontSize = -2</code>	79%

<code>\small</code>	<code>\set fontSize = -1</code>	89%
<code>\normalsize</code>	<code>\set fontSize = 0</code>	100%
<code>\large</code>	<code>\set fontSize = 1</code>	112%
<code>\huge</code>	<code>\set fontSize = 2</code>	126%

```

\relative c'' {
  \teeny
  c4.-> d8---3
  \tiny
  c4.-> d8---3
  \small
  c4.-> d8---3
  \normalsize
  c4.-> d8---3
  \large
  c4.-> d8---3
  \huge
  c4.-> d8---3
}

```



Los cambios en el tamaño de la fuente se obtienen por medio del escalado del tamaño del diseño que se encuentra más cerca del tamaño deseado. El tamaño estándar para la tipografía (para `font-size = 0`) depende de la altura estándar del pentagrama. Para un pentagrama de 20 puntos, se selecciona una fuente de 11 puntos.

## Instrucciones predefinidas

`\magnifyMusic`, `\teeny` (enano), `\tiny` (muy pequeño), `\small` (pequeño), `\normalsize` (normal), `\large` (grande), `\huge` (enorme).

## Véase también

Notation Reference: [Seleccionar la tipografía y su tamaño], página 266, Sección 4.2.2 [Establecer el tamaño del pentagrama], página 581, [Formateo de las notas guía], página 227, [Pentagramas de Ossia], página 212.

Installed Files: `ly/music-functions-init.ly`, `ly/property-init.ly`.

Fragmentos de código: Sección “Editorial annotations” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “font-interface” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Actualmente hay dos fallos del programa que impiden un correcto espaciado horizontal al usar `\magnifyMusic`. Solamente existe una forma de esquivarlo, y su funcionamiento correcto no está garantizado en todos los casos. En el ejemplo que aparece a continuación, sustituya la variable `mag` con su valor preferido. También puede tratar de quitar una o ambas instrucciones `\newSpacingSection`, y/o las instrucciones `\override` y `\revert`:

```

\magnifyMusic mag {
  \newSpacingSection
  \override Score.SpacingSpanner.spacing-increment = #(* 1.2 mag)
}

```

```
[music]
\newSpacingSection
\revert Score.SpacingSpanner.spacing-increment
}
```

## Indicaciones de digitación

Las instrucciones de digitación se pueden introducir usando ‘*nota-dígito*’:

```
\relative { c''4-1 d-2 f-4 e-3 }
```



Para los cambios de dedo se pueden usar elementos de marcado de texto o de cadenas de caracteres.

```
\relative {
  c''4-1 d-2 f\finger \markup \tied-lyric "4~3" c\finger "2 - 3"
}
```



Puede usar la digitación de pulgar para indicar que una nota se debe tocar con el pulgar (p.ej. en música de violoncello).

```
\relative { <a'_\thumb a'-3>2 <b'_\thumb b'-3> }
```



Las digitaciones para los acordes también se pueden añadir a las notas individuales escribiéndolas después de las alturas.

```
\relative {
  <c''-1 e-2 g-3 b-5>2 <d-1 f-2 a-3 c-5>
}
```



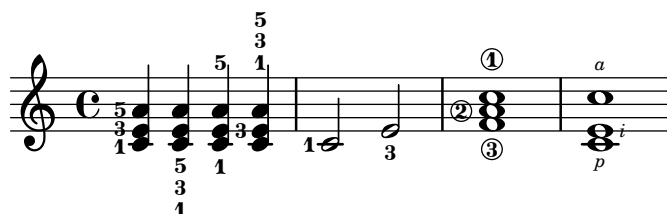
Las indicaciones de digitación se pueden situar manualmente encima o debajo del pentagrama, véase Sección 5.4.2 [Dirección y posición], página 661.

## Fragmentos de código seleccionados

### *Controlar la colocación de las digitaciones de acordes*

Se puede controlar con precisión la colocación de los números de digitación. Para que se tenga en cuenta la orientación de las digitaciones, se debe utilizar dentro de una construcción de acorde <> aunque sea una sola nota. Se puede establecer la orientación para los números de cuerda y las digitaciones de la mano derecha de una forma similar.

```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
  \set stringNumberOrientations = #'(up left down)
  <f\3 a\2 c\1>1
  \set strokeFingerOrientations = #'(down right up)
  <c\rightHandFinger #1 e\rightHandFinger #2 c'\rightHandFinger #4 >
}
```



### *Permitir que las digitaciones se impriman dentro del pentagrama*

Las cifras de digitación orientadas verticalmente se colocan de forma predeterminada fuera del pentagrama. Sin embargo, este comportamiento se puede cancelar. Hay que prestar atención en situaciones en las que las digitaciones y las plicas están en la misma dirección: por defecto, las digitaciones solo evitan la colisión con plicas unidas por una barra. Se puede cambiar este ajuste para no evitar ninguna plica o evitarlas todas; el ejemplo siguiente muestra las dos copciones, así como la manera de volver al comportamiento predeterminado.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering.staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 g'-0
  a8[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##f
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##t
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = #only-if-beamed
  a[-1 b]-2 g-0 r
}
```

}



## Véase también

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 661.

Fragmentos de código: Sección “Editorial annotations” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “FingeringEvent” en *Referencia de Funcionamiento Interno*, Sección “fingering-event” en *Referencia de Funcionamiento Interno*, Sección “Fingering-engraver” en *Referencia de Funcionamiento Interno*, Sección “New\_fingering-engraver” en *Referencia de Funcionamiento Interno*, Sección “Fingering” en *Referencia de Funcionamiento Interno*.

## Notas ocultas

Las notas ocultas (o invisibles, o transparentes) pueden ser de utilidad en la preparación de ejercicios de teoría o de composición.

```
\relative {
  c' '4 d
  \hideNotes
  e4 f
  \unHideNotes
  g a
  \hideNotes
  b
  \unHideNotes
  c
}
```



La cabeza, la plica y el corchete de las figuras, así como los silencios, son invisibles. Las barras son invisibles si comienzan en una nota oculta. Los objetos de notación que están anclados a notas invisibles, son a pesar de ello visibles.

```
\relative c' ' {
  e8(\p f g a)--
  \hideNotes
  e8(\p f g a)--
}
```



## Instrucciones predefinidas

`\hideNotes`, `\unHideNotes`.

## Véase también

Manual de aprendizaje: Sección “Visibilidad y color de los objetos” en *Manual de Aprendizaje*.

Notation Reference: [Silencios invisibles], página 64, Sección 5.4.7 [Visibilidad de los objetos], página 670, [Ocultar pentagramas], página 215.

Fragmentos de código: Sección “Editorial annotations” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Note-spacing-engraver” en *Referencia de Funcionamiento Interno*, Sección “NoteSpacing” en *Referencia de Funcionamiento Interno*.

## Colorear objetos

Se pueden asignar colores a los objetos individuales. Los nombres de color válidos se encuentran relacionados en la Sección A.7 [Lista de colores], página 710.

```
\override NoteHead.color = #red
c''4 c''
\override NoteHead.color = #(x11-color 'LimeGreen)
d''
\override Stem.color = "deepskyblue"
e''
```



Además de un conjunto limitado de colores sencillos que están disponibles como variables predefinidas (véase ‘Colores normales’ dentro de Sección A.7 [Lista de colores], página 710), se puede introducir cualquier color como una cadena de caracteres. Dicha cadena puede ser un nombre predefinido de color al estilo de CSS (<https://www.w3.org/Style/CSS/>), o un código de color hexadecimal con el carácter # como prefijo (*dentro* de las comillas):

```
\override NoteHead.color = "lightsalmon"
\override Flag.color = "#E30074"
\override Beam.color = "#5e45ad"
\override Rest.color = "#3058"
g'8 \huge r4 a'16 f'
```



Si dicho color incluye un canal alfa para la semitransparencia mediante el uso de un código de ocho caracteres del tipo "#RRGGBBAA" o su forma abreviada "#RGBA", se usará en la salida de SVG pero no en la salida de PostScript/PDF. En el ejemplo anterior, el silencio es semitransparente solo si el código se compila con el backend de SVG, como se explica en Sección 3.4.3 [Formatos de salida alternativos], página 550.

De manera completamente distinta, se puede acceder al rango de colores completo definido para X11 ([https://en.wikipedia.org/wiki/X11\\_color\\_names](https://en.wikipedia.org/wiki/X11_color_names)) mediante el uso de la función de Scheme `x11-color`. Esta función admite un argumento que puede ser un símbolo, como `'DarkSeaGreen4`, o una cadena de caracteres, como `"DarkSeaGreen4"`. La primera manera es más rápida de escribir y ligeramente más eficiente; sin embargo la segunda forma permite además especificar colores de X11 como frases de varias palabras: en el ejemplo, `"dark sea green 4"`.

Si `x11-color` no entiende el parámetro, el color predeterminado que se devuelve es el negro.

```
\new Staff \with {
```



```

instrumentName = \markup {
  \with-color #(x11-color 'red) "Clarinet"
}
}
\relative c'' {
\override Staff.StaffSymbol.color = #(x11-color 'SlateBlue2)
gis8 a
\override Beam.color = #(x11-color "medium turquoise")
gis a
\override Accidental.color = #(x11-color 'DarkRed)
gis a
\override NoteHead.color = #(x11-color "LimeGreen")
gis a
% this is deliberate nonsense; note that the stems remain black
\override Stem.color = #(x11-color 'Boggle)
b2 cis
}

```



Se pueden especificar colores RGB exactos utilizando la función de Scheme `rgb-color`. Esta función admite tres argumentos que se utilizan respectivamente para los canales *rojo*, *verde* y *azul*, y un número de *alfa* opcional para la semitransparencia (todos los valores deben ser números entre 0 y 1). De nuevo, la transparencia solo se contempla en la salida de SVG; por ejemplo, en el fragmento siguiente la clave es semitransparente cuando se produce un documento SVG.

```

\new Staff \with {
  instrumentName = \markup {
    \with-color #(x11-color 'red) "Clarinet"
  }
  \override Clef.color = #(rgb-color 0 0 0 0.5)
}
\relative c'' {
\override Staff.StaffSymbol.color = #(x11-color 'SlateBlue2)
\override Stem.color = #(rgb-color 0 0 0)
gis8 a
\override Stem.color = #(rgb-color 1 1 1)
gis8 a
\override Stem.color = #(rgb-color 0 0 0.5)
gis4 a
}

```



## Véase también

Referencia de la notación: Sección A.7 [Lista de colores], página 710, Sección 5.3.4 [La instrucción `\tweak`], página 650.

Fragmentos de código: Sección “Editorial annotations” en *Fragmentos de código*.

## Advertencias y problemas conocidos

Un color de X11 no es necesariamente de la misma tonalidad exacta que un color normal de nombre similar.

No todos los colores de X11 se distinguen entre sí en un navegador web, es decir, un navegador de web podría no mostrar ninguna diferencia entre **LimeGreen** (verde lima) y **ForestGreen** (verde bosque). Para la web se recomiendan los colores de CSS, como aparece detallado en Sección A.7 [Lista de colores], página 710.

Las notas de un acorde no se pueden colorear individualmente con `\override`; en su lugar utilice `\tweak` o su equivalente `\single\override` antes de la nota respectiva. Consulte Sección 5.3.4 [La instrucción `\tweak`], página 650, para ver más detalles.

## Paréntesis

Los objetos se pueden encerrar entre paréntesis anteponiendo `\parenthesize` al evento musical. Si se aplica a un acorde, encierra cada nota dentro de un par de paréntesis. También se pueden poner entre paréntesis las notas individuales de un acorde.

```
\relative {
  c'2 \parenthesize d
  c2 \parenthesize <c e g>
  c2 <c \parenthesize e g>
}
```



Los objetos que no son notas también se pueden poner entre paréntesis. Para las articulaciones se necesita un guión antes de la instrucción `\parenthesize`.

```
\relative {
  c'2-\parenthesize -. d
  c2 \parenthesize r
}
```



## Véase también

Fragmentos de código: Sección “Editorial annotations” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Parenthesis\_engraver” en *Referencia de Funcionamiento Interno*, Sección “ParenthesesItem” en *Referencia de Funcionamiento Interno*, Sección “parentheses-interface” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Al poner un acorde entre paréntesis, se encierra cada una de las notas individuales entre paréntesis, en vez de un solo paréntesis grande rodeando al acorde completo.

## Plicas

Cuando se encuentra con una nota, se crea automáticamente un objeto **Stem** (plica). Para las redondas y los silencios, también se crean pero se hacen invisibles.

Se puede hacer manualmente que las plicas apunten hacia arriba o hacia abajo; véase Sección 5.4.2 [Dirección y posición], página 661.

## Instrucciones predefinidas

`\stemUp`, `\stemDown`, `\stemNeutral`.

## Fragmentos de código seleccionados

### *Dirección predeterminada de las plicas sobre la tercera línea del pentagrama*

La dirección predeterminada de las plicas sobre la tercera línea del pentagrama está determinada por la propiedad `neutral-direction` del objeto `Stem`.

```
\relative c'' {
  a4 b c b
  \override Stem.neutral-direction = #up
  a4 b c b
  \override Stem.neutral-direction = #down
  a4 b c b
}
```



### *Cambiar la dirección de la plica de las notas de la tercera línea automáticamente, basado en la melodía*

LilyPond puede alterar la dirección de la plica de las notas que van en la tercera línea de un pentagrama de forma que siga la melodía, mediante la adición del grabador `Melody_engraver` al contexto `Voice`.

```
\relative c'' {
  \time 3/4
  a8 b g f b g |
  \set suspendMelodyDecisions = ##t
  a b g f b g |
  \unset suspendMelodyDecisions
  c b d c b c |
}

\layout {
  \context {
    \Voice
    \consists "Melody_engraver"
    \autoBeamOff
  }
}
```



## Véase también

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 661.

Fragmentos de código: Sección “Editorial annotations” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Stem\_engraver” en *Referencia de Funcionamiento Interno*, Sección “Stem” en *Referencia de Funcionamiento Interno*, Sección “stem-interface” en *Referencia de Funcionamiento Interno*.

### 1.7.2 Fuera del pentagrama

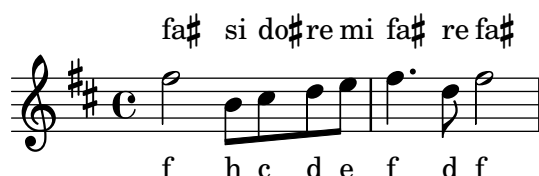
Esta sección trata sobre cómo enfatizar elementos que están dentro del pentagrama, desde fuera del pentagrama.

#### Nombre de las notas

Se pueden imprimir los nombres de las notas como texto, usando el contexto `NoteNames`. Si se usa simultáneamente con una pauta ordinaria, es posible sincronizar cada nota con su nombre impreso por encima o por debajo de la pauta.

```
\language "italiano"
melody = \relative do'' {
  fad2 si,8 dod re mi fad4. re8 fad2
}

<<
  \new NoteNames { \melody }
  \new Staff { \key si \minor \melody }
  \new NoteNames {
    \set printNotesLanguage = "deutsch"
    \set printAccidentalNames = ##f
    \melody
  }
>>
```



De manera predeterminada, los nombres de nota se imprimen en el mismo idioma que se ha usado para la entrada de la música; sin embargo, la propiedad `printNotesLanguage` permite seleccionar cualquier otro idioma disponible (véase [Nombres de las notas en otros idiomas], página 8). A través de la propiedad `printAccidentalNames` se determina si las alteraciones se imprimen o no.

Estableciendo tanto la mencionada propiedad a un símbolo, como la propiedad `printOctaveNames` al valor `##t` (verdadero), se pueden obtener nombres de notas de una manera muy similar a la sintaxis de entrada de LilyPond. Para un resultado más general, es posible conseguir nombres de nota ‘científicos’.

```
melody = \relative c'' {
  fis2 b,8 cis d e fis4. d8 fis2
}

<<
  \new NoteNames {
    \set printOctaveNames = ##t
    \set printAccidentalNames = #'lily
    \melody
  }
```

```

}
\new Staff { \key b \minor \melody }
\new NoteNames {
  \set printOctaveNames = #'scientific
  \melody
}
>>

```



La propiedad `noteNameSeparator` define cómo se imprimen los acordes. Se pueden definir otras funciones de formateo como `noteNameFunction`; tales funciones deben admitir los argumentos `altura` y `contexto`, incluso si alguno de ellos puede ignorarse.

```

somechords = \relative c' {
  <b d fis>2 <b cis e g> <b d fis> q
}

<<
\new NoteNames {
  \set noteNameSeparator = "+"
  \somechords
}
\new Staff { \key b \minor \somechords }
\new NoteNames {
  \set noteNameFunction =
    #(lambda (pitch ctx)
      (alteration->text-accidental-markup
        (ly:pitch-alteration pitch)))
  \somechords
}
>>

```



## Véase también

Referencia de la notación: [Nombres de las notas en otros idiomas], página 8.

Referencia de funcionamiento interno: Sección “NoteName” en *Referencia de Funcionamiento Interno*, Sección “NoteNames” en *Referencia de Funcionamiento Interno*, Sección “Note\_name\_engraver” en *Referencia de Funcionamiento Interno*.

## Globos de ayuda

Los elementos de notación se pueden marcar y nombrar con la ayuda de un cartel o globo de ayuda rectangular. El propósito principal de esta funcionalidad es la explicación de la notación.

```

\new Voice \with { \consists "Balloon_engraver" }

```

```

\relative c'' {
  \balloonGrobText #'Stem #'(3 . 4) \markup { "Soy una plica" }
  a8
  \balloonGrobText #'Rest #'(-4 . -4) \markup { "Soy un silencio" }
  r
  <c, g'-\balloonText #'(-2 . -2) \markup { "Soy una cabeza" } c>2.
}

```



Existen dos funciones musicales, `balloonGrobText` y `balloonText`; el primero se usa como `\once\override` para adjuntar un texto a cualquier grob, y el último se usa como `\tweak`, normalmente dentro de acordes, para adjuntar un texto a una nota individual.

Los textos de globo no influyen en el espaciado de las notas, pero esto puede cambiarse:

```

\new Voice \with { \consists "Balloon_engraver" }
\relative c'' {
  \balloonGrobText #'Stem #'(3 . 4) \markup { "Soy una plica" }
  a8
  \balloonGrobText #'Rest #'(-4 . -4) \markup { "Soy un silencio" }
  r
  \balloonLengthOn
  <c, g'-\balloonText #'(-2 . -2) \markup { "Soy una cabeza" } c>2.
}

```



## Instrucciones predefinidas

`\balloonLengthOn`, `\balloonLengthOff`.

## Véase también

Fragmentos de código: Sección “Editorial annotations” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Balloon\_engraver” en *Referencia de Funcionamiento Interno*, Sección “BalloonTextItem” en *Referencia de Funcionamiento Interno*, Sección “balloon-interface” en *Referencia de Funcionamiento Interno*.

## Líneas de rejilla

Se pueden dibujar líneas verticales entre los pentagramas sincronizadas con las notas.

Se debe usar el grabador `Grid_point_engraver` para crear los puntos extremos de las líneas, mientras que el grabador `Grid_line_span_engraver` se debe utilizar para trazar efectivamente las líneas. De forma predeterminada, esto centra las líneas de rejilla horizontalmente debajo y al

lado izquierdo de la cabeza de las notas. Las líneas de rejilla se extienden a partir de línea media de los pentagramas. El intervalo `gridInterval` debe especificar la duración entre las líneas de rejilla.

```
\layout {
  \context {
    \Staff
    \consists "Grid_point_engraver"
    gridInterval = #(ly:make-moment 1/4)
  }
  \context {
    \Score
    \consists "Grid_line_span_engraver"
  }
}

\score {
  \new ChoirStaff <<
    \new Staff \relative {
      \stemUp
      c''4. d8 e8 f g4
    }
    \new Staff \relative {
      \clef bass
      \stemDown
      c4 g' f e
    }
  >>
}
```



## Fragmentos de código seleccionados

### *Líneas de rejilla: modificar su aspecto*

Se puede cambiar el aspecto de las líneas de rejilla sobreescribiendo algunas de sus propiedades.

```
\score {
  \new ChoirStaff <<
    \new Staff {
      \relative c'' {
        \stemUp
        c'4. d8 e8 f g4
      }
    }
    \new Staff {
      \relative c {
        % this moves them up one staff space from the default position

```

```

\override Score.GridLine.extra-offset = #'(0.0 . 1.0)
\stemDown
\clef bass
\once \override Score.GridLine.thickness = #5.0
c4
\once \override Score.GridLine.thickness = #1.0
g'4
\once \override Score.GridLine.thickness = #3.0
f4
\once \override Score.GridLine.thickness = #5.0
e4
}
}
>>
\layout {
  \context {
    \Staff
    % set up grids
    \consists "Grid_point_engraver"
    % set the grid interval to one quarter note
    gridInterval = #(ly:make-moment 1/4)
  }
  \context {
    \Score
    \consists "Grid_line_span_engraver"
    % this moves them to the right half a staff space
    \override NoteColumn.X-offset = #-0.5
  }
}
}

```



## Véase también

Fragmentos de código: Sección “Editorial annotations” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Grid\_line\_span\_engraver” en *Referencia de Funcionamiento Interno*, Sección “Grid\_point\_engraver” en *Referencia de Funcionamiento Interno*, Sección “GridLine” en *Referencia de Funcionamiento Interno*, Sección “GridPoint” en *Referencia de Funcionamiento Interno*, Sección “grid-line-interface” en *Referencia de Funcionamiento Interno*, Sección “grid-point-interface” en *Referencia de Funcionamiento Interno*.

## Corchetes de análisis

Los corchetes se usan en análisis musical para indicar la estructura de las piezas musicales. Están contemplados los corchetes horizontales simples.



```

\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}
\relative {
  c' '2\startGroup
  d\stopGroup
}

```



Los corchetes de análisis se pueden anidar.

```

\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}
\relative {
  c' '4\startGroup\startGroup
  d4\stopGroup
  e4\startGroup
  d4\stopGroup\stopGroup
}

```



## Fragmentos de código seleccionados

### *Corchetes de análisis encima del pentagrama*

De forma predeterminada se añaden corchetes de análisis sencillos debajo del pentagrama. El ejemplo siguiente muestra una manera de colocarlos por encima.

```

\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}

\relative c'' {
  \once \override HorizontalBracket.direction = #UP
  c2\startGroup
  d2\stopGroup
}

```

}

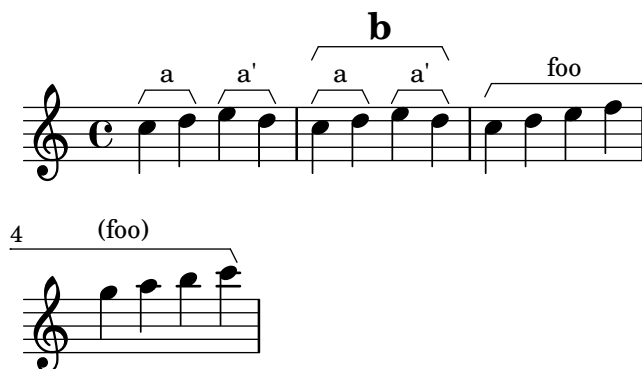


### *Analysis brackets with labels*

Text markup may be added to analysis brackets through the `text` property of the `HorizontalBracketText` grob. Adding different texts to brackets beginning at the same time requires the `\tweak` command. Bracket text will be parenthesized after a line break.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
    \override HorizontalBracket.direction = #UP
  }
}

{
  \once\override HorizontalBracketText.text = "a"
  c''\startGroup d''\stopGroup
  \once\override HorizontalBracketText.text = "a'"
  e''\startGroup d''\stopGroup |
  c''-\tweak HorizontalBracketText.text
    \markup \bold \huge "b" \startGroup
    -\tweak HorizontalBracketText.text "a" \startGroup
    d''\stopGroup
    e''-\tweak HorizontalBracketText.text "a'" \startGroup
    d''\stopGroup\stopGroup |
  c''-\tweak HorizontalBracketText.text foo \startGroup
    d'' e'' f'' | \break
    g'' a'' b'' c'''\stopGroup
}
```



### **Véase también**

Referencia de funcionamiento interno: Sección “Horizontal\_bracket\_engraver” en *Referencia de Funcionamiento Interno*, Sección “HorizontalBracket” en *Referencia de Funcionamiento Interno*, Sección “horizontal-bracket-interface” en *Referencia de Funcionamiento Interno*, Sección “HorizontalBracketText” en *Referencia de Funcionamiento Interno*, Sección “horizontal-bracket-text-interface” en *Referencia de Funcionamiento Interno*, Sección “Staff” en *Referencia de Funcionamiento Interno*.

## 1.8 Texto

**Moderato cantabile molto espressivo**

*p con amabilità (sanft)*

4

*p*

6

Esta sección explica cómo incluir texto (con diversos estilos de formateo) en las partituras.

### 1.8.1 Escritura del texto

Esta sección presenta las distintas formas de añadir texto a una partitura.

**Nota:** Para escribir texto con caracteres acentuados y especiales (como los de idiomas distintos del inglés), sencillamente inserte los caracteres directamente en el archivo de LilyPond. El archivo se debe guardar como UTF-8. Para ver más información, consulte [Codificación del texto], página 546.

### Panorámica de los objetos de texto

Los objetos de texto se introducen ya sea como simples cadenas entrecomilladas, o como bloques de `\markup` que pueden aceptar una variedad de posibilidades avanzadas de formato y gráficas, como se detalla en la sección Sección 1.8.2 [Formatear el texto], página 264.

Así, los bloques de marcado se pueden utilizar:

- dentro de cualquier objeto del tipo `TextScript` (aplicado a una nota con `-`, `^` o `_`); véase [Guiones de texto], página 257;
- como ‘spanners’ u objetos extensos, cuando ciertas indicaciones se prolongan sobre varios pulsos o compases. Véase [Extensiones de texto], página 258;

- dentro de cualquier marca impresa por encima de los pentagramas, tales como obletos `RehearsalMark` (letras de ensayo) o `MetronomeMark` (indicaciones de tempo) introducidos respectivamente con las palabras clave `\mark` y `\tempo`; véase [Indicaciones de texto], página 260;
- como bloques de marcado autosuficientes, que se introducen en el nivel jerárquico superior fuera de cualquier bloque `\score` (en este caso es obligatoria la instrucción `\markup {...}`, y no se puede omitir en favor de una simple cadena de texto entrecomillada); véase [Texto separado], página 262;
- en cualquier definición dentro del bloque `\header` (como `title`, `subtitle`, `composer`), o en elementos específicos definidos dentro del bloque `\paper` tales como `evenHeaderMarkup` para los números de página. Esto se explica en la sección Sección 3.2 [Títulos y encabezamientos], página 512.

Muchos otros objetos basados en texto se pueden escribir como bloques de marcado, incluso cuando este no sea su uso principal.

- Las digitaciones se pueden sustituir fácilmente con bloques de marcado, si se escriben con la instrucción `\finger`; véase [Indicaciones de digitación], página 240.
- Las sílabas de la letra de una canción se pueden formatear por medio de la instrucción `\markup`; véase Sección 2.1.1 [Notación común para música vocal], página 286.
- Los nombres de acordes están, de hecho, definidos como bloques de marcado, y por tanto se pueden redefinir de la misma forma para personalizar los modificadores o las excepciones de acorde; véase Sección 2.7.2 [Imprimir los acordes], página 448.
- Los matices dinámicos se escriben habitualmente de una forma sencilla; sin embargo, es posible definir [Indicaciones dinámicas nuevas], página 137, como objetos de marcado. Ciertas dinámicas como *crescendo* se imprimen como objetos extensos y se pueden redefinir a través de propiedades como `crescendoText`; véase [Matices dinámicos], página 131.
- Otros objetos menos frecuentes también están hechos con bloques de marcado, como las indicaciones del tipo [Globos de ayuda], página 248.

De hecho es posible usar la instrucción `\markup` para personalizar la apariencia de prácticamente cualquier objeto gráfico (o ‘grob’), bien sea sobrescribiendo su propiedad `text`, si la tiene, o bien su propiedad `stencil`. Una parte de la lógica que hace posible todo esto, está explicada en Sección “Arquitectura flexible” en *Ensayo*.

El ejemplo que aparece a continuación ilustra la omnipresencia de los bloques de marcado, no solo como algunos de los objetos relacionados más arriba, sino también en sustitución de objetos musicales por objetos de texto a través de distintos métodos.

```
\header { title = \markup "Header" }

dyn =
#(make-dynamic-script #{ \markup \text "DynamicText" #})

\markup \box "Top-level markup"

\score {
  <<
    \new ChordNames
    \with {
      majorSevenSymbol = \markup "majorSevenSymbol"
    }
    \chordmode { c1:maj7 }
    \new Staff {
```

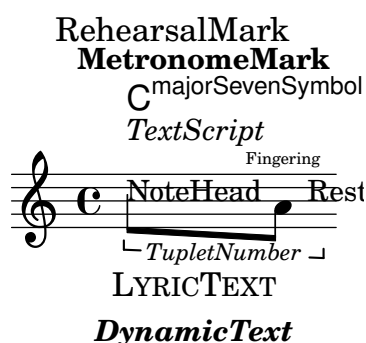
```

\tempo \markup "MetronomeMark"
\mark \markup \smaller "RehearsalMark"
\once \override TupletNumber.text =
  \markup "TupletNumber"
\tuplet 3/2 {
  \once \override NoteHead.stencil =
    #ly:text-interface::print
  \once \override NoteHead.text =
    \markup \lower #0.5 "NoteHead"
  c''8^\markup \italic "TextScript"
  a'\finger \markup \text "Fingering"
  \once \override Rest.stencil =
    #(lambda (grob)
      (grob-interpret-markup grob #{
        \markup "Rest"
      #}))
    r
}
}
\new Lyrics \lyricmode {
  \markup \smallCaps "LyricText" 1
}
\new Dynamics { s1\dyn }
>>
}

```

## Header

Top-level markup



### Véase también

Referencia de la notación: Sección 1.8.2 [Formatear el texto], página 264, [Guiones de texto], página 257, [Extensiones de texto], página 258, [Indicaciones de texto], página 260, [Texto separado], página 262, [Indicaciones de digitación], página 240, Sección 2.1.1 [Notación común para música vocal], página 286, Sección 2.7.2 [Imprimir los acordes], página 448, [Indicaciones dinámicas nuevas], página 137, [Matices dinámicos], página 131, [Globos de ayuda], página 248.

Ensayo sobre grabado musical automatizado: Sección “Arquitectura flexible” en *Ensayo*.

Fragmentos de código: Sección “Text” en *Fragmentos de código*.

## Guiones de texto

Es posible añadir indicaciones de “texto entre comillas” a una partitura, como se muestra en el ejemplo siguiente. Estas indicaciones se pueden colocar manualmente por encima o por debajo del pentagrama, utilizando la sintaxis que se describe en Sección 5.4.2 [Dirección y posición], página 661.

```
\relative { a'8^"pizz." g f e a4-"scherz." f }
```



Esta sintaxis es en realidad una abreviatura; se puede añadir explícitamente a una nota un formateado de texto más complejo utilizando un bloque `\markup`, como se describe bajo Sección 1.8.2 [Formatear el texto], página 264.

```
\relative {
  a'8^\markup { \italic pizz. } g f e
  a4_\markup { \tiny scherz. \bold molto } f }
```



De forma predeterminada, las indicaciones de texto no afectan al espaciado de las notas. Sin embargo, su anchura sí podría tenerse en cuenta: en el siguiente ejemplo, la primera cadena de texto no afecta al espaciado, pero la segunda sí afecta.

```
\relative {
  a'8^"pizz." g f e
  \textLength0n
  a4_"scherzando" f
}
```



Se pueden adjuntar articulaciones a las notas, además de inscripciones de texto. Para ver más información, consulte [Articulaciones y ornamentos], página 128.

Para ver más información sobre el orden relativo de las inscripciones de texto y las articulaciones, consulte Sección “Colocación de los objetos” en *Manual de Aprendizaje*.

## Instrucciones predefinidas

`\textLength0n`, `\textLength0ff`.

## Véase también

Manual de aprendizaje: Sección “Colocación de los objetos” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 1.8.2 [Formatear el texto], página 264, Sección 5.4.2 [Dirección y posición], página 661, [Articulaciones y ornamentos], página 128.

Fragmentos de código: Sección “Text” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

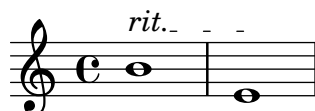
La comprobación necesaria para asegurar que las inscripciones de texto y la letra de las canciones se mantienen dentro de los márgenes, requiere cálculos adicionales. En caso de que desee un proceso ligeramente más rápido, puede utilizar

```
\override Score.PaperColumn.keep-inside-line = ##f
```

## Extensiones de texto

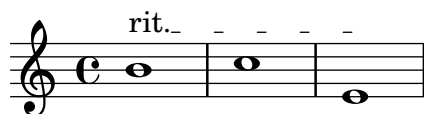
Ciertas indicaciones de ejecución, p.ej., *rallentando* o *accelerando*, se escriben como texto y se extienden sobre muchos compases mediante líneas de puntos. Estos objetos, que reciben el nombre de ‘spanners’ u objetos de extensión, se pueden crear desde una nota hasta otra usando la siguiente sintaxis:

```
\relative {
  \override TextSpanner.bound-details.left.text = "rit."
  b'1\startTextSpan
  e,\stopTextSpan
}
```



La cadena de texto que imprimir se establece a través de propiedades de objeto. De forma predeterminada se imprime en estilo cursiva, pero se pueden conseguir distintos efectos de formato utilizando bloques `\markup`, como se describe en Sección 1.8.2 [Formatear el texto], página 264.

```
\relative {
  \override TextSpanner.bound-details.left.text =
    \markup { \upright "rit." }
  b'1\startTextSpan c
  e,\stopTextSpan
}
```



El estilo de la línea, así como la cadena de texto, se pueden definir como una propiedad de objeto. Esta sintaxis se describe en Sección 5.4.8 [Estilos de línea], página 676.

## Instrucciones predefinidas

```
\textSpannerUp, \textSpannerDown, \textSpannerNeutral.
```

## Advertencias y problemas conocidos

LilyPond sólo puede manejar un objeto de extensión de texto por cada voz.

## Fragmentos de código seleccionados

### *Objetos extensores de texto postfijos para dinámica*

Los objetos de extensión `\cresc`, `\dim` y `\decrec` ahora se pueden redefinir como operadores postfijos y producir un solo objeto de extensión de texto. La definición de extensores personalizados también es fácil. Se pueden mezclar con facilidad los crescendi textuales y en forma

de reguladores. `\<` y `\>` producen reguladores gráficos de forma predeterminada, `\cresc` etc. producen elementos extensores de texto de forma predeterminada.

```
% Some sample text dynamic spanners, to be used as postfix operators
crpoco =
#(make-music 'CrescendoEvent
  'span-direction START
  'span-type 'text
  'span-text "cresc. poco a poco")

\relative c' {
  c4\cresc d4 e4 f4 |
  g4 a4\! b4\crpoco c4 |
  c4 d4 e4 f4 |
  g4 a4\! b4\< c4 |
  g4\dim a4 b4\decreasc c4\!
}
```



### *Objeto personalizado de extensión de texto de matices dinámicos, postfixo*

Funciones postfixas para la creación de objetos de extensión de texto personalizados. Los objetos de extensión deben comenzar en la primera nota del compás. Hay que utilizar `-\mycresc`, en caso contrario el comienzo del objeto de extensión se asignará a la nota siguiente.

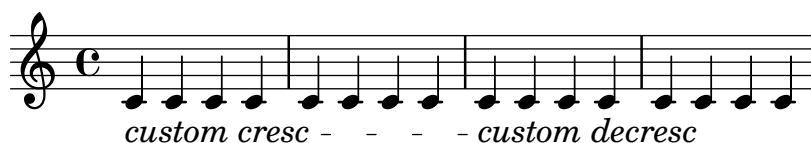
```
% Two functions for (de)crescendo spanners where you can explicitly
% give the spanner text.
mycresc =
#(define-music-function (mymarkup) (markup?)
  (make-music 'CrescendoEvent
    'span-direction START
    'span-type 'text
    'span-text mymarkup))

mydecreasc =
#(define-music-function (mymarkup) (markup?)
  (make-music 'DecrescendoEvent
    'span-direction START
    'span-type 'text
    'span-text mymarkup))

\relative c' {
  c4-\mycresc "custom cresc" c4 c4 c4 |
  c4 c4 c4 c4 |
  c4-\mydecreasc "custom decresc" c4 c4 c4 |
  c4 c4\! c4 c4
```



}



## Véase también

Referencia de la notación: Sección 5.4.8 [Estilos de línea], página 676, [Matices dinámicos], página 131, Sección 1.8.2 [Formatear el texto], página 264.

Fragmentos de código: Sección “Text” en *Fragmentos de código*, Sección “Expressive marks” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextSpanner” en *Referencia de Funcionamiento Interno*.

## Indicaciones de texto

Se pueden añadir a la partitura varios elementos de texto con la sintaxis que se describe en [Llamadas de ensayo], página 118:

```
\relative {
  \mark "Verse"
  c'2 g'
  \bar "||"
  \mark "Chorus"
  g2 c,
  \bar "|."
}
```



Esta sintaxis posibilita colocar cualquier texto sobre la línea divisoria; se pueden incorporar formatos más complejos para el texto usando un bloque `\markup`, como está explicado en Sección 1.8.2 [Formatear el texto], página 264:

```
\relative {
  <c' e>1
  \mark \markup { \italic { colla parte } }
  <d f>2 <e g>
  <c f aes>1
}
```



Esta sintaxis también permite imprimir símbolos especiales como llamadas, segno o calderones, especificando el nombre del símbolo correspondiente como se explica en [Notación musical dentro de elementos de marcado], página 276:

```
\relative {
  <bes' f>2 <aes d>
```

```
\mark \markup { \musicglyph "scripts.ufermata" }
<e g>1
}
```



Estos objetos se tipografían solamente sobre el pentagrama superior de la partitura; dependiendo de si están especificados al final o en medio de un compás, se colocarán sobre la línea divisoria o entre las notas. Si se especifican en un salto de línea, las llamadas se imprimen al principio de la línea siguiente.

```
\relative c'' {
  \mark "Allegro"
  c1 c
  \mark "assai" \break
  c c
}
```



## Instrucciones predefinidas

`\markLengthOn`, `\markLengthOff`.

## Fragmentos de código seleccionados

### *Imprimir marcas al final de una línea*

Se pueden imprimir marcas al final de la línea actual, en vez de al principio de la línea siguiente. En estos casos, puede ser preferible alinear el borde derecho de la marca con la línea divisoria.

```
\relative c'' {
  g2 c
  d,2 a'
  \once \override Score.RehearsalMark.break-visibility =
    #end-of-line-visible
  \once \override Score.RehearsalMark.self-alignment-X =
    #RIGHT
  \mark "D.C. al Fine"
  \break
  g2 b,
  c1 \bar "||"
}
```

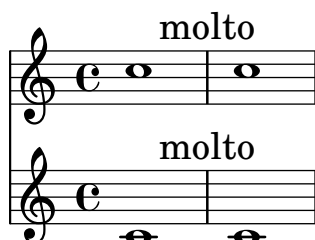




### *Imprimir marcas de ensayo en cualquier pentagrama*

Aunque normalmente las marcas de ensayo textuales sólo se imprimen sobre el pentagrama superior, también se pueden imprimir en otro pentagrama cualquiera.

```
\score {
  <<
    \new Staff { c''1 \mark "molto" c'' }
    \new Staff { c'1 \mark "molto" c' }
  >>
  \layout {
    \context {
      \Score
      \remove "Mark_engraver"
      \remove "Staff_collecting_engraver"
    }
    \context {
      \Staff
      \consists "Mark_engraver"
      \consists "Staff_collecting_engraver"
    }
  }
}
```



### Véase también

Referencia de la notación: [Llamadas de ensayo], página 118, Sección 1.8.2 [Formatear el texto], página 264, [Notación musical dentro de elementos de marcado], página 276, Sección A.8 [La tipografía Emmentaler], página 713.

Fragmentos de código: Sección “Text” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “MarkEvent” en *Referencia de Funcionamiento Interno*, Sección “Mark\_engraver” en *Referencia de Funcionamiento Interno*, Sección “RehearsalMark” en *Referencia de Funcionamiento Interno*.

### Texto separado

Un bloque `\markup` puede existir de forma independiente, fuera de cualquier bloque `\score`, como una “expresión de nivel superior”. Esta sintaxis se describe en Sección 3.1.5 [Estructura del archivo], página 510.

```
\markup {
  Tomorrow, and tomorrow, and tomorrow...
}
```

Tomorrow, and tomorrow, and tomorrow...

Esto hace posible imprimir texto separado de la música, lo que es especialmente útil cuando el archivo de entrada contiene varias piezas, tal y como se describe en Sección 3.1.2 [Varias partituras en un libro], página 507.

```
\score {
  c'1
}
\markup {
  Tomorrow, and tomorrow, and tomorrow...
}
\score {
  c'1
}
```



Tomorrow, and tomorrow, and tomorrow...



Los bloques de texto independientes pueden abarcar varias páginas, posibilitando la impresión de documentos de texto o libros íntegramente desde LilyPond. Esta funcionalidad y la sintaxis que requiere se describen en [Elementos de marcado de varias páginas], página 279.

## Instrucciones predefinidas

`\markup`, `\markuplist`.

## Fragmentos de código seleccionados

### *Elemento de marcado de texto independiente en dos columnas*

Los textos independientes se pueden disponer en varias columnas utilizando instrucciones `\markup`:

```
\markup {
  \fill-line {
    \hspace #1
    \column {
      \line { 0 sacrum convivium }
      \line { in quo Christus sumitur, }
      \line { recolitur memoria passionis ejus, }
      \line { mens impletur gratia, }
      \line { futurae gloriae nobis pignus datur. }
      \line { Amen. }
    }
  }
  \hspace #2
  \column \italic {
    \line { 0 sacred feast }
    \line { in which Christ is received, }
    \line { the memory of His Passion is renewed, }
```

```

\line { the mind is filled with grace, }
\line { and a pledge of future glory is given to us. }
\line { Amen. }
}
\hspace #1
}
}

```

<b>O sacrum convivium</b>	<i>O sacred feast</i>
<b>in quo Christus sumitur,</b>	<i>in which Christ is received,</i>
<b>recolitur memoria passionis ejus,</b>	<i>the memory of His Passion is renewed,</i>
<b>mens impletur gratia,</b>	<i>the mind is filled with grace,</i>
<b>futurae gloriae nobis pignus datur.</b>	<i>and a pledge of future glory is given to us.</i>
<b>Amen.</b>	<i>Amen.</i>

## Véase también

Referencia de la notación: Sección 1.8.2 [Formatear el texto], página 264, Sección 3.1.5 [Estructura del archivo], página 510, Sección 3.1.2 [Varias partituras en un libro], página 507, [Elementos de marcado de varias páginas], página 279.

Fragmentos de código: Sección “Text” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” en *Referencia de Funcionamiento Interno*.

## 1.8.2 Formatear el texto

Esta sección presenta los formateados básico y avanzado de texto, usando la sintaxis específica del modo de `\markup`.

### Introducción al marcado de texto

Se usa un bloque `\markup` para tipografiar texto con una sintaxis ampliable que se denomina “modo de marcado”. Tales bloques se pueden emplear en cualquier parte, como se explica en [Panorámica de los objetos de texto], página 254.

La sintaxis del marcado es similar a la sintaxis usual de LilyPond: una expresión `\markup` se encierra entre llaves `{ ... }`. Una sola palabra se considera como una expresión mínima, y por tanto no necesita estar encerrada entre llaves.

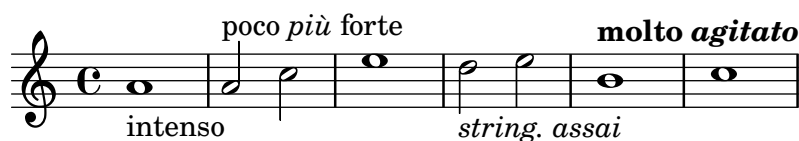
A diferencia de las indicaciones de “texto entrecomillado” simples, los bloques `\markup` pueden contener expresiones anidadas o instrucciones de marcado, que se introducen utilizando el carácter de barra invertida `\`. Estas instrucciones sólo afectan a la expresión que sigue inmediatamente.

```

\relative {
  a'1-\markup intenso
  a2^\markup { poco \italic più forte }
  c e1
  d2_\markup { \italic "string. assai" }
  e
  b1^\markup { \bold { molto \italic agitato } }
  c
}

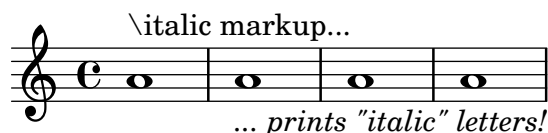
```

}



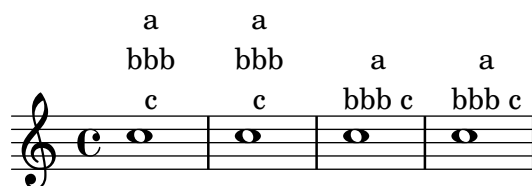
Un bloque `\markup` puede contener también cadenas de texto entre comillas. Dichas cadenas se tratan como expresiones de texto mínimas, y por tanto cualquier instrucción de marcado o carácter especial (como `\` y `#`) se imprimen literalmente sin afectar al formateo del texto. Se puede hacer que se impriman las propias comillas si se les antepone una barra invertida.

```
\relative {
  a'1^\italic markup..."
  a_\markup { \italic "... prints \"italic\" letters!" }
  a a
}
```



Para que se traten como una expresión distinta, es necesario que las listas de palabras estén encerradas entre comillas o precedidas de una instrucción. La forma en que están definidas las expresiones de marcado afecta a cómo se apilan, se centran y se alinean estas expresiones; en el ejemplo siguiente, la segunda expresión `\markup` se trata igual que la primera:

```
\relative c'' {
  c1^\markup { \center-column { a bbb c } }
  c1^\markup { \center-column { a { bbb c } } }
  c1^\markup { \center-column { a \line { bbb c } } }
  c1^\markup { \center-column { a "bbb c" } }
}
```



Los marcados se pueden almacenar dentro de variables. Estas variables se pueden adjuntar directamente a las notas:

```
allegro = \markup { \bold \large Allegro }

{
  d'8.^allegro
  d'16 d'4 r2
}
```



Se puede encontrar una lista exhaustiva de las instrucciones específicas de `\markup` en Sección A.11 [Instrucciones de marcado de texto], página 739. The inner workings of these commands, and how to implement new ones, is explained in Sección “Markup functions” en *Extender*.

## Véase también

Referencia de la notación: [Panorámica de los objetos de texto], página 254, Sección A.11 [Instrucciones de marcado de texto], página 739.

Manual de extensión: Sección “Funciones de marcado” en *Extender*.

Fragmentos de código: Sección “Text” en *Fragmentos de código*.

Archivos de inicio: `scm/markup.scm`.

## Advertencias y problemas conocidos

Los mensajes de error de sintaxis para el modo de marcado a menudo producen confusión.

## Seleccionar la tipografía y su tamaño

Está contemplado de forma básica el cambio de la fuente tipográfica en el modo de marcado:

```
\relative {
  d''1^\markup {
    \bold { Più mosso }
    \italic { non troppo \underline Vivo }
  }
  r2 r4 r8
  d, _\markup { \italic quasi \smallCaps Tromba }
  f1 d2 r
}
```



El tamaño de la fuente tipográfica se puede alterar en relación al tamaño global del pentagrama, de una serie de formas como se ve a continuación.

Se puede fijar a un tamaño predefinido:

```
\relative b' {
  b1_\markup { \huge Sinfonia }
  b1^\markup { \teeny da }
  b1-\markup { \normalsize camera }
}
```



Se puede establecer a un tamaño relativo al valor anterior:

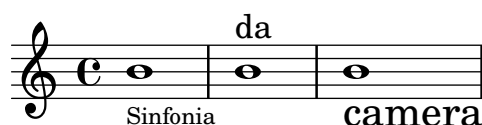
```
\relative b' {
  b1_\markup { \larger Sinfonia }
  b1^\markup { \smaller da }
  b1-\markup { \magnify #0.6 camera }
```

}



Se puede aumentar o disminuir de forma relativa al valor fijado por el tamaño global del pentagrama:

```
\relative b' {
  b1_\markup { \fontsize #-2 Sinfonia }
  b1^\markup { \fontsize #1 da }
  b1-\markup { \fontsize #3 camera }
}
```



También se puede establecer a un tamaño de puntos fijo, independientemente del tamaño global del pentagrama:

```
\relative b' {
  b1_\markup { \abs-fontsize #20 Sinfonia }
  b1^\markup { \abs-fontsize #8 da }
  b1-\markup { \abs-fontsize #14 camera }
}
```



Si el texto incluye espacios, es mejor escribir todo entre comillas, de forma que el tamaño de cada espacio sea correcto para el tamaño de los otros caracteres.

```
\markup \fontsize #6 \bold { Sinfonia da camera }
\markup \fontsize #6 \bold { "Sinfonia da camera" }
```

## Sinfonia da camera

## Sinfonia da camera

El texto se puede imprimir como subíndice o como superíndice. De forma predeterminada se imprimen en un tamaño menor, pero también se puede usar un tamaño normal:

```
\markup {
  \column {
    \line { 1 \super st movement }
    \line { 1 \normal-size-super st movement }
    \sub { (part two) } }
}
```

1<sup>st</sup> movement

1<sup>st</sup> movement<sub>(part two)</sub>



El modo de marcado ofrece una manera fácil de elegir familias de tipografía alternativas. A no ser que se especifique de otro modo, se selecciona automáticamente la fuente predeterminada con serifa, de tipo romano: en la última línea del ejemplo siguiente, no hay diferencia entre la primera y la segunda palabra.

```
\markup {
  \column {
    \line { Act \number 1 }
    \line { \sans { Scene I. } }
    \line { \typewriter { Verona. An open place. } }
    \line { Enter \roman Valentine and Proteus. }
  }
}
```

**Act 1**  
**Scene I.**  
 Verona. An open place.  
**Enter Valentine and Proteus.**

Algunas de estas familias, usadas para elementos específicos como números o matices, no ofrecen todos los caracteres, como se explica en [Indicaciones dinámicas nuevas], página 137, y [Marcas de repetición manual], página 166.

Si se usan dentro de una palabra, algunas instrucciones de cambio de tipografía o de formateo pueden producir un espacio vacío no deseado. Esto se puede solucionar concatenando en uno solo los distintos elementos de texto:

```
\markup {
  \column {
    \line {
      \concat { 1 \super st }
      movement
    }
    \line {
      \concat { \dynamic p , }
      \italic { con dolce espressione }
    }
  }
}
```

**1<sup>st</sup> movement**  
***p**, con dolce espressione*

Se puede encontrar una lista exhaustiva de instrucciones de cambio y utilización personalizada de las fuentes tipográficas en Sección 1.8.3 [Tipografías], página 279.

También es posible definir conjuntos personalizados de fuentes, tal y como se explica en Sección 1.8.3 [Tipografías], página 279.

## Instrucciones predefinidas

`\teeny`, `\tiny`, `\small`, `\normalsize`, `\large`, `\huge`, `\smaller`, `\larger`.

## Véase también

Referencia de la notación: Sección 1.8.3 [Tipografías], página 279, [Indicaciones dinámicas nuevas], página 137, [Marcas de repetición manual], página 166, Sección 1.8.3 [Tipografías], página 279.

Archivos de inicio: `scm/define-markup-commands.scm`.

Fragmentos de código: Sección “Text” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

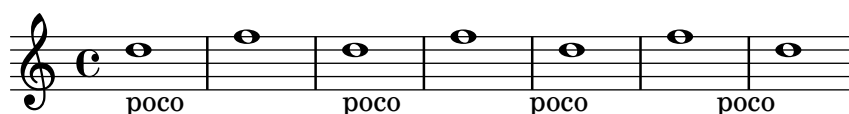
La utilización de las instrucciones de tamaño de texto `\teeny`, `\tiny`, `\small`, `\normalsize`, `\large` y `\huge` conducen a un espaciado de las líneas inconsistente comparado con el uso de `\fontsize`.

## Alineación de texto

Esta subsección trata sobre cómo colocar texto en el modo de marcado. Los objetos de marcado también se pueden mover como un todo, usando la sintaxis que se describe en Sección “Mover objetos” en *Manual de Aprendizaje*.

Los objetos de marcado se pueden alinear de distintas maneras. De forma predeterminada, una indicación de texto se alinea sobre el borde izquierdo: en el ejemplo siguiente, no existe diferencia entre los marcados primero y segundo. El ejemplo muestra también varias formas sintácticamente correctas de ubicar las instrucciones de alineación:

```
\relative {
  d''1-\markup { poco }
  f
  d-\markup { \left-align poco }
  f
  d-\markup { \center-align { poco } }
  f
  d-\markup \right-align { poco }
}
```



Se puede realizar un ajuste fino de la alineación horizontal usando un valor numérico:

```
\relative {
  a'1-\markup { \halign #-1 poco }
  e'
  a,-\markup { \halign #0 poco }
  e'
  a,-\markup { \halign #0.5 poco }
  e'
  a,-\markup { \halign #2 poco }
}
```



Por último, las palabras y otros objetos se pueden desplazar si los hacemos preceder de un relleno. También es posible usar un relleno negativo que hará moverse a los objetos subsiguientes en la dirección opuesta. Aunque normalmente el relleno es invisible, en el ejemplo siguiente se han añadido instrucciones para poder verlo con más claridad:

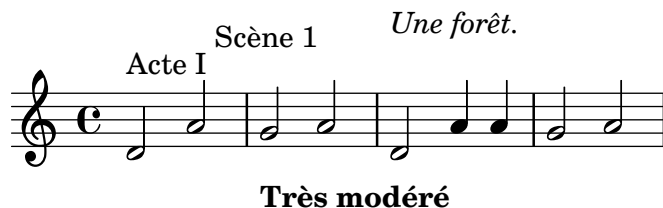
```
\relative {
  d''1-\markup { poco }
  f
  d-\markup { \with-color #darkred \box \hspace #4 poco }
  f
  d-\markup { \with-color #darkred \box \hspace #-4 poco }
  f
  d-\markup { \with-color #darkred \box \hspace #10 poco }
}
```



Ciertos objetos pueden poseer sus propios procedimientos de alineación, y por tanto no resultan afectados por estas instrucciones. Es posible mover estos objetos de marcado como un todo, como se muestra por ejemplo en [Indicaciones de texto], página 260.

La alineación vertical se puede establecer de una forma similar. Como se ha dicho más arriba, los objetos de marcado se pueden mover como un todo; sin embargo, también es posible mover elementos específicos dentro de un bloque de marcado.

```
\relative {
  d'2^\markup {
    Acte I
    \raise #2 { Scène 1 }
  }
  a'
  g_\markup {
    \lower #4 \bold { Très modéré }
  }
  a
  d,^\markup \raise #4 \italic {
    Une forêt.
  }
  a'4 a g2 a
}
```



Algunas instrucciones pueden afectar tanto a la alineación horizontal como a la vertical de los objetos de texto en el modo de marcado:

```
\relative {
  d'2^\markup {
    Acte I
```

```

\translate #'(-1 . 2) "Scène 1"
}
a'
g_\markup {
  \general-align #Y #3.2 \bold "Très modéré"
}
a
d,\markup \translate-scaled #'(-1 . 2) \teeny {
  "Une forêt."
}
a'4 a g2 a
}

```

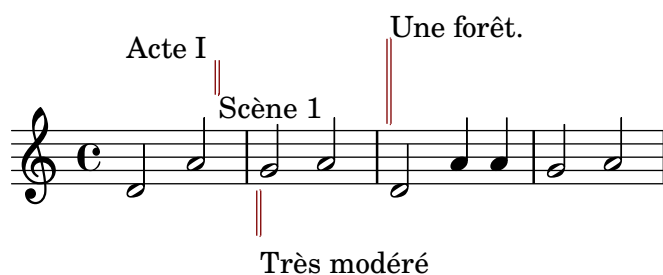


Una vez más resulta muy útil el relleno (ya sea positivo o negativo) para colocar los objetos verticalmente, cuando van dentro de columnas de elementos de marcado:

```

\relative {
  d'2^\markup {
    Acte I
    \column {
      \with-color #darkred \box \vspace #-1
      "Scène 1"
    }
  }
}
a'
g_\markup \column {
  \with-color #darkred \box \vspace #1
  "Très modéré"
}
a
d,\markup \column {
  "Une forêt."
  \with-color #darkred \box \vspace #2
}
a'4 a g2 a
}

```



Un objeto de marcado puede incluir varias líneas de texto. En el ejemplo siguiente, cada elemento o expresión se sitúa en su propia línea, ya sea alineada por la izquierda o centrada:

```

\markup {
  \column {
    a
    "b c"
    \line { d e f }
  }
  \hspace #10
  \center-column {
    a
    "b c"
    \line { d e f }
  }
}

a          a
b c        b c
d e f      d e f

```

De forma similar, una lista de elementos o expresiones se puede repartir de forma que rellene por completo el ancho de la línea horizontal (si hay un solo elemento, se centra en el papel). A su vez, estas expresiones pueden incluir texto de varias líneas o cualquier otra expresión de marcado:

```

\markup {
  \fill-line {
    \line { William S. Gilbert }
    \center-column {
      \huge \smallCaps "The Mikado"
      or
      \smallCaps "The Town of Titipu"
    }
    \line { Sir Arthur Sullivan }
  }
}
\markup {
  \fill-line { 1885 }
}

```

William S. Gilbert

THE MIKADO  
or  
THE TOWN OF TITIPU

Sir Arthur Sullivan

1885

Los elementos se pueden extender para que cubran cualquier anchura especificada mediante la sobreescritura de la propiedad `line-width`. De forma predeterminada está establecida a `#f` lo que implica toda la línea:

```

\markup {
  \column {
    \fill-line { left center right }
    \null
    \override #'(line-width . 30)
    \fill-line { left center right }
  }
}

```

```

    }
  }

  left                center                right

  left      center      right

```

Las indicaciones de texto largas se pueden también ajustar automáticamente según un ancho de línea dado. Estarán alineados por la izquierda o justificados, como se muestra en el ejemplo siguiente.

```

\markup {
  \column {
    \line \smallCaps { La vida breve }
    \line \bold { Acto I }
    \wordwrap \italic {
      (La escena representa el corral de una casa de
      gitanos en el Albaicín de Granada. Al fondo una
      puerta por la que se ve el negro interior de
      una Fragua, iluminado por los rojos resplandores
      del fuego.)
    }
  }
  \hspace #0

  \line \bold { Acto II }
  \override #'(line-width . 50)
  \justify \italic {
    (Calle de Granada. Fachada de la casa de Carmela
    y su hermano Manuel con grandes ventanas abiertas
    a través de las que se ve el patio
    donde se celebra una alegre fiesta)
  }
}

```

LA VIDA BREVE

### Acto I

*(La escena representa el corral de una casa de gitanos en el Albaicín de Granada. Al fondo una puerta por la que se ve el negro interior de una Fragua, iluminado por los rojos resplandores del fuego.)*

### Acto II

*(Calle de Granada. Fachada de la casa de Carmela y su hermano Manuel con grandes ventanas abiertas a través de las que se ve el patio donde se celebra una alegre fiesta)*

Hay una lista exhaustiva de instrucciones de alineación de texto en Sección A.11.2 [Align], página 750.

## Véase también

Manual de aprendizaje: Sección “Mover objetos” en *Manual de Aprendizaje*.

Referencia de la notación: Sección A.11.2 [Align], página 750, [Indicaciones de texto], página 260.

Archivos de inicio: `scm/define-markup-commands.scm`.

Fragmentos de código: Sección “Text” en *Fragmentos de código*.

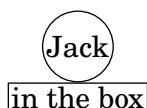
Referencia de funcionamiento interno: Sección “TextScript” en *Referencia de Funcionamiento Interno*.

## Notación gráfica dentro de elementos de marcado

Se puede añadir diversos objetos gráficos a una partitura, utilizando instrucciones de marcado.

Ciertas instrucciones de marcado permiten la decoración de elementos de texto con gráficos, como se muestra en el ejemplo siguiente.

```
\markup \fill-line {
  \center-column {
    \circle Jack
    \box "in the box"
    \null
    \line {
      Erik Satie
      \hspace #3
      \bracket "1866 - 1925"
    }
    \null
    \rounded-box \bold Prelude
  }
}
```



in the box

Erik Satie [1866 - 1925]

**Prelude**

Algunas instrucciones pueden requerir un aumento del relleno alrededor del texto: esto se puede conseguir con algunas instrucciones de marcado descritas exhaustivamente en Sección A.11.2 [Align], página 750.

```
\markup \fill-line {
  \center-column {
    \box "Charles Ives (1874 - 1954)"
    \null
    \box \pad-markup #2 "THE UNANSWERED QUESTION"
    \box \pad-x #8 "A Cosmic Landscape"
    \null
  }
}
\markup \column {
  \line {
    \hspace #10
    \box \pad-to-box #'(-5 . 20) #'(0 . 5)
```

```

\bold "Largo to Presto"
}
\pad-around #3
"String quartet keeps very even time,
Flute quartet keeps very uneven time."
}

```

Charles Ives (1874 - 1954)

THE UNANSWERED QUESTION

A Cosmic Landscape

**Largo to Presto**

String quartet keeps very even time, Flute quartet keeps very uneven time.

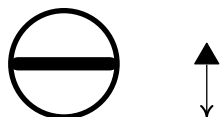
Se pueden imprimir otros símbolos o elementos gráficos sin que se requiera ningún otro texto. De igual manera que en el caso de las expresiones de marcado, dichos objetos se pueden combinar:

```

\markup {
  \combine
    \draw-circle #4 #0.4 ##f
    \filled-box #'(-4 . 4) #'(-0.5 . 0.5) #1
  \hspace #5

  \center-column {
    \triangle ##t
    \combine
      \draw-line #'(0 . 4)
      \arrow-head #Y #DOWN ##f
  }
}

```



Entre las funcionalidades gráficas avanzadas se encuentran la posibilidad de incluir archivos de imagen externos convertidos al formato de PostScript Encapsulado (*eps*), y la inclusión directa de gráficos dentro del archivo de entrada, usando código PostScript nativo. En tal caso puede ser de utilidad la especificación explícita del tamaño del dibujo, como se ejemplifica a continuación:

```

c'1^\markup {
  \combine
    \epsfile #X #10 "./context-example.eps"
    \with-dimensions #'(0 . 6) #'(0 . 10)
  \postscript "
    -2 3 translate
    2.7 2 scale

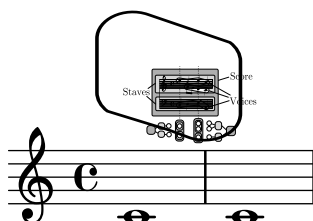
```



```

newpath
2 -1 moveto
4 -2 4 1 1 arct
4 2 3 3 1 arct
0 4 0 3 1 arct
0 0 1 -1 1 arct
closepath
stroke"
}
c'

```



Hay una lista exhaustiva de instrucciones específicas de gráficos en Sección A.11.3 [Graphic], página 765.

## Véase también

Referencia de la notación: Sección A.11.2 [Align], página 750, [\[<undefined>\]](#) [[<undefined>](#)], página [\[<undefined>\]](#), Sección 1.7 [Anotaciones editoriales], página 235, Sección A.11.3 [Graphic], página 765.

Archivos de inicio: `scm/define-markup-commands.scm`, `scm/stencil.scm`.

Fragmentos de código: Sección “Text” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” en *Referencia de Funcionamiento Interno*.

## Notación musical dentro de elementos de marcado

Se pueden añadir a la partitura diversos elementos de notación musical, dentro de un objeto de marcado.

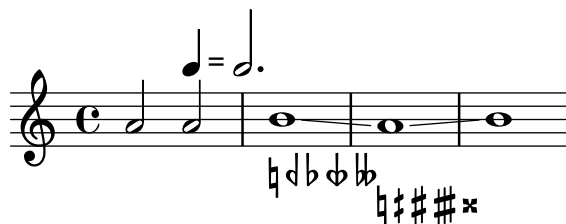
Las notas y las alteraciones se pueden escribir utilizando instrucciones de marcado:

```

a'2 a'^\markup {
  \note {4} #1
  =
  \note-by-number #1 #1 #1.5
}
b'1_\markup {
  \natural \semiflat \flat
  \sesquiflat \doubleflat
}
\glissando
a'1_\markup {
  \natural \semisharp \sharp
  \sesquisharp \doublesharp
}

```

`\glissando b'`



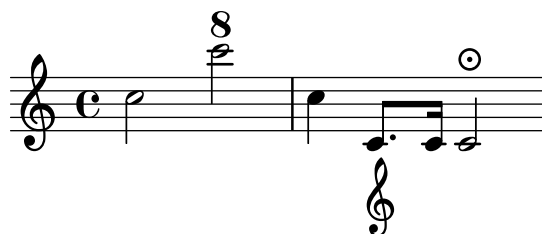
Otros objetos de notación se pueden también imprimir en el modo de marcado:

```
\relative {
  g1 bes
  ees\finger \markup \tied-lyric "4~1"
  fis\_markup { \dynamic rf }
  bes^\markup {
    \beam #8 #0.1 #0.5
  }
  cis
  d-\markup {
    \markalphabet #8
    \markletter #8
  }
}
```



De forma más general, cualquier símbolo musical disponible se puede incluir por separado dentro de un objeto de marcado, como se ejemplifica a continuación; hay una lista exhaustiva de estos símbolos y sus nombres en Sección A.8 [La tipografía Emmentaler], página 713.

```
\relative {
  c''2
  c'^\_markup { \musicglyph "eight" }
  c,4
  c,8._\markup { \musicglyph "clefs.G_change" }
  c16
  c2^\markup { \musicglyph "timesig.neomensural94" }
}
```



Otra forma de imprimir glifos que no son de texto se encuentra descrita en [Explicación de las fuentes tipográficas], página 280. Tiene la utilidad de imprimir llaves de distintos tamaños.

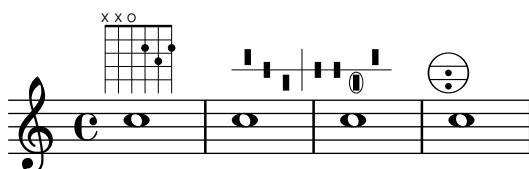
El modo de marcado también contempla diagramas para instrumentos específicos:

```
\relative {
```

```

c''1^\markup {
  \fret-diagram-terse "x;x;o;2;3;2;"
}
c^\markup {
  \harp-pedal "^-v|--ov^"
}
c
c^\markup {
  \combine
  \musicglyph "accordion.discant"
  \combine
  \raise #0.5 \musicglyph "accordion.dot"
  \raise #1.5 \musicglyph "accordion.dot"
}
}

```



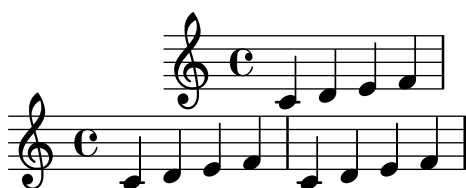
Dichos diagramas se encuentran documentados en Sección A.11.5 [Instrument Specific Markup], página 780.

Incluso una partitura completa se puede incluir dentro de un objeto de marcado:

```

\relative {
  c'4 d^\markup {
    \score {
      \relative { c'4 d e f }
    }
  }
  e f |
  c d e f
}

```



Hay una lista exhaustiva de instrucciones relacionadas con la notación musical en Sección A.11.4 [Music], página 774.

## Véase también

Referencia de la notación: Sección A.11.4 [Music], página 774, Sección A.8 [La tipografía Emmentaler], página 713, [Explicación de las fuentes tipográficas], página 280.

Archivos de inicio: `scm/define-markup-commands.scm`, `scm/fret-diagrams.scm`, `scm/harp-pedals.scm`.

Fragmentos de código: Sección “Text” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

El espaciado vertical de un `\score` dentro de un objeto de marcado está controlado por `baseline-skip`. Todo ajuste del bloque `\paper` se ignora.

## Elementos de marcado de varias páginas

Aunque los objetos de marcado estándar no se pueden dividir, una sintaxis específica hace posible la introducción de líneas de texto que pueden abarcar varias páginas:

```
\markuplist {
  \justified-lines {
    Un texto muy largo de líneas justificadas.
    ...
  }
  \wordwrap-lines {
    Otro párrafo muy largo.
    ...
  }
  ...
}
```

Un texto muy largo de líneas justificadas. ...

Otro párrafo muy largo. ...

...

Esta sintaxis acepta una lista de elementos de marcado, que pueden ser

- el resultado de una instrucción de lista de marcado,
- una lista de marcados,
- o una lista de listas de marcado.

Hay una lista exhaustiva de las instrucciones de lista de marcado en Sección A.12 [Instrucciones de listas de marcado de texto], página 796.

## Véase también

Referencia de la notación: Sección A.12 [Instrucciones de listas de marcado de texto], página 796.

Archivos de inicio: `scm/define-markup-commands.scm`.

Manual de extensión: Sección “Definición de nuevas instrucciones de lista de marcado” en *Extender*.

Fragmentos de código: Sección “Text” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” en *Referencia de Funcionamiento Interno*.

## Instrucciones predefinidas

`\markuplist`.

### 1.8.3 Tipografías

Esta sección presenta la forma en que se manejan las fuentes tipográficas, y cómo se pueden cambiar en las partituras.

## Explicación de las fuentes tipográficas

Las fuentes tipográficas se manejan a través de distintas bibliotecas. FontConfig se utiliza para detectar las tipografías disponibles; las tipografías seleccionadas se trazan utilizando Pango.

Además de cualquier fuente tipográfica que ya esté instalada en el sistema operativo, se pueden añadir fuentes adicionales al conjunto que detecta FontConfig (y por ello disponibles en las partituras de LilyPond) a través de las instrucciones siguientes:

```
#(ly:font-config-add-font "ruta/hacia/archivo-de-fuente.otf")
```

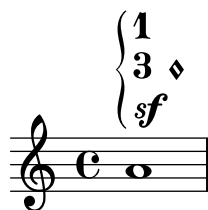
```
#(ly:font-config-add-directory "ruta/hacia/directorio/")
```

Las dos instrucciones admiten rutas absolutas y relativas, lo que permite compilar una partitura en cualquier sistema, simplemente copiando los archivos de fuente necesarios junto al código fuente de la partitura.

Cuando las fuentes han sido instaladas y detectadas, se pueden seleccionar usando los métodos que se describen en [Fuentes de un solo elemento], página 283, y en [Fuentes tipográficas del documento completo], página 284. Dado que imprime la lista completa de las fuentes tipográficas disponibles en el archivo de registro de la consola, puede ser de utilidad la instrucción  `#(ly:font-config-display-fonts)` para verificar qué fuentes se están efectivamente detectando, y bajo qué nombres (ya que pueden ser diferentes de los propios nombres de archivo).

Las tipografías de notación musical se pueden describir como un conjunto de glifos específicos, ordenados en varias familias. La siguiente sintaxis permite usar directamente varios glifos de la tipografía Feta de LilyPond distintas a las de texto en el modo de marcado:

```
a'1^\markup {
  \vcenter {
    \override #'(font-encoding . fetaBraces)
    \lookup "brace120"
    \override #'(font-encoding . fetaText)
    \column { 1 3 sf }
    \override #'(font-encoding . fetaMusic)
    \lookup "noteheads.s0petrucci"
  }
}
```



Sin embargo, todos estos glifos excepto las llaves de varios tamaños que están contenidas en la fuente tipográfica **fetaBraces** están disponibles utilizando la sintaxis más sencilla que se describe en [Notación musical dentro de elementos de marcado], página 276.

Cuando se usan los glifos de la fuente de llaves **fetaBraces**, el tamaño de la llave se especifica por la parte numérica del nombre del glifo, en unidades arbitrarias. Se puede especificar cualquier entero de 0 a 575 inclusive, siendo 0 el que produce la llave más pequeña. El valor óptimo se debe determinar por ensayo y error. Todos estos glifos son llaves izquierdas; las llaves derechas se pueden obtener mediante rotación, véase Sección 5.4.9 [Rotación de objetos], página 677.

Están disponibles tres familias de fuentes tipográficas para el texto: la fuente *roman* (romana, serif o con remates), la fuente *sans* (sin remates), y la fuente *typewriter* o monoespaciada.

Para el *backend* **svg**:

Familia	Fuente predeterminada
<i>roman</i>	<b>serif</b>
<i>sans</i>	<b>sans-serif</b>
<i>typewriter</i>	<b>monospace</b>

**serif**, **sans-serif** y **monospace** son **generic-family** (las familias estándar) en las especificaciones de SVG y CSS.

Para otros *backends*:

Familia	Fuente predeterminada (alias)	Listas de definición del alias
<i>roman</i>	LilyPond Serif	TeX Gyre Schola, C059, Century SchoolBook URW, Century Schoolbook L, DejaVu Serif, ..., serif
<i>sans</i>	LilyPond Sans Serif	TeX Gyre Heros, Nimbus Sans, Nimbus Sans L, DejaVu Sans, ..., sans-serif
<i>typewriter</i>	LilyPond Monospace	TeX Gyre Cursor, Nimbus Mono PS, Nimbus Mono, Nimbus Mono L, DejaVu Sans Mono, ..., monospace

LilyPond Serif, LilyPond Sans Serif y LilyPond Monospace son alias de las fuentes tipográficas que están definidos en el archivo dedicado de configuración de LilyPond `00-lilypond-fonts.conf`. Allí donde un carácter no exista en la primera fuente de la lista, será sustituido por el carácter correspondiente extraído de la siguiente fuente de la lista. Para ver más detalles acerca de las definiciones de los alias, consulte `00-lilypond-fonts.conf` dentro de la carpeta de instalación del programa.

Cada familia puede incluir distintas formas y series. El ejemplo siguiente muestra la posibilidad de seleccionar familias, formas, series y tamaños alternativos. El valor que se da a **font-size** es el cambio requerido a partir del tamaño predeterminado.

```
\override Score.RehearsalMark.font-family = #'typewriter
\mark \markup "Ouverture"
\override Voice.TextScript.font-shape = #'italic
\override Voice.TextScript.font-series = #'bold
d''2.^{\markup "Allegro"}
\override Voice.TextScript.font-size = #-3
c''4^smaller
```



Se puede usar una sintaxis similar en el modo de marcado, aunque en este caso es preferible usar la sintaxis sencilla que se explica en [Seleccionar la tipografía y su tamaño], página 266:

```
\markup {
  \column {
    \line {
      \override #'((font-shape . italic) (font-size . 4))
      Idomeneo,
    }
    \line {
```

```

\override #'(font-family . typewriter)
{
  \override #'(font-series . bold)
  re
  di
}
\override #'(font-family . sans)
Creta
}
}
}

```

***Idomeneo,***  
**re** di Creta

Al usar fuentes tipográficas de OpenType, pueden usarse las funcionalidades de fuente o *font features*. No se pueden especificar los idiomas ni los *scripts* de OpenType. Nota: no todas las fuentes de OpenType tienen todas las funciones. Si se requiere una funcionalidad que no existe en la fuente seleccionada, la funcionalidad sencillamente se ignora.

```

% Versalitas auténticas
\markup { Estilo normal: Hola HOLA }
\markup { \caps { Falsa versalita: Hola } }
\markup { \override #'(font-features . ("smcp"))
          { Versalitas auténticas: Hola } }

% Estilos de números
\markup { Cifras de estilo normal: 0123456789 }
\markup { \override #'(font-features . ("onum"))
          { Cifras de estilo antiguo: 0123456789 } }

% Alternativas de estilo
\markup { \override #'(font-features . ("salt 0"))
          { Alternativa de estilo 0:  $\epsilon\phi\pi\rho\theta$  } }
\markup { \override #'(font-features . ("salt 1"))
          { Alternativa de estilo 1:  $\epsilon\phi\pi\rho\theta$  } }

% Acumulación de funcionalidades
\markup { \override #'(font-features . ("onum" "smcp" "salt 1"))
          { Acumulación de funcionalidades: Hola 0123456789  $\epsilon\phi\pi\rho\theta$  } }

```

Estilo normal: Hola HOLA

FALSA VERSALITA: HOLA

Versalitas auténticas: Hola

Cifras de estilo normal: 0123456789

Cifras de estilo antiguo: 0123456789

Alternativa de estilo 0:  $\epsilon\phi\pi\rho\theta$

### Alternativa de estilo 1: $\epsilon\phi\pi\rho\theta$

### Acumulación de funcionalidades: Hola 0123456789 $\epsilon\phi\pi\rho\theta$

Para ver la lista completa de funcionalidades de fuente de OpenType font, consulte: <https://www.microsoft.com/typography/otspec/featurelist.htm>

Para la identificación de las funcionalidades de fuente de OpenType, consulte: <http://lists.gnu.org/archive/html/lilypond-devel/2017-08/msg00004.html>

Aunque es fácil cambiar entre las fuentes tipográficas preconfiguradas, también es posible usar otras fuentes, como se explica en las siguientes secciones: [Fuentes de un solo elemento], página 283, y [Fuentes tipográficas del documento completo], página 284.

### Véase también

Referencia de la notación: Sección A.8 [La tipografía Emmentaler], página 713, [Notación musical dentro de elementos de marcado], página 276, Sección 5.4.9 [Rotación de objetos], página 677, [Seleccionar la tipografía y su tamaño], página 266, Sección 1.8.3 [Tipografías], página 279.

### Fuentes de un solo elemento

Se puede usar en una partitura cualquier fuente tipográfica que esté instalada en el sistema operativo y reconocida por parte de FontConfig, usando la siguiente sintaxis:

```
\override Staff.TimeSignature.font-name = "Bitstream Charter"
\override Staff.TimeSignature.font-size = #2
\time 3/4

a'1_\markup {
  \override #'(font-name . "Bitstream Vera Sans,sans-serif, Oblique Bold")
    { Vera Oblique Bold }
}
```



*font-name* puede describirse usando una lista de ‘fuentes tipográficas’ separadas por comas y una lista de ‘estilos’ separados por espacios. En la medida en que la ‘fuente tipográfica’ de la lista esté instalada y contenga el glifo solicitado, se usará. En caso contrario, se usará como sustitución la *siguiente* fuente tipográfica de la lista.

La ejecución de lilypond con la opción siguiente presenta una lista de todas las tipografías disponibles en el sistema operativo:

```
lilypond -dshow-available-fonts
```

### Véase también

Referencia de la notación: [Explicación de las fuentes tipográficas], página 280, [Fuentes tipográficas del documento completo], página 284.

Fragmentos de código: Sección “Text” en *Fragmentos de código*.

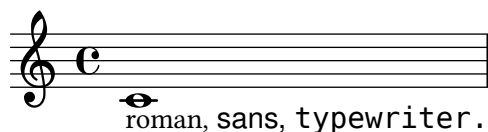


## Fuentes tipográficas del documento completo

Se pueden cambiar las fuentes tipográficas que se usan como tipografías predeterminadas en las familias *romana*, *sans-serif* y *monoespaciada*, especificándolas en este orden como se ve en el ejemplo de abajo, que escala las tipografías automáticamente con el valor fijado para el tamaño global del pentagrama. De forma parecida a [Fuentes de un solo elemento], página 283, se puede describir usando una lista de ‘fuentes tipográficas’ separadas por comas. Sin embargo, los ‘estilos’ de fuente no se pueden describir. Para leer más explicaciones sobre las fuentes tipográficas, consulte [Explicación de las fuentes tipográficas], página 280.

```
\paper {
  #(define fonts
    (make-pango-font-tree "Linux Libertine O"
                        "Nimbus Sans, Nimbus Sans L"
                        "DejaVu Sans Mono"
                        (/ staff-height pt 20)))
}
```

```
\relative c'{
  c1-\markup {
    roman,
    \sans sans,
    \typewriter typewriter. }
}
```



Nota: `make-pango-font-tree` reinicia las fuentes tipográficas al tipo predeterminado `Emmentaler`.

La sintaxis siguiente nos permite cambiar las tipografías específicas, dejando el resto en sus valores predeterminados. El siguiente ejemplo tiene el mismo efecto que el ejemplo anterior de utilización de `make-pango-font-tree`. De igual manera que con `make-pango-font-tree`, podemos especificar una lista de valores separados por comas de ‘fonts’ para las categorías Romana, Sans serif y Monoespaciada (`roman`, `sans` y `typewriter`). Si no queremos cambiar el tamaño del pentagrama a otro distinto del predeterminado de 20 pt, no es necesaria la parte `#:factor (/ staff-height pt 20)`.

```
\paper {
  #(define fonts
    (set-global-fonts
      #:roman "Linux Libertine O"
      #:sans "Nimbus Sans, Nimbus Sans L"
      #:typewriter "DejaVu Sans Mono"
      #:factor (/ staff-height pt 20) ; innecesario si el tamaño del pentagrama es el predeterminado
    ))
}
```

También se pueden especificar las fuentes de la notación. El ejemplo siguiente también tiene el mismo efecto que en los casos anteriores, porque establece las fuentes para la notación como las predeterminadas. Para más información, véase Sección 3.4.4 [Sustituir la tipografía de la notación], página 550.

```
\paper {
```

```

#(define fonts
  (set-global-fonts
    #:music "emmentaler"          ; predeterminado
    #:brace "emmentaler"          ; predeterminado
    #:roman "Linux Libertine O"
    #:sans "Nimbus Sans, Nimbus Sans L"
    #:typewriter "DejaVu Sans Mono"
    #:factor (/ staff-height pt 20) ; innecesario si el tamaño del pentagrama es el p
  ))
}

```

Nota: cada llamada a `set-global-fonts` reinicia completamente tanto la fuente tipográfica de la notación como las de texto. Si se deja sin especificar una cualquiera de las categorías, entonces se utiliza la fuente predeterminada para dicha categoría. Cada llamada a `set-global-fonts` afecta a cada uno de los bloques `\book` que le siguen. Si existen varios bloques `\book` y queremos usar distintas fuentes tipográficas para cada uno de ellos, sencillamente llamamos de nuevo a `set-global-fonts`, así:

```

\paper {
  #(define fonts
    (set-global-fonts
      ...
    ))
}
\book {
  ...
}

\paper {
  #(define fonts
    (set-global-fonts
      ...
    ))
}
\book {
  ...
}

```

## Véase también

Referencia de la notación: [Explicación de las fuentes tipográficas], página 280, [Fuentes de un solo elemento], página 283, [Seleccionar la tipografía y su tamaño], página 266, Sección 1.8.3 [Tipografías], página 279, Sección 3.4.4 [Sustituir la tipografía de la notación], página 550.

## 2 Notación especializada

Este capítulo explica cómo crear notación musical para instrumentos de un tipo específico o dentro de estilos específicos.

La música orquestal y la de grupos menores no se estudian en ningún apartado distinto; sin embargo, varias nociones que son con frecuencia útiles en las partituras orquestales pueden verse en [Referencias para ópera y musicales], página 331.

### 2.1 Música vocal

**Recitativo**  
Baritono

O Freun - - de, nicht die - se Töne!

Sondern laßt uns an - - ge -

nehmere an - stimmen, und freu -

- - - - - *ad libitum* denvollere!

Esta sección explica cómo tipografiar música vocal, y cómo asegurarse de que la letra se alinea con las notas de su melodía correspondiente.

#### 2.1.1 Notación común para música vocal

Esta sección se ocupa de las cuestiones comunes a la mayoría de los tipos de música vocal.

#### Referencias para música vocal

Esta sección indica dónde encontrar los detalles de las cuestiones de notación que pueden surgir en cualquier tipo de música vocal.

- Casi todos los estilos de música vocal utilizan texto escrito como letra. Hay una introducción a esta notación en Sección “Elaborar canciones sencillas” en *Manual de Aprendizaje*.
- La música vocal probablemente requiere el uso del modo de marcado o **markup**, ya sea para la letra o para otros elementos de texto (nombres de los personajes, etc.). Esta sintaxis está descrita en [Introducción al marcado de texto], página 264.
- Los *ambitus* o indicaciones de tesitura vocal se pueden añadir al principio de los pentagramas vocales, como se explica en [Tesitura], página 38.
- De forma predeterminada, las indicaciones dinámicas se sitúan debajo del pentagrama, pero en la música coral se suelen disponer por encima del pentagrama para evitar la letra, como se explica en [Disposiciones de la partitura para música coral], página 329.

## Véase también

Glosario musical: Sección “ambitus” en *Glosario Musical*.

Manual de aprendizaje: Sección “Elaborar canciones sencillas” en *Manual de Aprendizaje*.

Referencia de la notación: [Introducción al marcado de texto], página 264, [Tesitura], página 38, [Disposiciones de la partitura para música coral], página 329.

Fragmentos de código: Sección “Vocal music” en *Fragmentos de código*.

## Introducir la letra

La letra de las canciones se introduce en un modo de entrada especial que se inicia mediante la palabra clave `\lyricmode`, o bien mediante `\addlyrics` ó `\lyricsto`. En este modo especial de entrada, el carácter `d` no se analiza como una nota, sino más bien como una sílaba de una sola letra. Dicho de otra forma, las sílabas se introducen como las notas, pero la altura de las notas se sustituye por texto.

Por ejemplo:

```
\lyricmode { Three4 blind mice,2 three4 blind mice2 }
```

Existen dos métodos principales para especificar la colocación horizontal de las sílabas: mediante la indicación de la duración de cada sílaba explícitamente, como en el ejemplo anterior, o dejando que la letra se alinee automáticamente con una melodía u otra voz de música, utilizando `\addlyrics` o `\lyricsto`. El primer método se describe a continuación bajo el epígrafe [Duración manual de las sílabas], página 293. El segundo método se describe en [Duración automática de las sílabas], página 290.

Una palabra o sílaba de la letra comienza con un carácter alfabético (y algunos otros caracteres, véase más abajo) y termina con un espacio o un dígito. El resto de los caracteres dentro de la sílaba pueden ser cualesquier caracteres que no sean dígitos o espacios.

A causa de que cualquier carácter que no es un dígito o un espacio se considera parte de la sílaba, una palabra es válida incluso si termina con `}`, lo que a menudo lleva al siguiente fallo:

```
\lyricmode { la la la }
```

En este ejemplo, el símbolo `}` se encuentra incluido dentro de la sílaba final, por tanto la llave de apertura no se compensa con la correspondiente llave de cierre y el archivo de entrada probablemente no se podrá procesar. En vez de ello, las llaves siempre deben ir rodeadas por espacios:

```
\lyricmode { la la la }
```

Los signos de puntuación, las letras con caracteres acentuados o de cualquier idioma distinto del inglés, o los caracteres especiales (como el símbolo del corazón o las comillas inclinadas), se pueden insertar directamente en el archivo de entrada, siempre que éste se grabe con la codificación UTF-8. Para más información, consulte Sección 3.3.3 [Caracteres especiales], página 546.

```
\relative { d''8 c16 a bes8 f ees' d c4 }
\addlyrics { „Schad' um das schö -- ne grü -- ne Band, }
```



Se pueden usar las comillas normales dentro de la letra, pero deben ir precedidas por el carácter de barra invertida y la sílaba en su conjunto se debe encerrar en otro par de comillas. Por ejemplo,

```
\relative { \time 3/4 e'4 e4. e8 d4 e d c2. }
```

```
\addlyrics { "\"I" am so lone -- "ly,\"" said she }
```



La definición completa del comienzo de una palabra en el modo Lyrics (letra) es algo más compleja. Una palabra en el modo Lyrics comienza por: un carácter alfabético, `_`, `?`, `!`, `:`, `'`, los caracteres de control desde `^A` hasta `^F`, desde `^Q` hasta `^W`, `^Y`, `^^`, cualquier carácter de 8 bits con código ASCII por encima del 127, o una combinación de dos caracteres consistente en la combinación de una barra invertida seguida por ```, `'`, `"` ó `^`.

Un enorme control sobre la apariencia de la letra proviene del uso de elementos de marcado `\markup` dentro de la propia letra. Para ver una explicación de muchas de las opciones, consulte Sección 1.8.2 [Formatear el texto], página 264.

## Fragmentos de código seleccionados

### *Dar formato a sílabas de la letra*

Es posible usar el modo de marcado para dar formato a sílabas individuales dentro de la letra.

```
mel = \relative c'' { c4 c c c }
lyr = \lyricmode {
  Lyrics \markup { \italic can } \markup { \with-color #red contain }
  \markup { \fontsize #8 \bold Markup! }
}

<<
  \new Voice = melody \mel
  \new Lyrics \lyricsto melody \lyr
>>
```



## Véase también

Manual de aprendizaje: Sección “Canciones” en *Manual de Aprendizaje*.

Referencia de la notación: [Duración automática de las sílabas], página 290, Sección 1.8.3 [Tipografías], página 279, Sección 1.8.2 [Formatear el texto], página 264, Sección 5.4.1 [Modos de entrada], página 660, [Duración manual de las sílabas], página 293, Sección 3.3.3 [Caracteres especiales], página 546.

Referencia de funcionamiento interno: Sección “LyricText” en *Referencia de Funcionamiento Interno*.

Fragmentos de código: Sección “Text” en *Fragmentos de código*.

## Alineación de la letra a una melodía

La letra se interpreta en el modo `\lyricmode` y se imprime dentro del contexto llamado `Lyrics`, véase Sección 5.1.1 [Explicación de los contextos], página 622.

```
\new Lyrics \lyricmode { ... }
```

La letra se puede alinear con la melodía de dos maneras principales:

- Se puede alinear la letra automáticamente, tomándose las duraciones de las sílabas de otra voz de música, o (en circunstancias especiales) una melodía asociada, usando `\addlyrics`, `\lyricsto`, o estableciendo el valor de la propiedad `associatedVoice`. Para ver más detalles, consulte [Duración automática de las sílabas], página 290.



Life is\_\_ love,      live\_\_ life.  
No more let sins and sor-rows grow.

La segunda línea de texto muestra cómo se puede cambiar la voz de la que se toman las duraciones para las sílabas. Esto es útil si el texto de las distintas estrofas corresponde a las mismas notas de formas diferentes, y todas las duraciones están disponibles dentro de contextos de voz. Para ver más detalles, consulte Sección 2.1.3 [Versos], página 319.

- El texto se puede alinear independientemente de la duración de cualquier nota si las duraciones de las sílabas se especifican explícitamente, y se escriben con `\lyricmode`.

```
<<
  \new Voice = "one" \relative {
    \time 2/4
    c''4 b8. a16 g4. f8 e4 d c2
  }

% uses previous explicit duration of 2;
  \new Lyrics \lyricmode {
    Joy to the earth!
  }

% explicit durations, set to a different rhythm
  \new Lyrics \lyricmode {
    Life4 is love,2. live4 life.2
  }
>>
```



El primer verso no está alineado con las notas porque no se especificaron las duraciones, y se utiliza el valor previo de 2 (blanca) para todas las sílabas.

El segundo verso muestra cómo las palabras se pueden alinear de forma independiente de las notas. Esto es útil si el texto de los distintos versos se corresponde con las notas de maneras diferentes, pero las duraciones requeridas no están disponibles en un contexto de música. Para ver más detalles, consulte [Duración manual de las sílabas], página 293. Esta técnica también es útil cuando se quiere preparar un diálogo encima de la música; para ver ejemplos que lo muestran, consulte [Diálogos encima de la música], página 338.

## Véase también

Manual de aprendizaje: Sección “Alineación de la letra a una melodía” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.1.1 [Explicación de los contextos], página 622, [Duración automática de las sílabas], página 290. Sección 2.1.3 [Versos], página 319, [Duración manual de las sílabas], página 293, [Diálogos encima de la música], página 338, [Duración manual de las sílabas], página 293.

Referencia de funcionamiento interno: Sección “Lyrics” en *Referencia de Funcionamiento Interno*.

## Duración automática de las sílabas

Las sílabas de la letra se pueden alinear automáticamente con las notas de una melodía de tres formas:

- especificando por su nombre el contexto Voice que contiene la melodía, con `\lyricsto`,
- introduciendo la letra con `\addlyrics` y colocándola inmediatamente después del contexto de voz que contiene la melodía,

- estableciendo un valor para la propiedad `associatedVoice`, la alineación de la letra se puede mover a un contexto de voz distinto, especificado por su nombre, en cualquier momento musical.

En los tres métodos se pueden trazar guiones de separación entre las sílabas de una palabra y líneas extensoras después del final de una palabra. Para ver más detalles, véase [Líneas de extensión y guiones], página 298.

El contexto `Voice` que contiene la melodía con que se está alineando la letra, no debe haber “muerto”, o se perderá la letra a partir de este punto. Esto puede ocurrir si existen períodos en que dicha voz no tiene nada que hacer. Para ver métodos para mantener vivos los contextos, consulte Sección 5.1.3 [Mantener vivos los contextos], página 628.

## Uso de `\lyricsto`

Las sílabas de la letra se pueden alinear automáticamente bajo las notas de una melodía especificando por su nombre el contexto de voz que contiene la melodía, con `\lyricsto`:

```
<<
  \new Voice = "melody" \relative {
    a'1 a4. a8 a2
  }
  \new Lyrics \lyricsto "melody" {
    These are the words
  }
>>
```



Esto alinea las sílabas con las notas del contexto `Voice` nombrado, que debe existir previamente. Por tanto, normalmente se especifica primero el contexto `Voice` seguido del contexto `Lyrics`. La letra en sí sigue a la instrucción `\lyricsto`. La instrucción `\lyricsto` invoca automáticamente el modo de letra. De forma predeterminada, la letra se coloca por debajo de las notas. Para otras colocaciones, consulte [Posicionamiento vertical de la letra], página 301.

## Uso de `\addlyrics`

La instrucción `\addlyrics` es realmente tan sólo una forma cómoda de escribir una estructura de LilyPond más complicada que establece la letra.

```
{ MÚSICA }
\addlyrics { LETRA }
```

es lo mismo que

```
\new Voice = "blabla" { MÚSICA }
\new Lyrics \lyricsto "blabla" { LETRA }
```

He aquí un ejemplo:

```
{
  \time 3/4
  \relative { c'2 e4 g2. }
  \addlyrics { play the game }
```

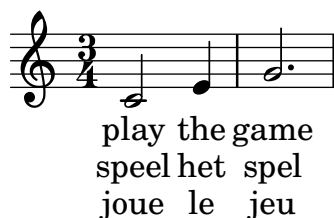


}



Se pueden añadir más versos poniendo más secciones `\addlyrics`:

```
{
  \time 3/4
  \relative { c'2 e4 g2. }
  \addlyrics { play the game }
  \addlyrics { speel het spel }
  \addlyrics { joue le jeu }
}
```



La instrucción `\addlyrics` no es capaz de manejar situaciones de polifonía. Asimismo, no puede usarse para asociar letra a un contexto de tablatura `TabVoice`. Para estos casos debería usar `\lyricsto`.

## Uso de `associatedVoice`

Se puede cambiar la melodía a la que se alinea la letra mediante el establecimiento de la propiedad `associatedVoice`,

```
\set associatedVoice = "lala"
```

El valor de la propiedad (aquí: `"lala"`) debe ser el nombre de un contexto `Voice`. Por razones técnicas, la instrucción `\set` se debe escribir una sílaba antes de aquella a la que se quiere aplicar el cambio de voz.

He aquí un ejemplo que muestra su uso:

```
<<
\new Staff <<
  \time 2/4
  \new Voice = "one" \relative {
    \voiceOne
    c''4 b8. a16 g4. r8 a4 ( b ) c2
  }
  \new Voice = "two" \relative {
    \voiceTwo
    s2 s4. f'8 e8 d4. c2
  }
  }
>>
% takes durations and alignment from notes in "one" initially
% then switches to "two"
\new Lyrics \lyricsto "one" {
  No more let
```

```

\set associatedVoice = "two" % must be set one syllable early
sins and sor -- rows grow.
}
>>

```



## Véase también

Referencia de la notación: [Líneas de extensión y guiones], página 298, Sección 5.1.3 [Mantener vivos los contextos], página 628, [Posicionamiento vertical de la letra], página 301.

## Duración manual de las sílabas

En ciertas músicas vocales complejas puede ser deseable colocar la letra de forma completamente independiente de las notas. En este caso, no use `\lyricsto` ni `\addlyrics` y no establezca un valor para `associatedVoice`. Las sílabas se escriben como notas – pero sustituyendo los nombres de las notas por texto – y la duración de cada sílaba se escribe explícitamente después de la sílaba.

Se pueden trazar líneas separadores entre las sílabas, como es usual, pero no se pueden trazar líneas extensoras cuando no hay una voz asociada.

He aquí dos ejemplos:

```

<<
\new Voice = "melody" \relative {
  c''2 a f f e e
}
\new Lyrics \lyricmode {
  c4. -- a -- f -- f -- e2. -- e
}
>>

```



```

<<
\new Staff {
  \relative {
    c''2 c2
    d1
  }
}
\new Lyrics {
  \lyricmode {
    I2 like4. my8 cat!1
  }
}
\new Staff {
  \relative {

```

```

      c'8 c c c c c c c
      c8 c c c c c c c
    }
  }
>>

```



Esta técnica es muy útil cuando se escriben diálogos encima de la música, véase [Diálogos encima de la música], página 338.

Para cambiar la alineación de las sílabas, simplemente sobrescriba la propiedad `self-alignment-X`:

```

<<
  \new Voice = "melody" \relative {
    \time 3/4
    c'2 e4 g2 f
  }
  \new Lyrics \lyricmode {
    \override LyricText.self-alignment-X = #LEFT
    play1 a4 game4
  }
>>

```



## Véase también

Referencia de la notación: Sección 5.1.3 [Mantener vivos los contextos], página 628.

Referencia de funcionamiento interno: Sección “Lyrics” en *Referencia de Funcionamiento Interno*, Sección “Voice” en *Referencia de Funcionamiento Interno*.

## Varias sílabas sobre una nota

Para asignar más de una sílaba a una única nota con espacios entre las sílabas, podemos encerrar la frase entre comillas o utilizar un carácter de guión bajo `_`. De forma alternativa, podemos usar el símbolo de tilde curva (`~`) para obtener una ligadura de texto.

```

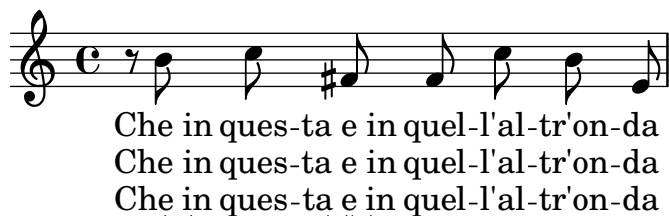
{
  \relative {
    \autoBeamOff
    r8 b' c fis, fis c' b e,
  }
  \addlyrics
  {
    \override LyricHyphen.minimum-distance = #1.0 % Ensure hyphens are visible
    Che_in ques -- ta_e_in quel -- l'al -- tr'on -- da
  }
}

```

```

}
\addlyrics { "Che in" ques -- "ta e in" quel -- l'al -- tr'on -- da }
\addlyrics { Che~in ques -- ta~e~in quel -- l'al -- tr'on -- da }
}

```



## Véase también

Referencia de funcionamiento interno: Sección “LyricCombineMusic” en *Referencia de Funcionamiento Interno*.

## Varias notas sobre una sílaba

A veces, y sobre todo en la música medieval y del Barroco, varias notas se cantan sobre una sílaba única; tales vocalizaciones reciben el nombre de melismas, o melismata (véase Sección “melisma” en *Glosario Musical*). La sílaba de un melisma se suele alinear por la izquierda con la primera nota del melisma.

Cuando se produce un melisma sobre una sílaba distinta de la última de una palabra, dicha sílaba se suele unir a la siguiente con un guión separador. Esto se indica en el código de entrada escribiendo un doble guión, --, inmediatamente después de la sílaba.

De forma alternativa, cuando se produce un melisma sobre la última o sobre la única sílaba de una palabra, se suele trazar una línea extensora desde el final de la sílaba hasta la última nota del melisma. Esto se indica en el código de entrada escribiendo una doble barra baja o carácter de subrayado, \_\_, inmediatamente después de la palabra.

Existen cinco formas de indicar los melismas:

- Se crean melismas automáticamente sobre las notas unidas mediante ligadura de unión:

```

<<
\new Voice = "melody" \relative {
  \time 3/4
  f' '4 g2 ~ |
  4 e2 ~ |
  8
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- e __
}
>>

```



- Se pueden crear melismas automáticamente a partir de la música escribiendo ligaduras de expresión sobre las notas de cada melisma. Ésta es la forma usual de escribir la letra:

```

<<
\new Voice = "melody" \relative {

```

```

\time 3/4
f''4 g8 ( f e f )
e8 ( d e2 )
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- e --
}
>>

```



Observe que las ligaduras de fraseo no afectan a la creación de melismas.

- Se considera que las notas forman un melisma cuando se unen manualmente mediante barra, siempre y cuando el barrado automático esté desactivado. Véase [Establecer el comportamiento de las barras automáticas], página 91.

```

<<
\new Voice = "melody" \relative {
  \time 3/4
  \autoBeamOff
  f''4 g8[ f e f]
  e2.
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- e
}
>>

```



Claramente, esto no es adecuado para los melismas formados por notas de duración más larga que la corchea.

- Un grupo de notas sin ligadura de expresión se trata como un melisma si están comprendidas entre `\melisma` y `\melismaEnd`.

```

<<
\new Voice = "melody" \relative {
  \time 3/4
  f''4 g8
  \melisma
  f e f
  \melismaEnd
  e2.
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- e
}
>>

```

&gt;&gt;



- Se puede definir un melisma enteramente dentro de la letra escribiendo un carácter de barra baja suelto, `_`, por cada nota adicional que se quiere añadir al melisma.

&lt;&lt;

```
\new Voice = "melody" \relative {
  \time 3/4
  f''4 g8 f e f
  e8 d e2
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- _ _ _ e _ _ _
}
```

&gt;&gt;

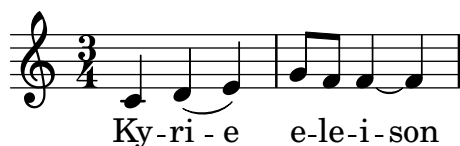


Es posible tener ligaduras de unión o de expresión y barras manuales en la melodía sin que indiquen melisma. Para hacerlo, ajuste el valor de `melismaBusyProperties`:

&lt;&lt;

```
\new Voice = "melody" \relative {
  \time 3/4
  \set melismaBusyProperties = #'()
  c'4 d ( e )
  g8 [ f ] f4 ~ 4
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- e e -- le -- i -- son
}
```

&gt;&gt;



Se pueden usar otros valores para `melismaBusyProperties` si queremos incluir o excluir selectivamente las ligaduras de unión, ligaduras de expresión o barras de la detección automática de los melismas; véase `melismaBusyProperties` en el apartado Sección “Tunable context properties” en *Referencia de Funcionamiento Interno*.

Como alternativa, si todas las indicaciones de melismas se van a ignorar, se puede establecer `ignoreMelismata` al valor verdadero; véase [Versos con ritmos distintos], página 321.

Si se requiere un melisma en el transcurso de un pasaje en el que `melismaBusyProperties` está activo, puede indicarse colocando una sola barra baja en la letra por cada nota que se debe incluir dentro del melisma:

&lt;&lt;

```

\new Voice = "melody" \relative {
  \time 3/4
  \set melismaBusyProperties = #'()
  c'4 d ( e )
  g8 [ f ] ~ 4 ~ f
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- _ e _ _ _ _
}
>>

```



## Instrucciones predefinidas

\autoBeamOff, \autoBeamOn, \melisma, \melismaEnd.

## Véase también

Glosario musical: Sección “melisma” en *Glosario Musical*.

Manual de aprendizaje: Sección “Alineación de la letra a una melodía” en *Manual de Aprendizaje*.

Referencia de la notación: [Alineación de la letra a una melodía], página 288, [Duración automática de las sílabas], página 290, [Establecer el comportamiento de las barras automáticas], página 91, [Versos con ritmos distintos], página 321.

Referencia de funcionamiento interno: Sección “Tunable context properties” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

No se crean automáticamente líneas extensoras bajo los melismas; se deben insertar manualmente con un doble guión bajo.

## Líneas de extensión y guiones

Los melismas se indican mediante una línea horizontal centrada entre una sílaba y la siguiente. Esta línea recibe el nombre de línea de extensión, y se escribe como ‘ \_ \_ ’ (fíjese en los espacios antes y después de los dos guiones bajos).

**Nota:** Los melismas se indican en la partitura con líneas extensoras, que se escriben con un doble guión bajo; pero también se pueden introducir melismas cortos mediante saltos de notas individuales, que se escriben como caracteres de guión bajo sueltos; de forma predeterminada éstos no producen la impresión de una línea de extensión.

Los guiones centrados se escriben como ‘ -- ’ entre sílabas de una misma palabra (fíjese en los espacios antes y después de los dos guiones). El guión quedará centrado entre las sílabas, y su longitud se ajustará en función del espacio que exista entre ellas.

En la música grabada de modo muy apretado, se pueden quitar los guiones. Se puede controlar si esto ocurre o no, mediante las propiedades `minimum-distance` (distancia mínima entre las

dos sílabas) y la `minimum-length` (umbral por debajo del cual se suprimen los guiones) de `LyricHyphen`.

De manera predeterminada no se repiten los guiones después de un salto de sistema cuando la línea siguiente comienza con una sílaba distinta. Si se establece la propiedad `after-line-breaking` al valor `#t`, se pueden imprimir los guiones en estas situaciones.

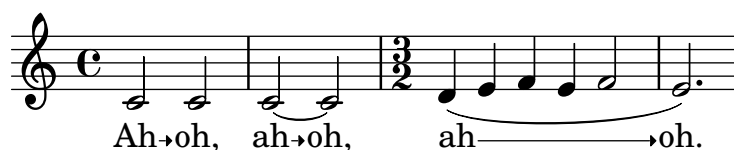
## Véase también

Referencia de funcionamiento interno: Sección “`LyricExtender`” en *Referencia de Funcionamiento Interno*, Sección “`LyricHyphen`” en *Referencia de Funcionamiento Interno*.

## Cambios de vocal graduales

Las transiciones de vocal (cambios graduales de una vocal o de una consonante mantenida), que pueden venir indicadas por flechas entre dos sílabas, se introducen con la instrucción `\vowelTransition` (véase Gould págs. 452–453). La flecha muestra la longitud de la transición, y por defecto siempre se imprime (se añade espacio en caso necesario si la música está muy apretada). Se indica que no hay ninguna articulación, a pesar del cambio de vocal, mediante ligaduras de unión o de expresión. La longitud mínima de las flechas puede ajustarse con la propiedad `minimum-length` del objeto `VowelTransition`.

```
{
  c'2 c'
  \set melismaBusyProperties = #'()
  c'2 ~ c'
  \time 3/2
  d'4( e' f' e' f'2
  e'2.)
}
\addlyrics
{
  Ah \vowelTransition oh,
  ah \vowelTransition oh,
  ah \vowelTransition _ _ _
  oh.
}
```



## Véase también

Glosario musical: Sección “vowel transition” en *Glosario Musical*.

Referencia de funcionamiento interno: Sección “`VowelTransition`” en *Referencia de Funcionamiento Interno*.

### 2.1.2 Técnicas específicas para la letra

#### Trabajar con letra y variables

Se pueden crear variables que contienen letra, pero la letra se debe introducir en el modo de letra:

```
musicOne = \relative {
```



```

c''4 b8. a16 g4. f8 e4 d c2
}
verseOne = \lyricmode {
  Joy to the world, the Lord is come.
}
\score {
  <<
    \new Voice = "one" {
      \time 2/4
      \musicOne
    }
    \new Lyrics \lyricsto "one" {
      \verseOne
    }
  >>
}

```



No es necesario escribir las duraciones si la variables se va a invocar con `\addlyrics` o con `\lyricsto`.

Para disposiciones distintas o más complejas, la mejor forma es definir en primer lugar las variables de la música y de la letra, después preparar una jerarquía de pentagramas y letras, omitiendo la letra en sí, y después poner la letra utilizando `\context` por debajo. Esto asegura que las voces referenciadas por `\lyricsto` siempre han sido definidas previamente. Por ejemplo:

```

sopranoMusic = \relative { c''4 c c c }
contraltoMusic = \relative { a'4 a a a }
sopranoWords = \lyricmode { Sop -- ra -- no words }
contraltoWords = \lyricmode { Con -- tral -- to words }

\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice = "sopranos" {
        \sopranoMusic
      }
    }
    \new Lyrics = "sopranos"
    \new Lyrics = "contraltos"
    \new Staff {
      \new Voice = "contraltos" {
        \contraltoMusic
      }
    }
  >>
  \context Lyrics = "sopranos" {
    \lyricsto "sopranos" {
      \sopranoWords
    }
  }
}

```

```

\context Lyrics = "contraltos" {
  \lyricsto "contraltos" {
    \contraltoWords
  }
}
>>
}

```



## Véase también

Referencia de la notación: [Posicionamiento vertical de la letra], página 301.

Referencia de funcionamiento interno: Sección “LyricCombineMusic” en *Referencia de Funcionamiento Interno*, Sección “Lyrics” en *Referencia de Funcionamiento Interno*.

## Posicionamiento vertical de la letra

Dependiendo del tipo de música, la letra puede colocarse sobre el pentagrama, debajo del pentagrama, o entre dos pentagramas. Lo más fácil es situar la letra debajo de la pauta asociada, y se puede conseguir simplemente definiendo el contexto Lyrics por debajo del contexto Staff:

```

\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative { c''4 c c c }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here are the words
      }
    }
  >>
}

```



Se puede colocar la letra sobre el pentagrama usando uno de los dos métodos disponibles: el más sencillo (y preferible) es usar la misma sintaxis que se ha visto más arriba y especificar explícitamente la posición de la letra:

```

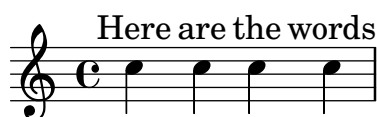
\score {
  <<
    \new Staff = "staff" {

```

```

    \new Voice = "melody" {
      \relative { c''4 c c c }
    }
  }
  \new Lyrics \with { alignAboveContext = "staff" } {
    \lyricsto "melody" {
      Here are the words
    }
  }
>>
}

```

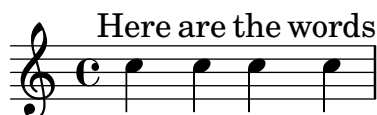


Como alternativa, se puede usar un proceso en dos pasos. En primer lugar se declara el contexto Lyrics (sin ningún contenido) antes de los contextos Staff y Voice, y después se escribe la instrucción `\lyricsto` después de la declaración de Voice a que hace referencia, usando `\context`, como se ve a continuación:

```

\score {
  <<
    \new Lyrics = "lyrics" \with {
      % lyrics above a staff should have this override
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff {
      \new Voice = "melody" {
        \relative { c''4 c c c }
      }
    }
    \context Lyrics = "lyrics" {
      \lyricsto "melody" {
        Here are the words
      }
    }
  }
  >>
}

```



Cuando hay dos voces sobre pentagramas separados, la letra se puede colocar entre ellos usando cualquiera de estos métodos. He aquí un ejemplo del segundo método:

```

\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice = "sopranos" {
        \relative { c''4 c c c }
      }
    }
  }
}

```

```

\new Lyrics = "sopranos"
\new Lyrics = "contraltos" \with {
  % lyrics above a staff should have this override
  \override VerticalAxisGroup.staff-affinity = #DOWN
}
\new Staff {
  \new Voice = "contraltos" {
    \relative { a'4 a a a }
  }
}
\context Lyrics = "sopranos" {
  \lyricsto "sopranos" {
    Sop -- ra -- no words
  }
}
\context Lyrics = "contraltos" {
  \lyricsto "contraltos" {
    Con -- tral -- to words
  }
}
>>
}

```



Es posible generar otras combinaciones de letra y pautas mediante la elaboración de los ejemplos anteriores, o examinando las plantillas del Manual de Aprendizaje, véase Sección “Plantillas de conjuntos vocales” en *Manual de Aprendizaje*.

## Fragmentos de código seleccionados

### *Conseguir el espaciado de la letra de la versión 2.12*

en versiones más recientes

El motor de espaciado vertical cambió en la versión 2.14. Esto puede hacer que se altere el espaciado de la letra de las canciones. Es posible fijar propiedades para los contextos **Lyric** y **Staff** de forma que el motor de espaciado se comporte como lo hacía en la versión 2.12.

```

global = {
  \key d \major
  \time 3/4
}

sopMusic = \relative c' {
  % VERSE ONE
  fis4 fis fis | \break
  fis4. e8 e4
}

```

```

}

altoMusic = \relative c' {
  % VERSE ONE
  d4 d d |
  d4. b8 b4 |
}

tenorMusic = \relative c' {
  a4 a a |
  b4. g8 g4 |
}

bassMusic = \relative c {
  d4 d d |
  g,4. g8 g4 |
}

words = \lyricmode {
  Great is Thy faith -- ful -- ness,
}

\score {
  \new ChoirStaff <<
    \new Lyrics = sopranos
    \new Staff = women <<
      \new Voice = "sopranos" {
        \voiceOne
        \global \sopMusic
      }
      \new Voice = "altos" {
        \voiceTwo
        \global \altoMusic
      }
    >>
    \new Lyrics = "altos"
    \new Lyrics = "tenors"
    \new Staff = men <<
      \clef bass
      \new Voice = "tenors" {
        \voiceOne
        \global \tenorMusic
      }
      \new Voice = "basses" {
        \voiceTwo \global \bassMusic
      }
    >>
    \new Lyrics = basses
    \context Lyrics = sopranos \lyricsto sopranos \words
    \context Lyrics = altos \lyricsto altos \words
    \context Lyrics = tenors \lyricsto tenors \words
    \context Lyrics = basses \lyricsto basses \words
  }
}

```

```

>>
\layout {
  \context {
    \Lyrics
    \override VerticalAxisGroup.staff-affinity = ##f
    \override VerticalAxisGroup.staff-staff-spacing =
      #'((basic-distance . 0)
        (minimum-distance . 2)
        (padding . 2))
  }
  \context {
    \Staff
    \override VerticalAxisGroup.staff-staff-spacing =
      #'((basic-distance . 0)
        (minimum-distance . 2)
        (padding . 2))
  }
}

```

Great is Thy

Great is Thy

Great is Thy

faith - - - ful - ness,

faith - - - ful - ness,

faith - - - ful - ness,

faith - - - ful - ness,

### Véase también

Manual de aprendizaje: Sección “Plantillas de conjuntos vocales” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.1.7 [Orden de disposición de los contextos], página 640, Sección 5.1.2 [Crear y referenciar contextos], página 624.

## Colocación horizontal de las sílabas

Para aumentar la separación entre las líneas de texto, establezca la propiedad `minimum-distance` de `LyricSpace`.

```
\relative c' {
  c c c c
  \override Lyrics.LyricSpace.minimum-distance = #1.0
  c c c c
}
\addlyrics {
  longtext longtext longtext longtext
  longtext longtext longtext longtext
}
```



Para efectuar este cambio en todas las letras de la partitura, ajuste la propiedad dentro del bloque `\layout`.

```
\score {
  \relative {
    c' c c c
    c c c c
  }
  \addlyrics {
    longtext longtext longtext longtext
    longtext longtext longtext longtext
  }
  \layout {
    \context {
      \Lyrics
      \override LyricSpace.minimum-distance = #1.0
    }
  }
}
```



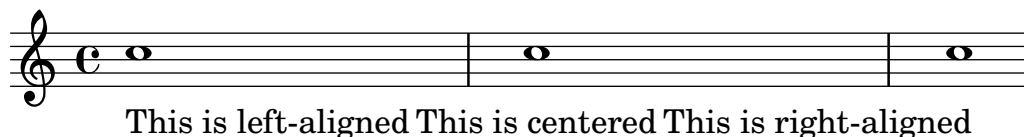


## Fragmentos de código seleccionados

### *Alineación de la letra*

La alineación horizontal de la letra se puede ajustar sobreescribiendo la propiedad `self-alignment-X` del objeto `LyricText`. `#-1` es izquierda, `#0` es centrado y `#1` es derecha; sin embargo, podemos usar también `#LEFT`, `#CENTER` y `#RIGHT`.

```
\layout { ragged-right = ##f }
\relative c'' {
  c1
  c1
  c1
}
\addlyrics {
  \once \override LyricText.self-alignment-X = #LEFT
  "This is left-aligned"
  \once \override LyricText.self-alignment-X = #CENTER
  "This is centered"
  \once \override LyricText.self-alignment-X = #1
  "This is right-aligned"
}
```



## Advertencias y problemas conocidos

La comprobación necesaria para asegurar que las inscripciones de texto están dentro de los márgenes, requiere cálculos adicionales. Para acelerar ligeramente el procesado, se puede desactivar esta funcionalidad:

```
\override Score.PaperColumn.keep-inside-line = ##f
```

Para hacer que el texto evite también las líneas divisorias, use

```
\layout {
  \context {
    \Lyrics
    \consists "Bar_engraver"
    \consists "Separating_line_group_engraver"
    \hide BarLine
  }
}
```

## Letra y repeticiones

### Repeticiones sencillas

Las repeticiones *en la música* se describen extensamente en alguna otra parte de este manual; véase Sección 1.4 [Repeticiones], página 158. Esta sección explica cómo añadir letra a secciones de música que se repiten.



La letra que va con una sección de música que se repite, debiera estar encerrada en la misma construcción de repeticiones exacta que la música, si el texto no cambia.

```
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative {
          a'4 a a a
          \repeat volta 2 { b4 b b b }
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Not re -- peat -- ed.
        \repeat volta 2 { Re -- peat -- ed twice. }
      }
    }
  >>
}
```



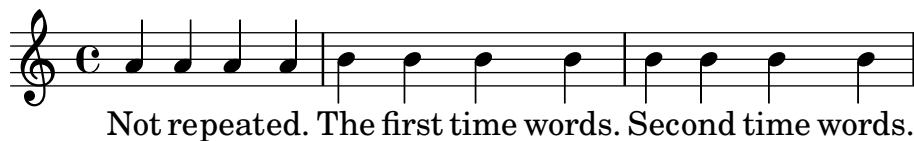
El texto se expandirá correctamente de esta manera si las repeticiones se despliegan.

```
\score {
  \unfoldRepeats {
    <<
      \new Staff {
        \new Voice = "melody" {
          \relative {
            a'4 a a a
            \repeat volta 2 { b4 b b b }
          }
        }
      }
      \new Lyrics {
        \lyricsto "melody" {
          Not re -- peat -- ed.
          \repeat volta 2 { Re -- peat -- ed twice. }
        }
      }
    >>
  }
}
```



Si la sección repetida ha de desplegarse y tiene distinta letra, escriba simplemente todo el texto:

```
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative {
          a'4 a a a
          \repeat unfold 2 { b4 b b b }
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Not re -- peat -- ed.
        The first time words.
        Sec -- ond time words.
      }
    }
  >>
}
```



Cuando el texto de las secciones de primera o segunda vez es distinto, la letra de cada repetición se debe escribir dentro de contextos `Lyrics` diferentes, anidados correctamente en secciones paralelas:

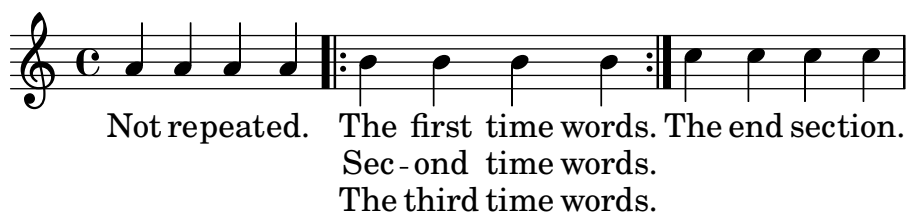
```
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative {
          a'4 a a a
          \repeat volta 2 { b4 b b b }
        }
      }
    }
    \new Lyrics \lyricsto "melody" {
      Not re -- peat -- ed.
      <<
        { The first time words. }
        \new Lyrics {
          \set associatedVoice = "melody"
          Sec -- ond time words.
        }
      >>
    }
  >>
}
```

}



Se pueden añadir más estrofas de una manera similar:

```
\score {
  <<
    \new Staff {
      \new Voice = "singleVoice" {
        \relative {
          a'4 a a a
          \repeat volta 3 { b4 b b b }
          c4 c c c
        }
      }
    }
  \new Lyrics \lyricsto "singleVoice" {
    Not re -- peat -- ed.
    <<
      { The first time words. }
      \new Lyrics {
        \set associatedVoice = "singleVoice"
        Sec -- ond time words.
      }
      \new Lyrics {
        \set associatedVoice = "singleVoice"
        The third time words.
      }
    >>
    The end sec -- tion.
  }
  >>
}
```



Sin embargo, si esta construcción se inserta dentro de un contexto de varios pentagramas tal como un `ChoirStaff`, la letra de los versos segundo y tercero aparecerán debajo del pentagrama inferior.

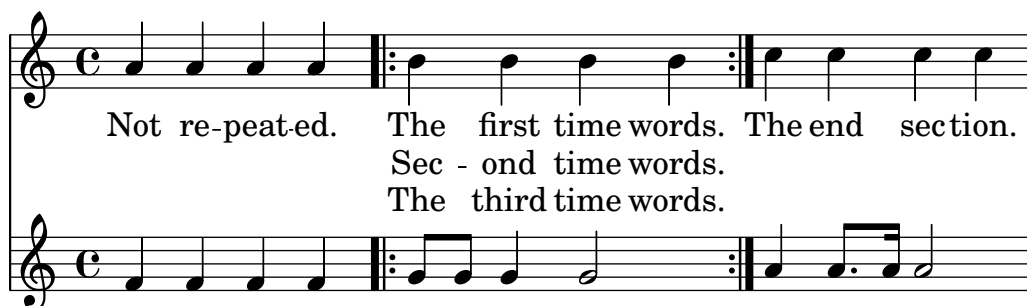
Para posicionarlos correctamente utilice `alignBelowContext`:

```
\score {
  <<
    \new Staff {
```

```

\new Voice = "melody" {
  \relative {
    a'4 a a a
    \repeat volta 3 { b4 b b b }
    c4 c c c
  }
}
}
\new Lyrics = "firstVerse" \lyricsto "melody" {
  Not re -- peat -- ed.
  <<
    { The first time words. }
    \new Lyrics = "secondVerse"
    \with { alignBelowContext = "firstVerse" } {
      \set associatedVoice = "melody"
      Sec -- ond time words.
    }
    \new Lyrics = "thirdVerse"
    \with { alignBelowContext = "secondVerse" } {
      \set associatedVoice = "melody"
      The third time words.
    }
  >>
  The end sec -- tion.
}
\new Voice = "harmony" {
  \relative {
    f'4 f f f \repeat volta 2 { g8 g g4 g2 } a4 a8. a16 a2
  }
}
}
>>
}

```



## Repeticiones con finales alternativos (primera y segunda vez)

Si el texto de la sección repetida es el mismo, y ninguno de los bloques `\alternative` comienza con un silencio, se puede usar exactamente la misma estructura tanto para la letra como para la música. Esto tiene la ventaja de que `\unfoldRepeats` expande correctamente tanto la música como la letra.

```

\score {
  <<
    \new Staff {
      \time 2/4

```

```

\new Voice = "melody" {
  \relative {
    a'4 a a a
    \repeat volta 2 { b4 b }
    \alternative { { b b } { b c } }
  }
}
\new Lyrics {
  \lyricsto "melody" {
    Not re -- peat -- ed.
    \repeat volta 2 { Re -- peat -- }
    \alternative { { ed twice. } { ed twice. } }
  }
}
>>
}

```



Pero cuando la sección repetida tiene un texto distinto, o cuando uno de los bloques `\alternative` comienza con un silencio, no se puede usar una construcción de repetición para el texto y es necesario insertar manualmente instrucciones `\skip` para saltar sobre las notas en las secciones alternativas que no correspondan.

Nota: no utilice un guión bajo, `_`, para saltar notas: el guión bajo indica un melisma, y hace que la sílaba anterior resulte alineada por la izquierda.

**Nota:** La instrucción `\skip` debe ir seguida por un número, pero este número se ignora dentro de la letra, que deriva las duraciones de las notas de una melodía asociada a través de `\addlyrics` o `\lyricsto`. Cada `\skip` salta una sola nota de cualquier valor, independientemente del valor del número que se escribe a continuación.

```

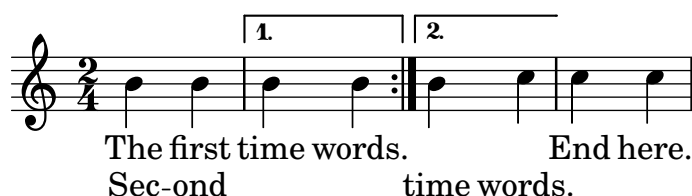
\score {
  <<
  \new Staff {
    \time 2/4
    \new Voice = "melody" {
      \relative {
        \repeat volta 2 { b'4 b }
        \alternative { { b b } { b c } }
        c4 c
      }
    }
  }
  \new Lyrics {
    \lyricsto "melody" {
      The first time words.
    }
  }
}

```

```

        \repeat unfold 2 { \skip 1 }
        End here.
    }
}
\new Lyrics {
    \lyricsto "melody" {
        Sec -- ond
        \repeat unfold 2 { \skip 1 }
        time words.
    }
}
>>
}

```



Cuando una nota está ligada sobre dos o más finales alternativos (sobre la primera nota de una casilla de primera y segunda vez), se usa una ligadura de unión para arrastrar la nota dentro del primer final alternativo y se usa una instrucción `\repeatTie` de ligadura de repetición en las casillas de repetición siguientes. Esta estructura produce problemas de alineación difíciles cuando hay letra de por medio; aumentar la longitud de las secciones alternativas de forma que las notas ligadas se vean contenidas completamente dentro de ellas, puede ofrecer un resultado más aceptable.

La ligadura de unión crea un melisma dentro de la casilla de primera vez pero no en la segunda vez y siguientes; por tanto, para alinear la letra correctamente es necesario inhabilitar la creación automática de melismas sobre la sección de finales alternativos e insertar saltos manuales.

```

\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative {
          \set melismaBusyProperties = #'()
          \repeat volta 2 { b'4 b ~}
          \alternative { { b b } { b \repeatTie c } }
          \unset melismaBusyProperties
          c4 c
        }
      }
    }
  }
  \new Lyrics {
    \lyricsto "melody" {
      \repeat volta 2 { Here's a __ }
      \alternative {
        { \skip 1 verse }
        { \skip 1 sec }
      }
    }
  }
}

```

```

    ond one.
  }
}
>>
}

```

Here's a\_\_verse      sec ond one.

Observe que si se usa `\unfoldRepeats` sobre una sección que contiene alguna `\repeatTie`, la `\repeatTie` se debe eliminar para evitar que se impriman al mismo tiempo los dos tipos de ligadura.

Cuando la sección repetida tiene textos distintos, no se puede usar `\repeat` para la letra y se deben insertar instrucciones `\skip` manualmente, como se vio antes.

```

\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative {
          \repeat volta 2 { b'4 b ~}
          \alternative { { b b } { b \repeatTie c } }
          c4 c
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here's a __ verse.
        \repeat unfold 2 { \skip 1 }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here's one
        \repeat unfold 2 { \skip 1 }
        more to sing.
      }
    }
  >>
}

```

Here's a\_\_verse.  
Here's one      more to sing.

Si queremos imprimir líneas extensoras y guiones que entran o salen de las casillas de primera y segunda vez, hemos de insertarlos manualmente.

```

\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative {
          \repeat volta 2 { b'4 b ~}
          \alternative { { b b } { b \repeatTie c } }
          c4 c
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here's a __ verse.
        \repeat unfold 2 { \skip 1 }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here's "a_"
        \skip 1
        "_" sec -- ond one.
      }
    }
  >>
}

```



## Véase también

Referencia de la notación: Sección 5.1.3 [Mantener vivos los contextos], página 628, Sección 1.4 [Repeticiones], página 158.

## Letras en divisi

Cuando solamente difieren las palabras y las duraciones de las dos partes mientras que la altura de las notas es la misma, el método adecuado puede ser la desactivación temporal de la detección automática de los melismas e indicar el melisma dentro de la letra:

```

\score {
  <<
    \new Voice = "melody" {
      \relative c' {
        \set melismaBusyProperties = #'()
        \slurDown
        \slurDashed
        e4 e8 ( e ) c4 c |
        \unset melismaBusyProperties
      }
    }
  >>
}

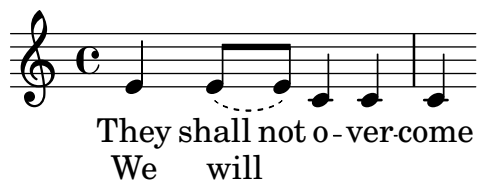
```



```

      c
    }
  }
  \new Lyrics \lyricsto "melody" {
    They shall not o -- ver -- come
  }
  \new Lyrics \lyricsto "melody" {
    We will _
  }
  >>
}

```



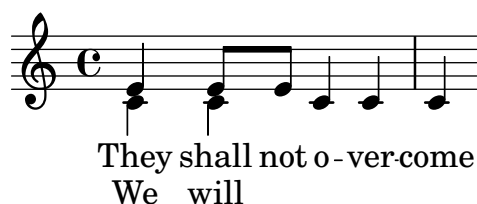
Cuando difieren tanto la música como las palabras, puede ser mejor imprimir la música y la letra que difieren mediante el nombrado de contextos de voz y adjuntando la letra a estos contextos específicos:

```

\score {
  <<
    \new Voice = "melody" {
      \relative {
        <<
          {
            \voiceOne
            e'4 e8 e
          }
          \new Voice = "splitpart" {
            \voiceTwo
            c4 c
          }
        >>
        \oneVoice
        c4 c |
      }
    }
  }
  \new Lyrics \lyricsto "melody" {
    They shall not o -- ver -- come
  }
  \new Lyrics \lyricsto "splitpart" {
    We will
  }
  >>
}

```

}



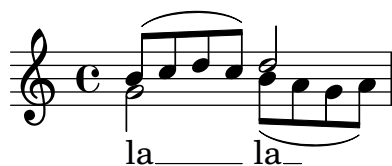
En la música coral es frecuente que una parte vocal se divida en dos durante varios compases. La construcción `<< {...} \ \ {...} >>`, en la que dos (o más) expresiones musicales están separadas por dos barras invertidas, puede en principio parecer la forma más adecuada de realizar las voces divididas. Sin embargo, esta construcción asigna **todas** las expresiones que están dentro de ella a **contextos de voz NUEVOS** lo que hace que no se asigne a ellas *ninguna letra* porque la letra se asigna al contexto de voz original, que no es lo que pretendemos, en general. La construcción correcta que debemos usar es el pasaje polifónico temporal, véase la sección *Pasajes polifónicos temporales* dentro de [Polifonía en un solo pentagrama], página 181.

## Polifonía con letras compartidas

Cuando dos voces cuyas notas tienen distintas duraciones comparten la misma letra, la alineación de ésta a una de las voces puede dar lugar a problemas en la otra voz. Por ejemplo, la segunda línea extensora que se ve debajo es demasiado corta, porque la letra está alineada solamente a la voz superior:

```
soprano = \relative { b'8( c d c) d2 }
alto = \relative { g'2 b8( a g a) }
words = \lyricmode { la __ la __ }

\new Staff <<
  \new Voice = "sopranoVoice" { \voiceOne \soprano }
  \new Voice { \voiceTwo \alto }
  \new Lyrics \lyricsto "sopranoVoice" \words
>>
```



Para obtener el resultado deseado, alinee la letra con un contexto `NullVoice` ('Voz Nula') nuevo que contenga una combinación adecuada de las dos voces. Las notas del contexto `NullVoice` no aparecen en la página impresa, pero pueden utilizarse para alinear la letra adecuadamente:

```
soprano = \relative { b'8( c d c) d2 }
alto = \relative { g'2 b8( a g a) }
aligner = \relative { b'8( c d c) b( a g a) }
words = \lyricmode { la __ la __ }

\new Staff <<
  \new Voice { \voiceOne \soprano }
  \new Voice { \voiceTwo \alto }
  \new NullVoice = "aligner" \aligner
  \new Lyrics \lyricsto "aligner" \words
>>
```

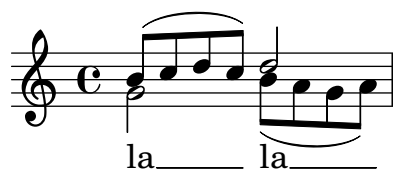
&gt;&gt;



Este método se puede usar también con la función `\partCombine`, que en sí no permite incluir letra:

```
soprano = \relative { b'8( c d c) d2 }
alto = \relative { g'2 b8( a g a) }
aligner = \relative { b'8( c d c) b( a g a) }
words = \lyricmode { la __ la __ }

\new Staff <<
  \new Voice \partCombine \soprano \alto
  \new NullVoice = "aligner" \aligner
  \new Lyrics \lyricsto "aligner" \words
>>
```



## Advertencias y problemas conocidos

La función `\addlyrics` solo funciona con letras de `Voice`, por lo que no puede utilizarse con `NullVoice`.

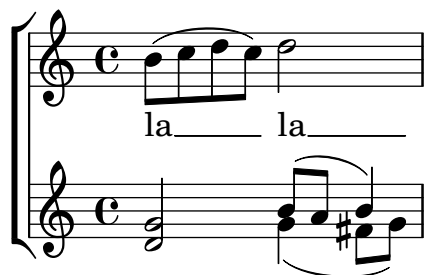
La función `\partCombine` se describe en [Combinación automática de las partes], página 192.

Por último, este método se puede usar incluso si las voces están en distintos pentagramas, y no se limita a dos voces únicamente:

```
soprano = \relative { b'8( c d c) d2 }
altoOne = \relative { g'2 b8( a b4) }
altoTwo = \relative { d'2 g4( fis8 g) }
aligner = \relative { b'8( c d c) d( d d d) }
words = \lyricmode { la __ la __ }

\new ChoirStaff \with {\accepts NullVoice } <<
  \new Staff \soprano
  \new NullVoice = "aligner" \aligner
  \new Lyrics \lyricsto "aligner" \words
  \new Staff \partCombine \altoOne \altoTwo
```

&gt;&gt;

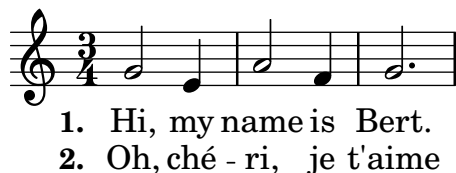


### 2.1.3 Versos

#### Añadir números de verso

Los números de los versos se pueden añadir estableciendo `stanza`, p.ej.,

```
\new Voice \relative {
  \time 3/4 g'2 e4 a2 f4 g2.
} \addlyrics {
  \set stanza = "1. "
  Hi, my name is Bert.
} \addlyrics {
  \set stanza = "2. "
  Oh, ché -- ri, je t'aime
}
```



Estos números aparecerán inmediatamente antes de la primera sílaba. También pueden agruparse dos líneas de estrofa, por ejemplo en el caso de una repetición con letras diferentes:

```
leftbrace = \markup {
  \override #'(font-encoding . fetaBraces)
  \lookup "brace80"
}
```

```
stanzaOneOne = \lyricmode {
  \set stanza = \markup {
    \column { \vspace #.33 "1. "}
    \leftbrace
  }
  Child, you're mine and I love you.
  Lend thine ear to what I say.
}
```

```
stanzaOneThree = \lyricmode {
  Child, I have no great -- er joy
  Than to have you walk in truth.
}
```

```

\new Voice {
  \repeat volta 2 {
    c'8 c' c' c' c' c' c'4
    c'8 c' c' c' c' c' c'4
  }
}
\addlyrics { \stanzaOneOne }
\addlyrics { \stanzaOneThree }

```



1. { Child, you're mine and I love you.  
 Child, I have no greater joy



Lend thine ear to what I say.  
 Than to have you walk in truth.

## Añadir expresiones dinámicas a los versos

Los versos que difieren en su sonoridad se pueden especificar escribiendo una indicación dinámica antes de cada verso. En LilyPond, todo lo que aparece delante de un verso está dentro del objeto `StanzaNumber`; las indicaciones dinámicas no son diferentes. Por razones técnicas, tendrá que establecer el valor de la sección stanza (verso) fuera de `\lyricmode`:

```

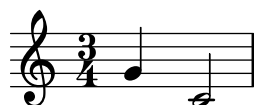
text = {
  \set stanza = \markup { \dynamic "ff" "1. " }
  \lyricmode {
    Big bang
  }
}

```

```

<<
  \new Voice = "tune" {
    \time 3/4
    g'4 c'2
  }
\new Lyrics \lyricsto "tune" \text
>>

```



**ff**1. Big bang

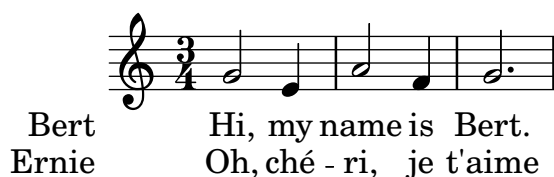
## Añadir el nombre de los cantantes a los versos

También se pueden poner los nombres de los cantantes. Se imprimen al comienzo de la línea, igual que los nombres de instrumento. Se crean estableciendo un valor para `vocalName`. Se puede definir una versión abreviada como `shortVocalName`.

```

\new Voice \relative {
  \time 3/4 g'2 e4 a2 f4 g2.
} \addlyrics {
  \set vocalName = "Bert "
  Hi, my name is Bert.
} \addlyrics {
  \set vocalName = "Ernie "
  Oh, ché -- ri, je t'aime
}

```



## Versos con ritmos distintos

Con frecuencia, los distintos versos de una canción encajan de formas ligeramente diferentes con la misma melodía. Estas variaciones se pueden capturar con el uso de `\lyricsto`.

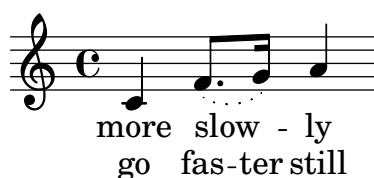
## Ignorar los melismas

Existe la posibilidad de que el texto tenga un melisma en un verso, pero varias sílabas en otro. Una solución es hacer que la voz más rápida ignore el melisma. Esto se consigue estableciendo `ignoreMelismata` en el contexto Lyrics.

```

<<
\relative \new Voice = "lahlah" {
  \set Staff.autoBeaming = ##f
  c'4
  \slurDotted
  f8.[( g16)]
  a4
}
\new Lyrics \lyricsto "lahlah" {
  more slow -- ly
}
\new Lyrics \lyricsto "lahlah" {
  go
  \set ignoreMelismata = ##t
  fas -- ter
  \unset ignoreMelismata
  still
}
>>

```



## Advertencias y problemas conocidos

A diferencia de casi todas las instrucciones `\set`, `\set ignoreMelismata` no funciona si va precedido de `\once`. Es necesario utilizar `\set` y `\unset` para delimitar la letra en que se quieren ignorar los melismas.

## Aplicar sílabas a notas de adorno

De forma predeterminada, las notas de adorno (p.ej. insertadas por medio de `\grace`) no pueden recibirla asignación de sílabas si se utiliza `\lyricsto`, pero este comportamiento puede cambiarse:

```
<<
\new Voice = melody \relative {
  f'4 \appoggiatura a32 b4
  \grace { f16 a16 } b2
  \afterGrace b2 { f16[ a16] }
  \appoggiatura a32 b4
  \acciaccatura a8 b4
}
\new Lyrics
\lyricsto melody {
  normal
  \set includeGraceNotes = ##t
  case,
  gra -- ce case,
  after -- grace case,
  \set ignoreMelismata = ##t
  app. case,
  acc. case.
}
>>
```



## Advertencias y problemas conocidos

Como ocurre con `associatedVoice`, `includeGraceNotes` se tiene que establecer al menos una sílaba antes de la que se va a colocar bajo una nota de adorno. En caso de una nota de adorno al comienzo mismo de una pieza, se recomienda usar un bloque `\with` o `\context`:

```
<<
\new Voice = melody \relative c' {
  \grace { c16( d e f }
  g1) f
}
\new Lyrics \with { includeGraceNotes = ##t }
\lyricsto melody {
  Ah -- fa
}
>>
```

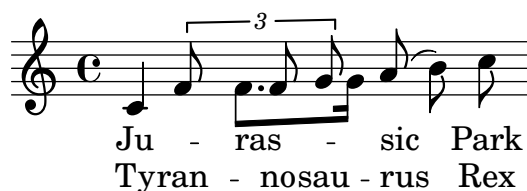
&gt;&gt;



## Cambio a una melodía alternativa

Son posibles variaciones más complejas en la coordinación de la letra y la música. La melodía a la que se está alineando la letra puede cambiarse desde dentro de la letra mediante el establecimiento de la propiedad `associatedVoice`:

```
<<
\relative \new Voice = "lahlah" {
  \set Staff.autoBeaming = ##f
  c'4
  <<
    \new Voice = "alternative" {
      \voiceOne
      \tuplet 3/2 {
        % show associations clearly.
        \override NoteColumn.force-hshift = #-3
        f8 f g
      }
    }
  {
    \voiceTwo
    f8.[ g16]
    \oneVoice
  } >>
  a8( b) c
}
\new Lyrics \lyricsto "lahlah" {
  Ju -- ras -- sic Park
}
\new Lyrics \lyricsto "lahlah" {
  % Tricky: need to set associatedVoice
  % one syllable too soon!
  \set associatedVoice = "alternative" % applies to "ran"
  Ty --
  ran --
  no --
  \set associatedVoice = "lahlah" % applies to "rus"
  sau -- rus Rex
} >>
```





El texto del primer verso se fija a la melodía llamada ‘lahlah’ de la forma usual, pero el segundo verso se fija inicialmente al contexto **lahlah** y después se cambia a la melodía **alternative** para las sílabas desde ‘ran’ hasta ‘sau’ mediante las líneas:

```
\set associatedVoice = "alternative" % se aplica a "ran"
Ty --
ran --
no --
\set associatedVoice = "lahlah" % se aplica a "rus"
sau -- rus Rex
```

Aquí, **alternative** es el nombre del contexto de **Voice** que contiene el tresillo.

Observe la posición de la instrucción `\set associatedVoice`: parece estar situada una sílaba antes de lo normal, pero está bien así.

**Nota:** La instrucción `\set associatedVoice` se debe escribir una sílaba *antes* de aquella en la que queremos que se produzca el cambio a la voz nueva. Dicho de otra forma, el cambio de la voz asociada se produce una sílaba más tarde de lo esperado. Esto es por razones técnicas, y no es un fallo.

## Imprimir los versos al final

En ocasiones es conveniente tener un verso ajustado a la música, y el resto añadido en forma de estrofa al final de la pieza. Esto se puede conseguir escribiendo los versos adicionales dentro de una sección `\markup` fuera del bloque **Score** principal de la partitura. Tenga en cuenta que existen dos formas distintas de forzar los saltos de línea al utilizar `\markup`.

```
melody = \relative {
  e' d c d | e e e e |
  d d e d | c1 |
}

text = \lyricmode {
  \set stanza = "1." Ma- ry had a lit- tle lamb,
  its fleece was white as snow.
}

\score {
  <<
    \new Voice = "one" { \melody }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
}

\markup {
  \column {
    \line { Verse 2. }
    \line { All the children laughed and played }
    \line { To see a lamb at school. }
  }
}

\markup {
  \wordwrap-string "
```

Verse 3.

Mary took it home again,

It was against the rule."

}



Verse 2.

All the children laughed and played

To see a lamb at school.

Verse 3.

Mary took it home again,

It was against the rule.

## Imprimir los versos al final en varias columnas

Cuando una pieza tiene muchos versos, a menudo se imprimen en varias columnas a lo largo de toda la página. Con frecuencia un número de verso fuera del margen precede a cada verso. El ejemplo siguiente muestra cómo producir dicha salida en LilyPond.

```
melody = \relative {
  c'4 c c c | d d d d
}

text = \lyricmode {
  \set stanza = "1." This is verse one.
  It has two lines.
}

\score {
  <<
    \new Voice = "one" { \melody }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
}

\markup {
  \fill-line {
    \hspace #0.1 % moves the column off the left margin;
    % can be removed if space on the page is tight
    \column {
      \line { \bold "2."
        \column {
          "This is verse two."
          "It has two lines."
        }
      }
    }
  }
}
```

```

    }
    \combine \null \vspace #0.1 % adds vertical spacing between verses
    \line { \bold "3."
      \column {
        "This is verse three."
        "It has two lines."
      }
    }
  }
}
\hspace #0.1 % adds horizontal spacing between columns;
\column {
  \line { \bold "4."
    \column {
      "This is verse four."
      "It has two lines."
    }
  }
}
\combine \null \vspace #0.1 % adds vertical spacing between verses
\line { \bold "5."
  \column {
    "This is verse five."
    "It has two lines."
  }
}
}
\hspace #0.1 % gives some extra space on the right margin;
% can be removed if page space is tight
}
}

```



1. This is verse one. It has two lines.

2. This is verse two.  
It has two lines.

3. This is verse three.  
It has two lines.

4. This is verse four.  
It has two lines.

5. This is verse five.  
It has two lines.

## Véase también

Referencia de funcionamiento interno: Sección “LyricText” en *Referencia de Funcionamiento Interno*, Sección “StanzaNumber” en *Referencia de Funcionamiento Interno*.

### 2.1.4 Canciones

#### Referencias para canciones

Por lo general, las canciones se escriben sobre tres pentagramas con la melodía del cantante en el pentagrama superior y dos pentagramas de acompañamiento de piano en la parte inferior. La letra del primer verso se imprime inmediatamente debajo del pentagrama superior. Si existe

solamente una pequeña cantidad de versos adicionales, pueden imprimirse inmediatamente debajo del primero, pero si hay más versos de los que caben cómodamente allí, el segundo verso y siguientes se imprimen después de la música, como texto independiente.

Todos los elementos notacionales necesarios para escribir canciones se describen completamente en otras secciones de la documentación:

- Para montar la disposición de las pautas, véase Sección 1.6.1 [Impresión de los pentagramas], página 200.
- Para escribir música para piano, véase Sección 2.2 [Teclados y otros instrumentos de varios pentagramas], página 352.
- Para imprimir la letra de una línea melódica, véase Sección 2.1.1 [Notación común para música vocal], página 286.
- Para colocar la letra, véase [Posicionamiento vertical de la letra], página 301.
- Para escribir versos véase Sección 2.1.3 [Versos], página 319.
- Las canciones se imprimen frecuentemente con los acordes indicados mediante cifrado americano (los nombres de los acordes en letras mayúsculas) sobre las pautas. Esto se describe en Sección 2.7.2 [Imprimir los acordes], página 448.
- Para imprimir diagramas de posiciones de los acordes para el acompañamiento de guitarra o para otros instrumentos con trastes en el mástil, consulte “Marcados de diagramas de posiciones de acorde” dentro de Sección 2.4.1 [Notación común para cuerdas con trastes], página 369.

## Véase también

Manual de aprendizaje: Sección “Canciones” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 2.1.1 [Notación común para música vocal], página 286, Sección 2.7.2 [Imprimir los acordes], página 448, Sección 1.6.1 [Impresión de los pentagramas], página 200, Sección 2.2 [Teclados y otros instrumentos de varios pentagramas], página 352, [Posicionamiento vertical de la letra], página 301, Sección 2.1.3 [Versos], página 319.

Fragmentos de código: Sección “Vocal music” en *Fragmentos de código*.

## Hojas guía de acordes

Se pueden imprimir hojas guía de acordes combinando partes vocales y el ‘modo de acordes’; esta sintaxis se explica en Sección 2.7 [Notación de acordes], página 442.

## Fragmentos de código seleccionados

### *Hoja guía de acordes o «lead sheet» sencilla*

Al juntar nombres de acorde en cifrado americano, melodía y letra, obtenemos una hoja guía de acordes o «lead sheet»:

```
<<
\chords { c2 g:sus4 f e }
\new Staff \relative c'' {
  a4 e c8 e r4
  b2 c4( d)
}
\addlyrics { One day this shall be free __ }
```

&gt;&gt;



## Véase también

Referencia de la notación: Sección 2.7 [Notación de acordes], página 442.

### 2.1.5 Música coral

Esta sección trata los asuntos de notación más directamente relacionados con la música coral. Esto incluye los anthems, las canciones por partes, los oratorios, etc.

## Referencias para música coral

La notación de música coral se realiza normalmente sobre dos, tres o cuatro pentagramas dentro de un grupo de pautas **ChoirStaff**. Si se necesita un acompañamiento, se sitúa debajo dentro de un grupo **PianoStaff**, que se suele reducir de tamaño, para ensayar las obras corales *a cappella*. Las notas de cada parte vocal se sitúan dentro de un contexto **Voice**, y cada pentagrama recibe o una sola parte vocal (es decir, una **Voice**) o un par de partes vocales (es decir, dos **Voices**).

Los textos se disponen en contextos **Lyrics**, bien debajo de cada pauta de música correspondiente, o bien una encima y una debajo de la pauta de música, si ésta contiene la música de las dos partes.

En otras partes del manual se describen completamente algunos otros temas sobre música coral:

- Hay una introducción a la creación de partituras vocales SATB en el Manual de aprendizaje, véase Sección “Partitura vocal a cuatro voces SATB” en *Manual de Aprendizaje*. También existe una plantilla incorporada que simplifica la introducción de música vocal SATB, véase see Sección “Plantillas incorporadas” en *Manual de Aprendizaje*.
- También en el Manual de aprendizaje hay varias plantillas adecuadas para diversos estilos de música coral, véase Sección “Plantillas de conjuntos vocales” en *Manual de Aprendizaje*.
- Para ver más información acerca de **ChoirStaff** y **PianoStaff**, consulte [Agrupar pentagramas], página 202.
- Las figuras con cabezas de formas, como las que se usan en la notación del estilo Arpa Sacra y otros similares, se describen en [Cabezas de notas con formas diversas], página 44.
- Cuando dos partes vocales comparten un pentagrama, las plicas, ligaduras, etc. de la parte aguda se orientan hacia arriba, y los de la parte grave hacia abajo. Para hacerlo, utilice `\voiceOne` y `\voiceTwo`. Véase [Polifonía en un solo pentagrama], página 181.
- Cuando una parte vocal se divide temporalmente, debemos usar *Pasajes polifónicos temporales* (véase [Polifonía en un solo pentagrama], página 181).

## Instrucciones predefinidas

`\oneVoice`, `\voiceOne`, `\voiceTwo`.

## Véase también

Manual de aprendizaje: Sección “Partitura vocal a cuatro voces SATB” en *Manual de Aprendizaje*, Sección “Plantillas de conjuntos vocales” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.1.7 [Orden de disposición de los contextos], página 640, [Agrupar pentagramas], página 202, [Cabezas de notas con formas diversas], página 44, [Polifonía en un solo pentagrama], página 181.

Fragmentos de código: Sección “Vocal music” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “ChoirStaff” en *Referencia de Funcionamiento Interno*, Sección “Lyrics” en *Referencia de Funcionamiento Interno*, Sección “PianoStaff” en *Referencia de Funcionamiento Interno*.

## Disposiciones de la partitura para música coral

La música coral que contiene cuatro pentagramas, con o sin acompañamiento de piano, se suele disponer con dos sistemas por página. Dependiendo del tamaño de la página, conseguirlo puede requerir cambios en ciertos ajustes predeterminados. Se deben tener en cuenta los siguientes ajustes:

- Se puede modificar el tamaño global de pentagrama para alterar el tamaño general de los elementos de la partitura. Véase Sección 4.2.2 [Establecer el tamaño del pentagrama], página 581.
- Las diferentes distancias entre sistemas, pentagramas y líneas de texto se pueden ajustar de manera independiente. Véase Sección 4.4 [Espaciado vertical], página 589.
- Se pueden imprimir las dimensiones de las variables de disposición verticales como ayuda para el ajuste del espaciado vertical. Esta y otras posibilidades para hacer encajar la música en un número de páginas menor, se describen en Sección 4.6 [Encajar la música en menos páginas], página 618.
- Si el número de sistemas por página cambia de uno a dos, es costumbre indicarlo mediante una marca separadora de sistemas entre los dos sistemas. Véase [Separación de sistemas], página 207.
- Para ver más detalles sobre otras propiedades de formateo de las páginas, consulte Sección 4.1 [Disposición de la página], página 567.

Las indicaciones dinámicas se colocan por omisión debajo del pentagrama, pero en la música coral se suelen situar encima del pentagrama para evitar las colisiones con el texto. La instrucción predefinida `\dynamicUp` hace esta tarea para las indicaciones dinámicas en un único contexto `Voice`. Si hay muchos contextos de voz, esta instrucción predefinida tendría que colocarse en cada uno de ellos. Como alternativa, se puede usar su forma expandida para poner las indicaciones dinámicas de toda la partitura encima de sus pautas respectivas, como se muestra aquí:

```
\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice {
        \relative { g'4\f g g g }
      }
    }
    \new Staff {
      \new Voice {
        \relative { d'4 d d\p d }
      }
    }
  >>
  \layout {
    \context {
      \Score
```

```

\override DynamicText.direction = #UP
\override DynamicLineSpanner.direction = #UP
}
}
}

```



## Instrucciones predefinidas

`\dynamicUp`, `\dynamicDown`, `\dynamicNeutral`.

## Véase también

Referencia de la notación: Sección 4.6.2 [Cambiar el espaciado], página 619, Sección 4.6.1 [Mostrar el espaciado], página 618, Sección 4.6 [Encajar la música en menos páginas], página 618, Sección 4.1 [Disposición de la página], página 567, Sección 4.2 [Disposición de la partitura], página 579, [Separación de sistemas], página 207, Sección 4.2.2 [Establecer el tamaño del pentagrama], página 581, Sección 4.3 [Saltos], página 582, Sección 4.4 [Espaciado vertical], página 589.

Referencia de funcionamiento interno: Sección “VerticalAxisGroup” en *Referencia de Funcionamiento Interno*, Sección “StaffGrouper” en *Referencia de Funcionamiento Interno*.

## Fragmentos de código seleccionados

### *Uso de arpeggioBracket para hacer más visible un divisi*

El corchete de arpeggios `arpeggioBracket` se puede usar para indicar la división de voces cuando no hay plicas que puedan ofrecer esta información. Se suele encontrar en la música coral.

```

\include "english.ly"

\score {
  \relative c'' {
    \key a \major
    \time 2/2
    <<
      \new Voice = "upper"
      <<
        { \voiceOne \arpeggioBracket
          a2( b2
            <b d>1\arpeggio)
            <cs e>\arpeggio ~
            <cs e>4
          }
        \addlyrics { \lyricmode { A -- men. } }
      >>
    \new Voice = "lower"
    { \voiceTwo
      a1 ~
    }
  }
}

```

```

a
a ~
a4 \bar "|."
}
>>
}
\layout { ragged-right = ##t }
}

```



## Véase también

Referencia de la notación: Sección 1.3.3 [Expresiones como líneas], página 147.

### 2.1.6 Ópera y musicales

La música, letra y diálogos de las óperas y musicales se disponen normalmente de una o más de las siguientes formas:

- Una *Partitura del director* que contiene todas las partes orquestales y vocales, junto a notas guía del libreto si existen pasajes hablados.
- *Particellas orquestales* que contienen la música de los instrumentos individuales de la orquesta o banda.
- Una *Partitura vocal* que contiene todas las partes vocales con acompañamiento de piano. El acompañamiento es normalmente una reducción de orquesta, y en este caso es frecuente indicar el nombre del instrumento original de la orquesta. Las partituras vocales a veces incluyen indicaciones de escena y notas guía del libreto.
- Un *Libro vocal* que contiene sólo las partes vocales (sin acompañamiento), a veces en combinación con el libreto.
- Un *Libreto* que contiene los diálogos completos que normalmente hay en los musicales, junto a la letra de las partes cantadas. Suelen incluirse también las indicaciones de escena. Se puede utilizar LilyPond para tipografiar libretos, pero dado que no contienen música, puede ser preferible algún método alternativo.

En la sección de referencias que aparece a continuación están relacionadas las secciones de la documentación de LilyPond que se ocupan de los temas necesarios para crear partituras en los estilos habituales de la ópera y los musicales. Después aparecen secciones que cubren las técnicas peculiares de la tipografía musical de las partituras de ópera y de musicales.

## Referencias para ópera y musicales

La gran mayoría de las nociones que aparecen a continuación se pueden aplicar a prácticamente cualquier música orquestal y de conjunto, además de a la música vocal y de escena:

- Una partitura de director contiene muchos pentagramas y legtras agrupados. En [Agrupar pentagramas], página 202, se explican maneras de agrupar pentagramas. Para anidar unos grupos dentro de otros, consulte [Grupos de pentagramas anidados], página 205.
- La impresión de pentagramas vacíos en las partituras del director y vocales se suele inhibir. Para crear estas partituras, conocidas como “Partituras francesas” consulte [Ocultar pentagramas], página 215.



- La escritura de partituras orquestales se estudia en Sección 1.6.3 [Escritura de las particellas], página 220. Pueden ser de relevancia otras secciones del capítulo de Notación especializada, dependiendo de la orquestación utilizada. Muchos instrumentos son transpositores, consulte [Transposición de los instrumentos], página 28.
- Si el número de sistemas por página varía de una a otra página, es costumbre separar los sistemas con una marca separadora de sistemas. Véase See [Separación de sistemas], página 207.
- Para ver detalles acerca de otras propiedades de formateo de las páginas, consulte Sección 4.1 [Disposición de la página], página 567.
- Se pueden insertar notas guía de diálogos, instrucciones de escena y notas al pie, véase Sección 3.2.4 [Crear notas al pie], página 526, y Sección 1.8 [Texto], página 254. También se pueden añadir indicaciones de escena extensas con una sección de elementos de marcado independientes entre dos bloques `\score`, véase [Texto separado], página 262.

## Véase también

Glosario musical: Sección “Frenched score” en *Glosario Musical*, Sección “Frenched staves” en *Glosario Musical*, Sección “transposing instrument” en *Glosario Musical*.

Referencia de la notación: Sección 3.2.4 [Crear notas al pie], página 526, [Agrupar pentagramas], página 202, [Ocultar pentagramas], página 215, [Transposición de los instrumentos], página 28, [Grupos de pentagramas anidados], página 205, Sección 4.1 [Disposición de la página], página 567, [Separación de sistemas], página 207, [Transposición], página 11, Sección 1.6.3 [Escritura de las particellas], página 220, Sección 1.8.1 [Escritura del texto], página 254.

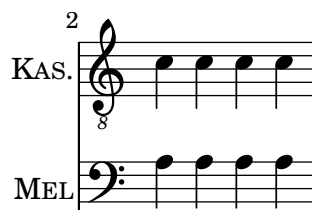
Snippets: Sección “Vocal music” en *Fragmentos de código*.

## Nombres de los personajes

Se suelen mostrar los nombres de los personajes a la izquierda del pentagrama cuando éste está dedicado a dicho personaje solamente:

```
\score {
  <<
    \new Staff {
      \set Staff.vocalName = \markup \smallCaps Kaspar
      \set Staff.shortVocalName = \markup \smallCaps Kas.
      \relative {
        \clef "G_8"
        c'4 c c c
        \break
        c4 c c c
      }
    }
    \new Staff {
      \set Staff.vocalName = \markup \smallCaps Melchior
      \set Staff.shortVocalName = \markup \smallCaps Mel
      \clef "bass"
      \relative {
        a4 a a a
        a4 a a a
      }
    }
  }
  >>
```

}



Cuando dos o más personajes comparten un pentagrama, el nombre del personaje se suele imprimir encima del pentagrama al principio de cada sección correspondiente a dicho personaje. Esto se puede hacer con elementos de marcado. Con frecuencia se usa un tipo de letra especial para este propósito.

```
\relative c' {
  \clef "G_8"
  c4^\markup \fontsize #1 \smallCaps Kaspar
  c c c
  \clef "bass"
  a4^\markup \fontsize #1 \smallCaps Melchior
  a a a
  \clef "G_8"
  c4^\markup \fontsize #1 \smallCaps Kaspar
  c c c
}
```



Como alternativa, si hay muchos cambios de personajes, puede ser más fácil establecer variables que contengan las definiciones de cada personaje, para así poder indicar el cambio de personaje de manera fácil y concisa.

```
kaspar = {
  \clef "G_8"
  \set Staff.shortVocalName = "Kas."
  \set Staff.midiInstrument = "voice oohs"
  <>^\markup \smallCaps "Kaspar"
}
```

```
melchior = {
  \clef "bass"
  \set Staff.shortVocalName = "Mel."
  \set Staff.midiInstrument = "choir aahs"
  <>^\markup \smallCaps "Melchior"
```

```

}

\relative c' {
  \kaspar
  c4 c c c
  \melchior
  a4 a a a
  \kaspar
  c4 c c c
}

```



## Véase también

Manual de aprendizaje: Sección “Organizar las piezas mediante variables” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 1.8 [Texto], página 254, Sección A.11 [Instrucciones de marcado de texto], página 739.

## Guías musicales

Se pueden insertar guías musicales en las partituras vocales, en los libros vocales y en las partes orquestales para indicar qué música de otra parte precede inmediatamente a una entrada. Asimismo, las guías se suelen insertar en la reducción de piano en las partituras vocales para indicar lo que está tocando cada parte orquestas. Esto sirve de ayuda al director cuando no está disponible una partitura del director.

El mecanismo básico para insertar guías se explica de forma exhaustiva en el texto principal, véase [Citar otras voces], página 224, y [Formateo de las notas guía], página 227. Pero cuando se deben insertar muchas guías, por ejemplo, como ayuda para el director en una partitura vocal, el nombre del instrumento se debe colocar cuidadosamente justo antes del comienzo de las notas guía, y cerca de ellas. El siguiente ejemplo muestra cómo se hace esto.

```

flute = \relative {
  s4 s4 e'' g
}
\addQuote "flute" { \flute }

pianoRH = \relative {
  c''4. g8
  % position name of cue-ing instrument just before the cue notes,
  % and above the staff
  <>^\markup { \right-align { \tiny "Flute" } }
  \cueDuring "flute" #UP { g4 bes4 }
}
pianoLH = \relative { c4 <c' e> e, <g c> }

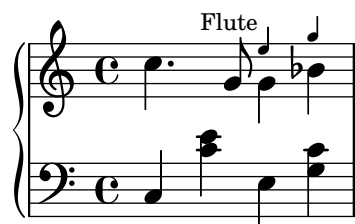
\score {
  \new PianoStaff <<
    \new Staff {
      \pianoRH

```

```

    }
    \new Staff {
      \clef "bass"
      \pianoLH
    }
  >>
}

```



Si un instrumento transpositor se está citando, la parte instrumental debe especificar la tonalidad de manera que la conversión de las notas citadas se realice automáticamente. El ejemplo de abajo muestra esta transposición para un clarinete en Si bemol. Las notas de este ejemplo son bajas en el pentagrama, por lo que se especifica `DOWN` en `\cueDuring` (para que las plicas se orienten hacia abajo) y el nombre del instrumento se coloca debajo del pentagrama.

```

clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

pianoRH = \relative c'' {
  \transposition c'
  % position name of cue-ing instrument below the staff
  <>_\markup { \right-align { \tiny "Clar." } }
  \cueDuring "clarinet" #DOWN { c4. g8 }
  g4 bes4
}
pianoLH = \relative { c4 <c' e> e, <g c> }

\score {
  <<
    \new PianoStaff <<
      \new Staff {
        \new Voice {
          \pianoRH
        }
      }
      \new Staff {
        \clef "bass"
        \pianoLH
      }
    >>
  >>
}

```

}



A partir de estos dos ejemplos debería quedar claro que insertar muchas notas guía en una partitura vocal puede ser tedioso, y la visión de las notas de la parte de piano quedaría entorpecida. Sin embargo, como muestra el siguiente fragmento de código, es posible definir una función musical para reducir la cantidad de texto que se teclea y para hacer que las notas del piano estén más claras.

## Fragmentos de código seleccionados

### *Añadir notas guía orquestales a una partitura vocal*

Este ejemplo muestra una forma de simplificar la adición de muchas notas guía orquestales a la reducción de piano en una partitura vocal. La función musical `\cueWhile` toma cuatro argumentos: la música de la que se toma la cita, como viene definida por `\addQuote`, el nombre que insertar antes de las notas guía, y después `#UP` o `#DOWN` para especificar `\voiceOne` con el nombre encima del pentagrama o bien `\voiceTwo` con el nombre debajo del pentagrama, y finalmente la música de piano con la que las notas guía deben aparecer en paralelo. El nombre del instrumento citado se posiciona a la izquierda de las notas guía. Se pueden citar muchos pasajes como guía, pero no se pueden superponer en el tiempo entre ellos.

```
cueWhile =
#(define-music-function
  (instrument name dir music)
  (string? string? ly:dir? ly:music?)
  #{
    \cueDuring $instrument #dir {
      \once \override TextScript.self-alignment-X = #RIGHT
      \once \override TextScript.direction = $dir
      <>-\markup { \tiny #name }
      $music
    }
  })

flute = \relative c' {
  \transposition c'
  s4 s4 e g
}
\addQuote "flute" { \flute }

clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }
```

```

singer = \relative c'' { c4. g8 g4 bes4 }
words = \lyricmode { here's the lyr -- ics }

pianoRH = \relative c'' {
  \transposition c'
  \cueWhile "clarinet" "Clar." #DOWN { c4. g8 }
  \cueWhile "flute" "Flute" #UP { g4 bes4 }
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  <<
    \new Staff {
      \new Voice = "singer" {
        \singer
      }
    }
    \new Lyrics {
      \lyricsto "singer"
      \words
    }
    \new PianoStaff <<
      \new Staff {
        \new Voice {
          \pianoRH
        }
      }
      \new Staff {
        \clef "bass"
        \pianoLH
      }
    }
  >>
}

```



## Véase también

Glosario musical: Sección “cue-notes” en *Glosario Musical*.

Referencia de la notación: Sección 5.5.1 [Alineación de objetos], página 678, Sección 5.4.2 [Dirección y posición], página 661, [Formateo de las notas guía], página 227, [Citar otras voces], página 224, Sección 5.6 [Uso de las funciones musicales], página 691.

Fragmentos de código: Sección “Vocal music” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “CueVoice” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

`\cueDuring` inserta automáticamente un contexto `CueVoice` y todas las notas guía se colocan dentro de este contexto. Ello significa que no es posible tener dos secuencias de notas guía superpuestas mediante esta técnica. Las secuencias superpuestas se podrían introducir declarando explícitamente contextos `CueVoice` separados y usando `\quoteDuring` para extraer e insertar las notas citadas.

## Música hablada

Efectos tales como el ‘parlato’ o el ‘Sprechgesang’ requieren que los intérpretes hablen sin altura definida pero con el ritmo adecuado; la notación de estos estilos se realiza mediante figuras con la cabeza en forma de cruz, como se muestra en [Cabezas de nota especiales], página 41.

## Diálogos encima de la música

Los diálogos sobre la música se imprimen normalmente por encima de los pentagramas en un estilo inclinado, con el comienzo de cada frase ligado a un momento musical en particular.

Para interjecciones cortas, basta un simple elemento de marcado.

```
\relative {
  a'4^\markup { \smallCaps { Alex - } \italic { He's gone } } a a a
  a4 a a^\markup { \smallCaps { Bethan - } \italic Where? } a
  a4 a a a
}
```



Para frases más largas puede ser necesario expandir la música de forma que que las palabras quepan holgadamente. No está previsto en LilyPond hacer esto de manera completamente automática, y puede ser necesaria cierta intervención manual sobre la disposición.

Para frases largas o pasajes con un diálogo en disposición apretada, puede dar mejores resultados la utilización de un contexto de letra. El contexto `Lyrics` no se debe asociar con una voz musical; en vez de ello, cada sección del diálogo recibe una duración explícita. Si hay una pausa en medio del diálogo, se debe separar la palabra final del resto y dividirse la duración entre ellas de forma que la música subyacente reciba un espaciado continuo y adecuado.

Si el diálogo se extiende a más de una línea, será necesario insertar `\breaks` manualmente y ajustar la colocación del diálogo para evitar que se extienda más allá del margen derecho. La última palabra del último compás de una línea debe separarse también, como se ha visto antes.

He aquí un ejemplo que ilustra cómo puede hacerse.

```
music = \relative {
  \repeat unfold 3 { a'4 a a a }
}

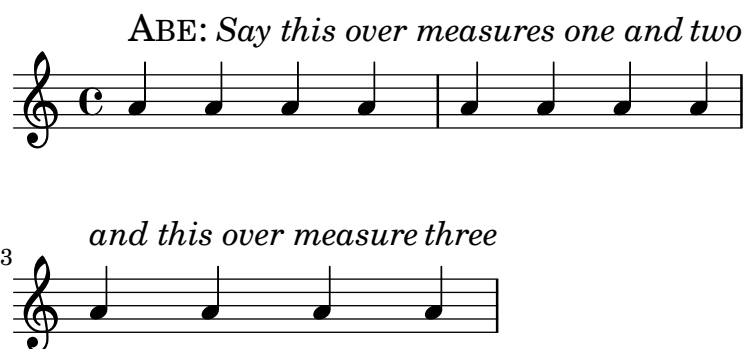
dialogue = \lyricmode {
  \markup {
    \fontsize #1 \upright \smallCaps Abe:
    "Say this over measures one and"
```

```

}4*7
"two"4 |
\break
"and this over measure"4*3
"three"4 |
}

\score {
  <<
    \new Lyrics \with {
      \override LyricText.font-shape = #'italic
      \override LyricText.self-alignment-X = #LEFT
    }
    { \dialogue }
    \new Staff {
      \new Voice { \music }
    }
  >>
}

```



## Véase también

Referencia de la notación: [Duración manual de las sílabas], página 293, Sección 1.8 [Texto], página 254.

Referencia de funcionamiento interno: Sección “LyricText” en *Referencia de Funcionamiento Interno*.

### 2.1.7 Cánticos salmos e himnos

La música y la letra de los cánticos, salmos e himnos suelen seguir un formato bien establecido dentro de cualquier iglesia en particular. Aunque los formatos pueden diferir de una iglesia a otra, los problemas de tipografiado que surgen son bastante similares, y se estudian en esta sección.

## Referencias para cánticos y salmos

El tipografiado del canto gregoriano en distintos estilos de notación antigua se describe en Sección 2.9 [Notación antigua], página 463.

## Véase también

Referencia de la notación: Sección 2.9 [Notación antigua], página 463.

Snippets: Sección “Vocal music” en *Fragmentos de código*.



## Preparar un cántico

La preparación moderna de los cánticos utiliza notación moderna con un número variable de elementos tomados de la notación antigua. Algunos de los elementos y métodos que considerar se presentan aquí.

Los cánticos usan con frecuencia figuras negras sin plica para indicar la altura, tomando las duraciones del ritmo hablado del texto.

```
stemOff = { \hide Staff.Stem }

\relative c' {
  \stemOff
  a'4 b c2 |
}
```



Los cánticos con frecuencia omiten las líneas divisorias o utilizan barras de compás recortadas o discontinuas para indicar pausas en la música. Para omitir todas las barras de compás de todos los pentagramas, suprime completamente el grabador de barras:

```
\score {
  \new StaffGroup <<
    \new Staff {
      \relative {
        a'4 b c2 |
        a4 b c2 |
        a4 b c2 |
      }
    }
  \new Staff {
    \relative {
      a'4 b c2 |
      a4 b c2 |
      a4 b c2 |
    }
  }
  >>
  \layout {
    \context {
      \Staff
      \remove "Bar_engraver"
    }
  }
}
```



Las líneas divisorias también se pueden quitar pentagrama a pentagrama:

```
\score {
  \new ChoirStaff <<
    \new Staff
      \with { \remove "Bar_engraver" } {
        \relative {
          a'4 b c2 |
          a4 b c2 |
          a4 b c2 |
        }
      }
    \new Staff {
      \relative {
        a'4 b c2 |
        a4 b c2 |
        a4 b c2 |
      }
    }
  >>
}
```



Para suprimir las líneas divisorias sólo de una sección de música, trátela como si fuera una cadencia. Si la sección es larga, podría necesitar insertar líneas divisorias “mudas” con `\bar ""` para indicar dónde se debe producir el salto de línea.

```
\relative a' {
  a4 b c2 |
  \cadenzaOn
  a4 b c2
  a4 b c2
  \bar ""
  a4 b c2
  a4 b c2
  \cadenzaOff
  a4 b c2 |
  a4 b c2 |
}
```



Los silencios o pausas en los cánticos pueden indicarse mediante barras de compás modificadas.

```
\relative a' {
  a4
```

```

\cadenzaOn
b c2
a4 b c2
\bar "'
a4 b c2
a4 b c2
\bar ";"
a4 b c2
\bar "!"
a4 b c2
\bar "||"
}

```



Como alternativa, se usa a veces la notación del canto gregoriano para las pausas o silencios, aunque el resto de la notación sea moderna. Este ejemplo utiliza una marca `\breathe` modificada:

```

divisioMinima = {
  \once \override BreathingSign.stencil = #ly:breathing-sign::divisio-minima
  \once \override BreathingSign.Y-offset = #0
  \breathe
}
divisioMaior = {
  \once \override BreathingSign.stencil = #ly:breathing-sign::divisio-maior
  \once \override BreathingSign.Y-offset = #0
  \breathe
}
divisioMaxima = {
  \once \override BreathingSign.stencil = #ly:breathing-sign::divisio-maxima
  \once \override BreathingSign.Y-offset = #0
  \breathe
}
finalis = {
  \once \override BreathingSign.stencil = #ly:breathing-sign::finalis
  \once \override BreathingSign.Y-offset = #0
  \breathe
}

\score {
  \relative {
    g'2 a4 g
    \divisioMinima
    g2 a4 g
    \divisioMaior
    g2 a4 g
    \divisioMaxima
    g2 a4 g
    \finalis
  }
  \layout {

```

```

\context {
  \Staff
  \remove "Bar_engraver"
}
}
}

```



Los cánticos suelen omitir la indicación de compás y con frecuencia omiten la clave también.

```

\score {
  \new Staff {
    \relative {
      a'4 b c2 |
      a4 b c2 |
      a4 b c2 |
    }
  }
  \layout {
    \context {
      \Staff
      \remove "Bar_engraver"
      \remove "Time_signature_engraver"
      \remove "Clef_engraver"
    }
  }
}

```



Los cánticos de salmos en la tradición anglicana suelen ser *sencillos*, con siete compases de música, o *dobles*, con dos períodos de siete compases. Cada grupo de siete compases se divide en dos mitades, que corresponden a las dos mitades de cada verso, normalmente separados mediante una doble línea divisoria. Sólo se utilizan redondas y blancas. El primer compás de cada mitad siempre contiene un único acorde en redondas. Ésta es la “nota de recitado”. Los cánticos se suelen centrar sobre la página.

```

SopranoMusic = \relative {
  g'1 | c2 b | a1 | \bar "||"
  a1 | d2 c | c b | c1 | \bar "||"
}

```

```

AltoMusic = \relative {
  e'1 | g2 g | f1 |
  f1 | f2 e | d d | e1 |
}

```

```

TenorMusic = \relative {
  c'1 | c2 c | c1 |
  d1 | g,2 g | g g | g1 |
}

```

```

}

BassMusic = \relative {
  c1 | e2 e | f1 |
  d1 | b2 c | g' g | c,1 |
}

global = {
  \time 2/2
}

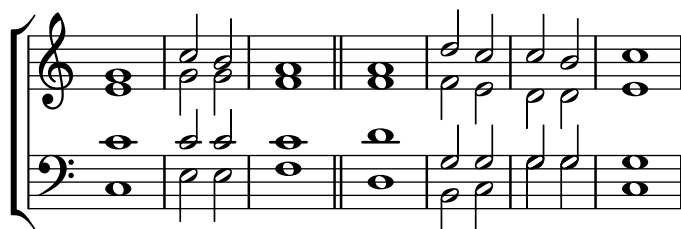
% Use markup to center the chant on the page
\markup {
  \fill-line {
    \score { % centered
      <<
        \new ChoirStaff <<
          \new Staff <<
            \global
            \clef "treble"
            \new Voice = "Soprano" <<
              \voiceOne
              \SopranoMusic
            >>
            \new Voice = "Alto" <<
              \voiceTwo
              \AltoMusic
            >>
          >>
          \new Staff <<
            \clef "bass"
            \global
            \new Voice = "Tenor" <<
              \voiceOne
              \TenorMusic
            >>
            \new Voice = "Bass" <<
              \voiceTwo
              \BassMusic
            >>
          >>
        >>
      >>
    }
  }
  \layout {
    \context {
      \Score
      \override SpacingSpanner.base-shortest-duration = #(ly:make-moment 1/2)
    }
    \context {
      \Staff
      \remove "Time_signature_engraver"
    }
  }
}

```

```

    }
  } % End score
}
} % End markup

```



Otros enfoques para la preparación de este cántico se muestran en el primero de los siguientes fragmentos de código.

## Fragmentos de código seleccionados

### *Notación de responsos o salmos*

Este tipo de notación se utiliza para los cantos salmódicos, en que las estrofas no siempre tienen la misma longitud.

```

stemOff = \hide Staff.Stem
stemOn  = \undo \stemOff

\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \bar "||"
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \bar "||"
    \stemOff a'\breve^\markup { \italic flexe }
    \stemOn g'2 \bar "||"
  }
}

```



Los cánticos y otros textos litúrgicos pueden disponerse de una forma más libre, y posiblemente usen elementos notacionales tomados de la música antigua. Con frecuencia la letra se imprime por debajo y alineada con las notas. En este caso, las notas se disponen espaciadas de acuerdo con las sílabas en vez de hacerlo con las duraciones de las notas.

### *Plantilla para notación de música antigua (transcripción moderna de canto gregoriano)*

Este ejemplo muestra cómo hacer una transcripción moderna de canto gregoriano. El canto gregoriano no tiene compás ni plicas; utiliza solamente cabezas de nota de blanca y de negra, y unas marcas especiales que indican silencios de distintas longitudes.

```

\include "gregorian.ly"

```

```

chant = \relative c' {
  \set Score.timing = ##f
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g) a2 \finalis
}

verba = \lyricmode {
  Lo -- rem ip -- sum do -- lor sit a -- met
}

\score {
  \new Staff <<
    \new Voice = "melody" \chant
    \new Lyrics = "one" \lyricsto melody \verba
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \remove "Bar_engraver"
      \hide Stem
    }
    \context {
      \Voice
      \override Stem.length = #0
    }
    \context {
      \Score
      barAlways = ##t
    }
  }
}

```



## Véase también

Manual de aprendizaje: Sección “Visibilidad y color de los objetos” en *Manual de Aprendizaje*, Sección “Plantillas de conjuntos vocales” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 2.9 [Notación antigua], página 463, [Barras de compás], página 104, Sección 5.1.4 [Modificar los complementos (plug-ins) de contexto], página 631, Sección 2.9.4 [Tipografiado del canto gregoriano], página 475, [Música sin compasear], página 79, Sección 5.4.7 [Visibilidad de los objetos], página 670.

## Puntuación de un salmo

El texto de un salmo anglicano se suele imprimir en versos separados debajo del canto.

Los cantos sencillos (con siete compases) se repiten para cada verso. Los cantos dobles (con 14 compases) se repiten para cada par de versos. Se insertan marcas dentro del texto para indicar la forma en que encajan con el canto. Cada verso se divide en dos mitades. Se suele utilizar un signo de dos puntos para indicar esta división. Esto corresponde a la doble línea divisoria de la música. El texto antes de los dos puntos se canta con los tres primeros compases de la música; el texto después de los dos puntos se canta con los últimos cuatro compases.

Se insertan líneas de compás simples (o en algunos casos una coma invertida u otro símbolo similar) entre el texto para indicar dónde caen las líneas divisorias de la música. En el modo de marcado, se puede introducir una línea de compás simple con el símbolo de comprobación de compás, la barra vertical, |.

```
\markup {
  \fill-line {
    \column {
      \left-align {
        \line { O come let us sing | unto the | Lord : let }
        \line { us heartily rejoice in the | strength of | our }
        \line { sal- | -vation. }
      }
    }
  }
}
```

O come let us sing | unto the | Lord : let  
us heartily rejoice in the | strength of | our  
sal- | -vation.

Otros símbolos pueden requerir glifos de las fuentes tipográficas `fetaMusic`. Para ver más detalles, consulte Sección 1.8.3 [Tipografías], página 279.

```
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line { O come let us sing \tick unto the \tick Lord : let }
        \line {
          us heartily rejoice in the \tick strength of \tick our
        }
        \line { sal \tick vation. }
      }
    }
  }
}
```

O come let us sing 'unto the ' Lord : let  
us heartily rejoice in the ' strength of ' our  
sal 'vation.

Donde hay una redonda en un compás, todo el texto que corresponde a ese compás se recita sobre esa nota en el ritmo de la palabra. Donde hay dos notas en un compás, suele haber



solamente una o dos sílabas correspondientes. Si hay más de dos sílabas, se suele insertar un punto para indicar dónde se produce el cambio de nota.

```
dot = \markup {
  \raise #0.7 \musicglyph "dots.dot"
}
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line {
          O come let us sing \tick unto \dot the \tick Lord : let
        }
        \line {
          us heartily rejoice in the \tick strength of \tick our
        }
        \line { sal \tick vation. }
      }
    }
  }
}
```

O come let us sing 'unto • the ' Lord : let  
us heartily rejoice in the ' strength of ' our  
sal ' vation.

En algunos salterios (libros de salmos) se usa un asterisco para indicar un corte en una sección recitada en lugar de una coma y las sílabas acentuadas o ligeramente alargadas se indican en estilo negrita.

```
dot = \markup {
  \raise #0.7 \musicglyph "dots.dot"
}
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line { Today if ye will hear his voice * }
        \line {
          \concat { \bold hard en }
          | not your | hearts : as in the pro-
        }
        \line { vocation * and as in the \bold day of tempt- | }
        \line { -ation | in the | wilderness. }
      }
    }
  }
}
```

}

Today if ye will hear his voice \*  
**harden** | not your | hearts : as in the pro-  
 vocation \* and as in the **day** of tempt- |  
 -ation | in the | wilderness.

En otros salterios se coloca un acento sobre la sílaba para indicar el énfasis.

```
tick = \markup {
  \raise #2 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line {
          O come let us \concat {
            si \combine \tick ng
          }
          | unto the | Lord : let
        }
        \line {
          us heartily \concat {
            rejo \combine \tick ice
          }
          in the | strength of | our
        }
        \line { sal- | -vation. }
      }
    }
  }
}
```

O come let us **síng** | unto the | Lord : let  
 us heartily re**joíce** in the | strength of | our  
 sal- | -vation.

El uso del marcado para centrar texto y para disponer las líneas en columnas se describe en Sección 1.8.2 [Formatear el texto], página 264.

Casi todos estos elementos se muestran en uno u otro de los dos versos en la plantilla, véase Sección “Salmos” en *Manual de Aprendizaje*.

## Véase también

Manual de aprendizaje: Sección “Salmos” en *Manual de Aprendizaje*, Sección “Plantillas de conjuntos vocales” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 1.8.3 [Tipografías], página 279, Sección 1.8.2 [Formatear el texto], página 264.

## Compases parciales en melodías de himno

Con frecuencia las melodías de himno comienzan y terminan cada línea de música con compases parciales de forma que cada línea de música corresponde exactamente con una línea de texto.

Esto requiere una instrucción `\partial` al principio de la música e instrucciones `\bar "||"` ó `\bar "||"` al final de cada línea.

### *Plantilla para himnos*

Este fragmento de código muestra una forma de preparar un himno cuando cada línea comienza con un compás parcial. También muestra cómo añadir los versos como texto independiente debajo de la música.

```
Timeline = {
  \time 4/4
  \tempo 4=96
  \partial 2
  s2 | s1 | s2 \breathe s2 | s1 | s2 \bar "||" \break
  s2 | s1 | s2 \breathe s2 | s1 | s2 \bar "||"
}

SopranoMusic = \relative g' {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

AltoMusic = \relative c' {
  d4 d | d d d d | d d d d | d d d d | d2
  d4 d | d d d d | d d d d | d d d d | d2
}

TenorMusic = \relative a {
  b4 b | b b b b | b b b b | b b b b | b2
  b4 b | b b b b | b b b b | b b b b | b2
}

BassMusic = \relative g {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

global = {
  \key g \major
}

\score { % Start score
  <<
    \new PianoStaff << % Start pianostaff
      \new Staff << % Start Staff = RH
        \global
        \clef "treble"
        \new Voice = "Soprano" << % Start Voice = "Soprano"
          \Timeline
          \voiceOne
          \SopranoMusic
        >> % End Voice = "Soprano"
      \new Voice = "Alto" << % Start Voice = "Alto"
```

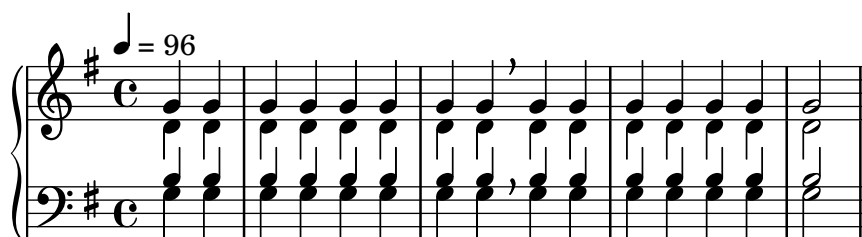
```

\Timeline
\voiceTwo
\AltoMusic
>> % End Voice = "Alto"
>> % End Staff = RH
\new Staff << % Start Staff = LH
\global
\clef "bass"
\new Voice = "Tenor" << % Start Voice = "Tenor"
\Timeline
\voiceOne
\TenorMusic
>> % End Voice = "Tenor"
\new Voice = "Bass" << % Start Voice = "Bass"
\Timeline
\voiceTwo
\BassMusic
>> % End Voice = "Bass"
>> % End Staff = LH
>> % End pianostaff
>>
} % End score

\markup {
  \fill-line {
    ""
    {
      \column {
        \left-align {
          "This is line one of the first verse"
          "This is line two of the same"
          "And here's line three of the first verse"
          "And the last line of the same"
        }
      }
    }
  }
  ""
}

\paper { % Start paper block
  indent = 0 % don't indent first system
  line-width = 130 % shorten line length to suit music
} % End paper block

```





This is line one of the first verse  
 This is line two of the same  
 And here's line three of the first verse  
 And the last line of the same

### 2.1.8 Música vocal antigua

Está contemplada la música vocal en estilo de tipografía antiguo, como se explica en Sección 2.9 [Notación antigua], página 463.

### Véase también

Referencia de la notación: Sección 2.9 [Notación antigua], página 463.

## 2.2 Teclados y otros instrumentos de varios pentagramas

**Un peu retenu**  
*très expressif*

**Rall.** *long*

**a Tempo**

*pp*

*ped.*



Esta sección se ocupa de varios aspectos de la notación musical que son exclusivos de los instrumentos de teclado y otros instrumentos cuya notación se realiza sobre varios pentagramas, como el arpa o el vibráfono. A los efectos de nomenclatura, en esta sección se denominan abreviadamente “teclados” a todo este grupo de instrumentos de varios pentagramas, aunque algunos de ellos no tienen teclado.

### 2.2.1 Notación común para instrumentos de teclado

Esta sección trata aspectos de notación que pueden aparecer en casi todos los instrumentos de teclado.

#### Referencias para teclados

La notación de los instrumentos de teclado se suele realizar con sistemas de piano. Éstos son dos pentagramas normales agrupados mediante una llave. También se utiliza esta notación para otros instrumentos de teclado. La música de órgano se escribe normalmente con dos pentagramas dentro de un grupo `PianoStaff` y un tercer pentagrama normal para los pedales.

Los pentagramas son hasta cierto punto independientes, pero a veces las voces pueden cruzarse entre los dos pentagramas. Esta sección trata técnicas de notación que son particulares de la música de teclado.

En otros lugares se tratan varios problemas comunes de la música de teclado:

- La música para teclado suele contener varias voces y el número de éstas puede variar ampliamente; esto se describe en [Resolución de las colisiones], página 186.
- La música para teclado se puede escribir en paralelo, como se describe en [Escribir música en paralelo], página 197.
- Los matices dinámicos se pueden colocar en un contexto `Dynamics`, entre los dos contextos `Staff` para alinear las indicaciones de matiz sobre una línea horizontal equidistante de los dos pentagramas; véase [Matices dinámicos], página 131.
- Las digitaciones se indican como puede verse en [Indicaciones de digitación], página 240.
- Las indicaciones de pedal de órgano se insertan como articulaciones, véase Sección A.14 [Lista de articulaciones], página 800.
- Las líneas verticales de rejilla se pueden mostrar como se describe en [Líneas de rejilla], página 249.
- La música para teclado incluye con frecuencia ligaduras *Laissez vibrer* así como ligaduras sobre arpeggios y trémolos, descritos en [Ligaduras de unión], página 58.
- La colocación de arpeggios a través de varias voces y pentagramas se cubre en [Arpeggio], página 152.
- Las indicaciones de trémolo se describen en [Repeticiones de trémolo], página 173.
- Varios de los trucos que pueden ser necesarios en música para teclado se muestran en Sección “Ejemplos reales de música” en *Manual de Aprendizaje*.

- Las notas ocultas se pueden usar para producir ligaduras de unión que se cruzan entre voces, como se explica en Sección “Otras aplicaciones de los trucos” en *Manual de Aprendizaje*.

## Véase también

Manual de aprendizaje: Sección “Ejemplos reales de música” en *Manual de Aprendizaje*, Sección “Otras aplicaciones de los trucos” en *Manual de Aprendizaje*.

Referencia de la notación: [Nombres de instrumentos], página 220, [Resolución de las colisiones], página 186, [Escribir música en paralelo], página 197, [Indicaciones de digitación], página 240, Sección A.14 [Lista de articulaciones], página 800, [Líneas de rejilla], página 249, [Ligaduras de unión], página 58, [Arpeggio], página 152, [Repeticiones de trémolo], página 173.

Referencia de funcionamiento interno: Sección “PianoStaff” en *Referencia de Funcionamiento Interno*.

Fragmentos de código: Sección “Keyboards” en *Fragmentos de código*.

## Cambiar de pentagrama manualmente

Las voces se pueden cambiar de pentagrama manualmente usando la instrucción

```
\change Staff = nombre_del_pentagrama
```

La cadena *nombre\_del\_pentagrama* es el nombre del pentagrama. Cambia la voz actual de su pentagrama en curso al pentagrama de nombre *nombre\_del\_pentagrama*. Los valores más corrientes para *nombre\_del\_pentagrama* son "arriba" y "abajo", o bien "MD" y "MI".

El pentagrama al que salta la voz debe existir en el momento del salto. Si es necesario, los pentagramas se tienen que “mantener con vida”, véase Sección 5.1.3 [Mantener vivos los contextos], página 628, o instanciados explícitamente, por ejemplo mediante la utilización del acorde vacío, <>, véase [Notas en acorde], página 176.

```
\new PianoStaff <<
  \new Staff = "up" {
    % enforce creation of all contexts at this point of time
    <>
    \change Staff = "down" c2
    \change Staff = "up" c'2
  }
  \new Staff = "down" {
    \clef bass
    % keep staff alive
    s1
  }
>>
```



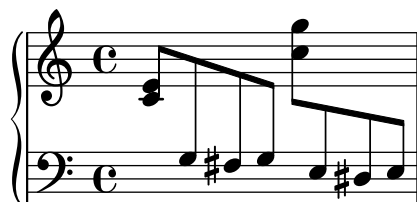
Las notas de pentagrama cruzado reciben la barra automáticamente:

```
\new PianoStaff <<
  \new Staff = "up" {
    <e' c'>8
    \change Staff = "down"
```

```

g8 fis g
\change Staff = "up"
<g'' c''>8
\change Staff = "down"
e8 dis e
\change Staff = "up"
}
\new Staff = "down" {
  \clef bass
  % keep staff alive
  s1
}
>>

```



Si el barrado necesita algún trucaje, efectúe en primer lugar los cambios en las direcciones de las plicas. Las posiciones de las barras se miden posteriormente a partir del centro del pentagrama que está más cerca de la barra. Para ver un ejemplo sencillo de trucaje de las barras, consulte Sección “Arreglar notación con superposiciones” en *Manual de Aprendizaje*.

Se puede producir una superposición en la notación cuando las voces se cruzan de un pentagrama a otro:

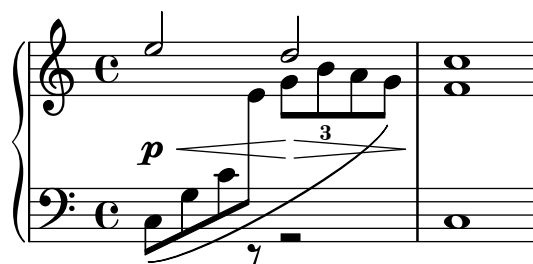
```

\new PianoStaff <<
  \new Staff = "up" {
    \voiceOne
    % Make space for fingering in the cross-staff voice
    \once\override DynamicLineSpanner.staff-padding = #4
    e''2\p\< d''\>
    c''1\!
  }
  \new Staff = "down" <<
  {
    \clef bass
    s4. e,8\rest g,2\rest
    c1
  } \ {
    c8\ ( g c'
    \change Staff = "up"
    e' g' b'-3 a' g'\ )
    f'1
  }
>>

```



&gt;&gt;



La plica y la ligadura de expresión se superponen a la línea del símbolo de dinámica porque la resolución automática de colisiones se suspende para las barras, ligaduras y otros objetos de extensión que conectan notas situadas en distintos pentagramas, así como para las plicas y articulaciones si su colocación está afectada por un objeto de extensión de pentagrama cruzado. Las colisiones resultantes se pueden resolver manualmente donde sea necesario, utilizando los métodos descritos en la sección Sección “Arreglar notación con superposiciones” en *Manual de Aprendizaje*.

## Véase también

Manual de aprendizaje: Sección “Arreglar notación con superposiciones” en *Manual de Aprendizaje*.

Referencia de la notación: [Plicas], página 245, [Barras automáticas], página 88, Sección 5.1.3 [Mantener vivos los contextos], página 628.

Fragmentos de código: Sección “Keyboards” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Beam” en *Referencia de Funcionamiento Interno*, Sección “ContextChange” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

El mecanismo para evitar las colisiones de barras de corchea no funciona para las barras automáticas que terminan justo antes de un cambio de pauta. En este caso, utilice barras manuales.

## Cambiar de pentagrama automáticamente

Se puede hacer que las voces cambien automáticamente entre los pentagramas inferior y superior. La sintaxis para conseguir esto es

```
\autoChange ...música...
```

Esto creará dos pentagramas dentro del grupo actual de pentagramas (normalmente un **PianoStaff**), llamados "up" (arriba) y "down" (abajo). El pentagrama inferior estará en clave de Fa de forma predeterminada. El cambiador automático hace el cambio basándose en la altura (el Do central es el punto de inflexión), y se adelanta saltando sobre los silencios para hacer cambios por adelantado.

```
\new PianoStaff {
  \autoChange {
    g4 a b c'
    d'4 r a g
  }
}
```

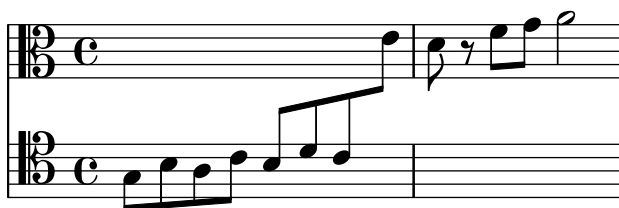
}



Se pueden especificar otras notas para el punto de corte. Si los pentagramas no han sido instanciados de forma explícita, se pueden usar otras claves.

```
music = {
    g8 b a c' b8 d' c'8 e'
    d'8 r f' g' a'2
}
```

```
\autoChange d' \music
\autoChange b \with { \clef soprano } \music
\autoChange d' \with { \clef alto } \with { \clef tenor } \music
```



Una sección marcada `\relative` que se encuentra fuera de `\autoChange` no tiene efecto sobre las notas de la música; por tanto, en caso necesario escriba `\relative` dentro de `\autoChange`.

Si se necesita más control sobre los pentagramas individuales, se pueden crear manualmente con los nombres "up" y "down". Entonces, la instrucción `\autoChange` alternará su voz entre los pentagramas existentes.

**Nota:** Si se crean los pentagramas manualmente, *se deben* llamar "up" y "down" (en inglés).

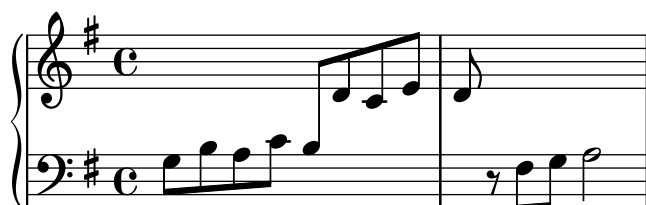
Por ejemplo, esto es necesario para colocar una armadura de tonalidad en el pentagrama inferior:

```
\new PianoStaff <<
```

```

\new Staff = "up" {
  \new Voice = "melodiaUno" {
    \key g \major
    \autoChange \relative {
      g8 b a c b d c e
      d8 r fis, g a2
    }
  }
}
\new Staff = "down" {
  \key g \major
  \clef bass
}
>>

```



## Véase también

Referencia de la notación: [Cambiar de pentagrama manualmente], página 354.

Fragmentos de código: Sección “Keyboards” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “AutoChangeMusic” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Los cambios de pentagrama pueden no acabar en lugares óptimos. Para conseguir una salida de alta calidad es mejor especificar manualmente los cambios de pentagrama.

Los acordes no se dividen entre los pentagramas; se asignan a un pentagrama en función de la primera nota nombrada dentro de la construcción del acorde.

## Líneas de cambio de pentagrama

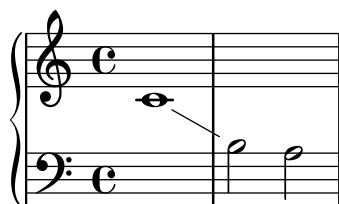
Cuando una voz cambia a otro pentagrama, se puede imprimir automáticamente una línea que conecte las notas:

```

\new PianoStaff <<
  \new Staff = "one" {
    \showStaffSwitch
    c'1
    \change Staff = "two"
    b2 a
  }
  \new Staff = "two" {
    \clef bass
    s1*2
  }
}

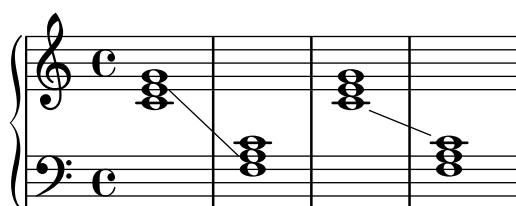
```

&gt;&gt;



Una línea de cambio de pauta entre acordes conecta las “últimas notas” de cada uno de ellos según se han escrito en el archivo de entrada; Esto se puede usar para ajustar rápidamente las posiciones verticales del comienzo y el final de la línea.

```
\new PianoStaff <<
  \new Staff = "one" {
    <c' e' g'>1
    \showStaffSwitch
    \change Staff = "two"
    <a c' f>1
    \hideStaffSwitch
    \change Staff = "one"
    <e' g' c'>1
    \showStaffSwitch
    \change Staff = "two"
    <f a c'>1
  }
  \new Staff = "two" {
    \clef bass
    s1*4
  }
>>
```



## Instrucciones predefinidas

`\showStaffSwitch`, `\hideStaffSwitch`.

## Véase también

Fragmentos de código: Sección “Keyboards” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Note\_head\_line\_engraver” en *Referencia de Funcionamiento Interno*, Sección “VoiceFollower” en *Referencia de Funcionamiento Interno*.

## Fragmentos de código seleccionados

### *Plicas de pentagrama cruzado*

Este fragmento de código muestra el uso del grabador `Span_stem_engraver` y de `\crossStaff` para conectar automáticamente plicas de un pentagrama a otro. No es necesario especificar la

longitud de la plica porque la distancia variable entre las cabezas de las notas y los pentagramas se calcula automáticamente.

```
\layout {
  \context {
    \PianoStaff
    \consists "Span_stem_engraver"
  }
}

{
  \new PianoStaff <<
    \new Staff {
      <b d'>4 r d'16\> e'8. g8 r\!
      e'8 f' g'4 e'2
    }
    \new Staff {
      \clef bass
      \voiceOne
      \autoBeamOff
      \crossStaff { <e g>4 e, g16 a8. c8} d
      \autoBeamOn
      g8 f g4 c2
    }
  >>
}
```



Por el momento no se puede especificar este grabador por su nombre entre comillas, sino prefijando el nombre con un símbolo de almohadilla (#), a causa de la manera en que está implementado.

### ***Indicar acordes de pentagrama cruzado con corchetes de arpeggio***

Un corchete de arpeggio puede indicar que se tienen que tocar con la misma mano notas que están en dos pentagramas distintos. Para hacerlo, el `PianoStaff` se debe configurar para que acepte símbolos de arpeggio de pentagrama cruzado y los símbolos de arpeggio se deben configurar a la forma de corchete en el contexto de `PianoStaff`.

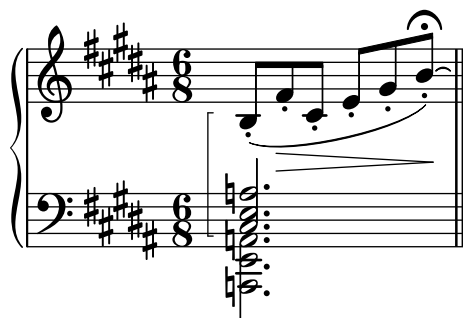
(Debussy, Les collines d'Anacapri, m. 65)

```
\new PianoStaff <<
  \set PianoStaff.connectArpeggios = ##t
  \override PianoStaff.Arpeggio.stencil =
    #ly:arpeggio::brew-chord-bracket
  \new Staff {
    \relative c' {
      \key b \major
      \time 6/8
```

```

        b8-(\arpeggio fis'-.\> cis-.
        e-. gis-. b-.)\!\fermata^\laissezVibrer \bar "||"
    }
}
\new Staff {
  \relative c' {
    \clef bass
    \key b \major
    <<
    {
      <a e cis>2.\arpeggio
    }
    \\\
    {
      <a, e a,>2.
    }
    >>
  }
}
>>

```



## Véase también

Fragmentos de código: Sección “Keyboards” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Stem” en *Referencia de Funcionamiento Interno*.

### 2.2.2 Piano

Esta sección trata de aspectos de la notación directamente relacionados con el piano.

#### Pedales de piano

En general, los pianos tienen tres pedales que alteran la forma de producir el sonido: el pedal de mantenimiento (pedal *derecho* o pedal *forte*), el pedal *sostenuto* (sos., pedal central) y el pedal *una corda* (U.C., pedal izquierdo). Los pedales de mantenimiento como el pedal derecho del piano se encuentran también en el vibráfono y en la celesta.

```

\relative {
  c''4\sustainOn d e g
  <c, f a>1\sustainOff
  c4\sostenutoOn e g c,
  <bes d f>1\sostenutoOff
  c4\unaCorda d e g
  <d fis a>1\treCorde

```

}



Existen tres estilos de indicaciones de pedal: textual, corchete y mixta. El pedal derecho y el pedal de una corda utilizan el estilo textual de forma predeterminada, mientras que el pedal sostenuto utiliza el método mixto de forma predeterminada.

```
\relative {
  c' '4\sustainOn g c2\sustainOff
  \set Staff.pedalSustainStyle = #'mixed
  c4\sustainOn g c d
  d\sustainOff\sustainOn g, c2\sustainOff
  \set Staff.pedalSustainStyle = #'bracket
  c4\sustainOn g c d
  d\sustainOff\sustainOn g, c2
  \bar "|."
}
```



La colocación de las instrucciones de pedal se corresponden con el movimiento físico del pedal derecho en el transcurso de la ejecución. La pedalización hasta la barra final de compás se indica mediante la omisión de la instrucción final de levantar el pedal.

Las indicaciones de pedal se pueden poner dentro de un contexto `Dynamics`, que los alinea sobre una línea horizontal.

## Véase también

Referencia de la notación: [Ligaduras de unión], página 58.

Fragmentos de código: Sección “Keyboards” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “SustainPedal” en *Referencia de Funcionamiento Interno*, Sección “SustainPedalLineSpanner” en *Referencia de Funcionamiento Interno*, Sección “SustainEvent” en *Referencia de Funcionamiento Interno*, Sección “SostenutoPedal” en *Referencia de Funcionamiento Interno*, Sección “SostenutoPedalLineSpanner” en *Referencia de Funcionamiento Interno*, Sección “SostenutoEvent” en *Referencia de Funcionamiento Interno*, Sección “UnaCordaPedal” en *Referencia de Funcionamiento Interno*, Sección “UnaCordaPedalLineSpanner” en *Referencia de Funcionamiento Interno*, Sección “UnaCordaEvent” en *Referencia de Funcionamiento Interno*, Sección “PianoPedalBracket” en *Referencia de Funcionamiento Interno*, Sección “Piano\_pedal\_engraver” en *Referencia de Funcionamiento Interno*.

### 2.2.3 Acordeón

Esta sección se ocupa de la notación que es exclusiva del acordeón.

## Símbolos de Discanto

Los acordeones se suelen construir con más de un conjunto de lengüetas que pueden estar al unísono, una octava por encima, o una octava por debajo de la altura escrita. Cada fabricante de

acordeones utiliza distintos nombres para los *cambios* que seleccionan las distintas combinaciones de lengüetas, como *oboe*, *musette* o *bandoneón*, de manera que ha entrado en uso un sistema de signos para simplificar las instrucciones de ejecución.

Puede verse una lista completa de todos los registros de acordeón disponibles en Sección “Registros de acordeón” en *Referencia de la Notación*

## Fragmentos de código seleccionados

### *Accordion register symbols*

Accordion register symbols are available as `\markup` as well as as standalone music events (as register changes tend to occur between actual music events). Bass registers are not overly standardized. The available commands can be found in ‘Discant symbols’ in the Notation Reference.

```

\\
{ d r a r bes r }
>> |
<cis e a>1
}

\new Staff \relative {
  \clef treble
  \freeBass "1"
  r8 d'32 s16. c32 s16. bes32 s16. a32[ cis] s16
  \clef bass \stdBass "Master"
  <<
    { r16 <f, bes d>^"b" r <e a c>^"am" r <d g bes>^"gm" |
      <e a cis>1^"a" }
    \\
    { d8_"D" c_"C" bes_"B" | a1_"A" }
  >>
}
>>

```



## Véase también

Fragmentos de código: Sección “Keyboards” en *Fragmentos de código*.

### 2.2.4 Arpa

Esta sección trata sobre asuntos de notación específicos del arpa.



## Referencias para notación de arpa

Ciertas características comunes de la música de arpa se estudian en otras partes de la documentación:

- El glissando es la técnica más característica del arpa, [Glissando], página 147.
- El *bisbigliando* se escribe como un trémolo, [Repeticiones de trémolo], página 173.
- Los armónicos naturales se estudian bajo el epígrafe [Armónicos], página 366.
- Para los arpeggios dirigidos y los no arpeggios, véase [Arpeggio], página 152.

## Véase también

Referencia de la notación: [Repeticiones de trémolo], página 173, [Glissando], página 147, [Arpeggio], página 152, [Armónicos], página 366.

## Pedales de arpa

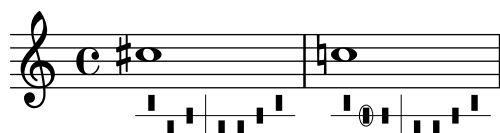
Las arpas tienen siete cuerdas por octava que pueden sonar a la altura natural, bemol o sostenido. En el arpa de palancas o arpa celta, cada cuerda se ajusta individualmente, pero en las arpas de pedales todas las cuerdas con igual nombre de nota se controlan por un único pedal. Visto de izquierda a derecha desde la posición del ejecutante, los pedales son Re, Do y Si a la izquierda y Mi, Fa, Sol y La a la derecha. La posición de los pedales se puede indicar con indicaciones textuales:

```
\textLengthOn
cis''1\_markup \concat \vcenter {
  [D \flat C \sharp B | E \sharp F \sharp G A \flat] }
c''!1\_markup \concat \vcenter {
  [ C \natural ] }
```



o mediante diagramas de pedal:

```
\textLengthOn
cis''1\_markup { \harp-pedal "^v-|vv-^" }
c''!1\_markup { \harp-pedal "^o--|vv-^" }
```



La instrucción `\harp-pedal` acepta una cadena de caracteres en la que  $\wedge$  es la posición alta del pedal (altura bemol),  $-$  es la posición intermedia (altura natural),  $v$  es la posición baja (altura sostenido), y  $|$  es la línea vertical separadora. Si se antepone  $o$ , el siguiente símbolo de pedal se rodea por una elipse.

## Véase también

Referencia de la notación: [Guiones de texto], página 257, Sección A.11.5 [Instrument Specific Markup], página 780.

## 2.3 Instrumentos de cuerda sin trastes

**1** **lentement**

*fatigué* s. vib. n. p. vib. s. vib.

IV IV IV

*mf* *mf* *mf* *ff* *pp*

*accel...* s.p. n. s.p. n. p. vib.

IV IV

*mf* *ff*

s.p. n. s.p. n. p. vib. m. vib.

IV IV IV

*ppp*

Esta sección aporta información y referencias que resultan útiles si se escribe música para instrumentos de cuerda sin trastes, principalmente instrumentos de cuerda orquestales.

### 2.3.1 Notación común para cuerdas sin trastes

Hay poca notación especializada para los instrumentos de cuerdas sin trastes. La notación de la música se realiza en un solo pentagrama, y se suele usar una sola voz. Se pueden necesitar dos voces para algunos pasajes en dobles cuerdas o en divisi.

### Referencias para cuerdas sin trastes

La mayor parte de la notación que resulta de utilidad para las cuerdas orquestales y otros instrumentos de arco se estudia en otras partes del manual:

- Las indicaciones textuales como “pizz.” y “arco” se añaden como texto simple: véase [Guiones de texto], página 257.
- Las digitaciones, entre ellas la indicación de pulgar, se describen en [Indicaciones de digitación], página 240.
- Se pueden añadir números de cuerda (en general como números romanos para los instrumentos de cuerda frotada) como se explica en [Indicación de los números de cuerda], página 369.
- Las dobles cuerdas se indican normalmente escribiendo un acorde, véase [Notas en acorde], página 176. Hay otras indicaciones que se pueden añadir para la interpretación de acordes, véase [Arpeggio], página 152.

- Hay plantillas para cuarteto de cuerda en Sección “Plantillas de cuarteto de cuerda” en *Manual de Aprendizaje*. Otras se muestran en la sección de fragmentos de código.

## Véase también

Manual de aprendizaje: Sección “Plantillas de cuarteto de cuerda” en *Manual de Aprendizaje*.

Referencia de la notación: [Guiones de texto], página 257, [Indicaciones de digitación], página 240, [Notas en acorde], página 176, [Arpeggio], página 152.

Fragmentos de código: Sección “Unfretted strings” en *Fragmentos de código*.

## Indicaciones de arco

Las indicaciones de arco se crean como articulaciones, que se describen en [Articulaciones y ornamentos], página 128.

Las instrucciones de arco `\upbow` (arco arriba) y `\downbow` (arco abajo) se usan con ligaduras de expresión de la siguiente manera:

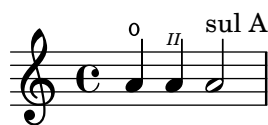
```
\relative { c'4(\downbow d) e(\upbow f) }
```



Se pueden usar números romanos para los números de cuerda (en sustitución de la opción predeterminada, que son cifras arábigas rodeadas por un círculo), tal y como se explica en [Indicación de los números de cuerda], página 369.

Como alternativa, se pueden imprimir las indicaciones de cuerda mediante instrucciones de marcado; los símbolos de articulación también pueden indicar cuerdas al aire.

```
a'4 \open
\romanStringNumbers
a'\2
a'2^\markup { \small "sul A" }
```



## Instrucciones predefinidas

`\downbow`, `\upbow`, `\open`, `\romanStringNumbers`.

## Véase también

Referencia de la notación: [Articulaciones y ornamentos], página 128, [Indicación de los números de cuerda], página 369, [Ligaduras de expresión], página 140.

## Armónicos

### *Armónicos naturales*

La notación de los armónicos naturales se puede realizar de varias formas. Generalmente, una nota con la cabeza en forma de rombo significa tocar (sin apretar) la cuerda en el lugar donde se pisaría la nota si no fuese un rombo.

```
\relative d' {
  d4 e4.
```

```

\harmonicsOn
d8 e e
d4 e4.
\harmonicsOff
d8 e e
}

```



Como posibilidad alternativa está la de mostrar una cabeza normal de nota en la altura de la nota que debe sonar, con un pequeño círculo que indica que se debe tocar como armónico:

```

d''2^\flageolet d''_\flageolet

```



Se puede hacer un círculo de menor tamaño, véase la lista de fragmentos de código en [Referencias para cuerdas sin trastes], página 365.

### *Armónicos artificiales*

La notación de los armónicos artificiales se realiza con dos notas, una con una cabeza normal que indica la posición donde se pisa, y otra con una cabeza en forma de rombo hueco para indicar la posición donde se roza la cuerda (sin pisar) para producir el armónico.

Los armónicos artificiales indicados con `\harmonic` no presentan los puntillos. La propiedad de contexto `harmonicDots` se debe activar si se necesitan los puntillos.

```

\relative e' {
  <e a\harmonic>2. <c g'\harmonic>4
  \set harmonicDots = ##t
  <e a\harmonic>2. <c g'\harmonic>4
}

```



## Véase también

Glosario musical: Sección “harmonics” en *Glosario Musical*.

Referencia de la notación: [Cabezas de nota especiales], página 41, [Referencias para cuerdas sin trastes], página 365.

## Snap (Bartók) pizzicato

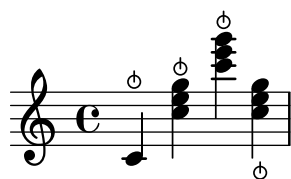
El *snap pizzicato* (conocido también como “pizzicato de Bartók”) es un tipo de pizzicato en el que la cuerda se pulsa hacia arriba (y no lateralmente) de forma deliberada, de forma que golpea la madera del diapasón.

```

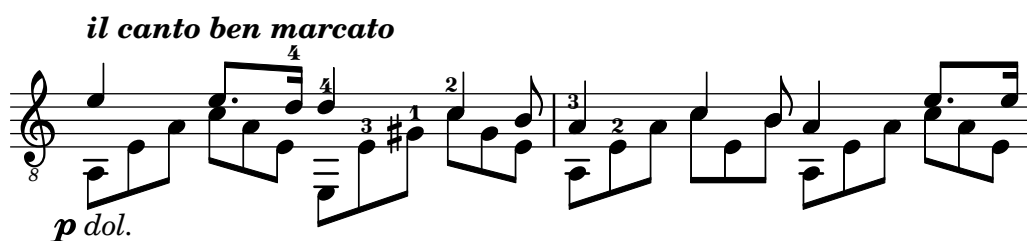
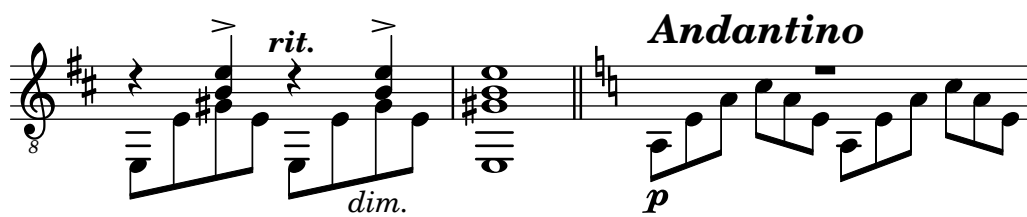
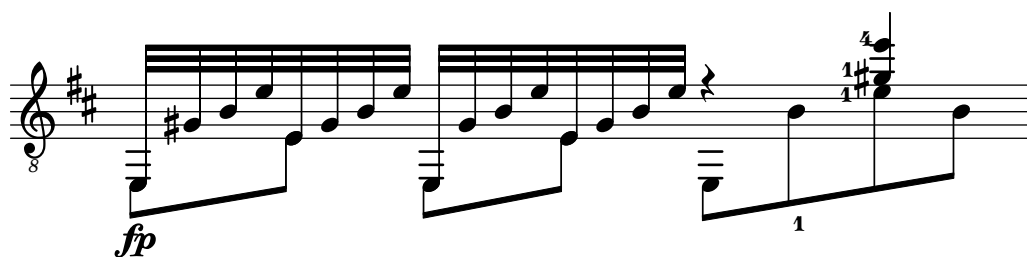
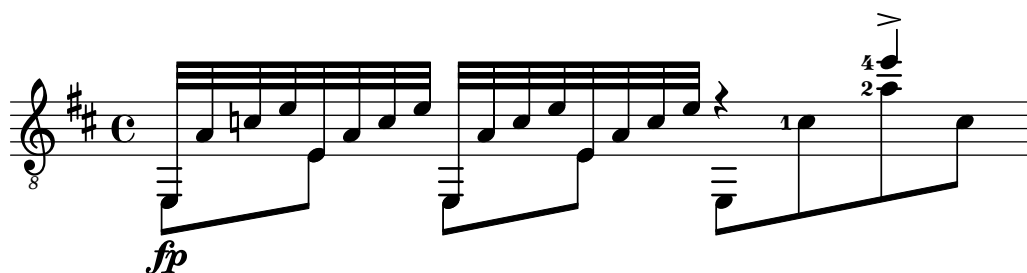
\relative {
  c'4\snappizzicato
  <c' e g>4\snappizzicato
  <c' e g>4^\snappizzicato
  <c, e g>4_\snappizzicato
}

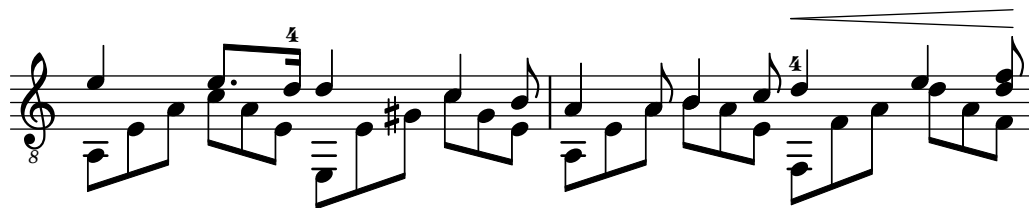
```

}



## 2.4 Instrumentos de cuerda con trastes





Esta sección trata varios aspectos de la notación musical que son exclusivos de los instrumentos de cuerda con trastes.

### 2.4.1 Notación común para cuerdas con trastes

Esta sección se ocupa de la notación común exclusiva de los instrumentos de cuerda con trastes.

#### Referencias para cuerdas con trastes

La notación musical para instrumentos de cuerda con trastes se realiza normalmente en una sola pauta, ya sea en notación musical tradicional o en tablatura. A veces se combinan los dos tipos, y es común en música popular utilizar diagramas de acordes por encima de un pentagrama de notación tradicional. La guitarra y el banjo son instrumentos transpositores, que suenan una octava por debajo de lo escrito. Las partituras para estos instrumentos deben usar la clave de Sol octava baja "treble\_8" (o `\transposition c` para obtener una salida MIDI correcta). Otros elementos relativos a los instrumentos de cuerda con trastes se estudian en otras partes de manual:

- Las digitaciones se indican como se explica en [Indicaciones de digitación], página 240.
- Las instrucciones para ligaduras *Laissez vibrer* así como ligaduras de unión sobre arpeggios y trémolos están en [Ligaduras de unión], página 58.
- Las instrucciones para el manejo de varias voces se encuentran en [Resolución de las colisiones], página 186.
- Las instrucciones para la indicación de armónicos está en [Armónicos], página 366.

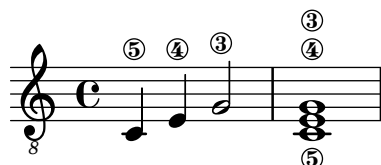
#### Véase también

Referencia de la notación: [Indicaciones de digitación], página 240, [Ligaduras de unión], página 58, [Resolución de las colisiones], página 186, [Nombres de instrumentos], página 220, [Escribir música en paralelo], página 197, [Arpeggio], página 152, Sección A.14 [Lista de articulaciones], página 800, [Clave], página 18, [Transposición de los instrumentos], página 28.

#### Indicación de los números de cuerda

Se puede indicar la cuerda en que se debe tocar una nota añadiendo `\ número` a una nota.

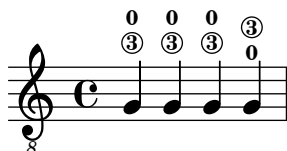
```
\clef "treble_8"
c4\5 e\4 g2\3
<c\5 e\4 g\3>1
```



Cuando se usan al mismo tiempo digitaciones e indicaciones del número de cuerda, su colocación se puede controlar por el orden en que aparecen estos elementos en el código *solamente* si aparecen dentro de un acorde explícito: aplicado a acordes completos o notas individuales *fuera* de acordes, las digitaciones se colocan usando un mecanismo distinto.

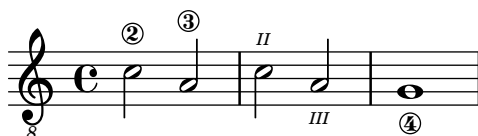
```
\clef "treble_8"
g4\3-0
```

```
g-0\3
<g\3-0>
<g-0\3>
```



Los números de cuerda también pueden (así como en los instrumentos sin trastes, donde es obligatorio) imprimirse en números romanos y colocarse bajo el pentagrama, en lugar de encima de él.

```
\clef "treble_8"
c'2\2
a\3
\romanStringNumbers
c'\2
\set stringNumberOrientations = #'(down)
a\3
\arabicStringNumbers
g1\4
```



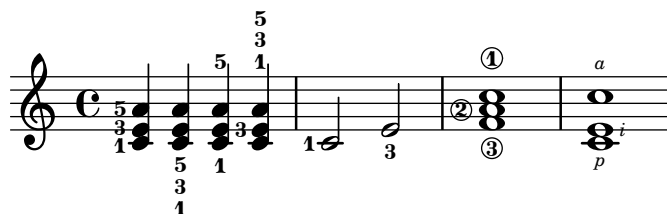
## Fragmentos de código seleccionados

### *Controlar la colocación de las digitaciones de acordes*

Se puede controlar con precisión la colocación de los números de digitación. Para que se tenga en cuenta la orientación de las digitaciones, se debe utilizar dentro de una construcción de acorde <> aunque sea una sola nota. Se puede establecer la orientación para los números de cuerda y las digitaciones de la mano derecha de una forma similar.

```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
  \set stringNumberOrientations = #'(up left down)
  <f\3 a\2 c\1>1
  \set strokeFingerOrientations = #'(down right up)
  <c\rightHandFinger #1 e\rightHandFinger #2 c'\rightHandFinger #4 >
```

}



### *Permitir que las digitaciones se impriman dentro del pentagrama*

Las cifras de digitación orientadas verticalmente se colocan de forma predeterminada fuera del pentagrama. Sin embargo, este comportamiento se puede cancelar. Hay que prestar atención en situaciones en las que las digitaciones y las plicas están en la misma dirección: por defecto, las digitaciones solo evitan la colisión con plicas unidas por una barra. Se puede cambiar este ajuste para no evitar ninguna plica o evitarlas todas; el ejemplo siguiente muestra las dos copciones, así como la manera de volver al comportamiento predeterminado.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering.staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 g'-0
  a8[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##f
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##t
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = #only-if-beamed
  a[-1 b]-2 g-0 r
}
```



### Instrucciones predefinidas

`\arabicStringNumbers`, `\romanStringNumbers`.

### Véase también

Referencia de la notación: [Indicaciones de digitación], página 240.

Fragmentos de código: Sección “Fretted strings” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “StringNumber” en *Referencia de Funcionamiento Interno*, Sección “Fingering” en *Referencia de Funcionamiento Interno*.

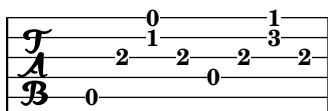
### Tablaturas predeterminadas

La notación para instrumentos de cuerda pulsada se realiza con frecuencia utilizando un sistema de dedos y cifras, o tablatura. A diferencia de la notación tradicional, las notas no se designan mediante figuras, sino mediante números (o símbolos semejantes a letras en la intavolatura histórica). Las líneas de la tablatura indican sobre qué cuerda se debe tocar la nota, y el número sobre una línea indica el traste en que la cuerda correspondiente debe pisarse. Las notas que se deben tocar simultáneamente se alinean en sentido vertical.



De forma predeterminada, la cuerda 1 es la más aguda y corresponde a la línea superior de la pauta `TabStaff`. La afinación predeterminada de las cuerdas del `TabStaff` es la afinación estándar de guitarra (con 6 cuerdas). Las notas se imprimen como tablatura, usando los contextos `TabStaff` y `TabVoice`. Se añade automáticamente una clave caligráfica de tablatura.

```
\new TabStaff \relative {
  a,8 a' <c e> a
  d,8 a' <d f> a
}
```



Las tablaturas predeterminadas no contienen símbolos para la duración de las notas ni ningún otro símbolo musical, como p.ej. indicaciones expresivas.

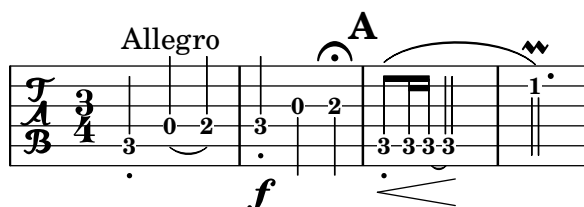
```
symbols = {
  \time 3/4
  c4-.~"Allegro" d( e)
  f4-. \f g a~\fermata
  \mark \default
  c8_.\<\( c16 c~ 2\!
  c'2.\prall\
}

\score {
  <<
    \new Staff { \clef "G_8" \symbols }
    \new TabStaff { \symbols }
  >>
}
```

Si queremos que todos los símbolos musicales que se usan en la notación tradicional aparezcan también en la tablatura, tenemos que aplicar la instrucción `\tabFullNotation` dentro de un contexto `TabStaff`. Tenga en cuenta que, en la tablatura, las blancas tienen doble plica para poder distinguirlas de las negras.

```
symbols = {
  \time 3/4
  c4-.~"Allegro" d( e)
  f4-. \f g a~\fermata
  \mark \default
  c8_.\<\( c16 c~ 2\!
  c'2.\prall\
}
```

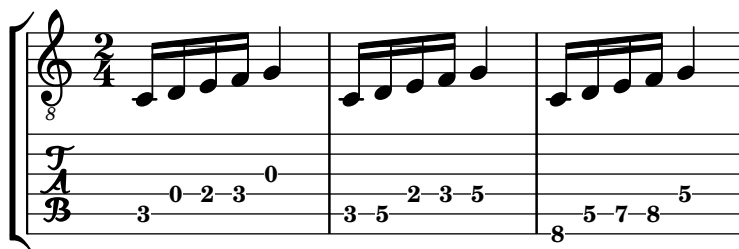
```
\score {
  \new TabStaff {
    \tabFullNotation
    \symbols
  }
}
```



De forma predeterminada, las notas se asignan a la posición más baja del mástil (primera posición). Las cuerdas al aire se toman como preferencia, automáticamente. Si queremos que una nota determinada se toque sobre una cuerda concreta, podemos añadir una indicación de número de cuerda al nombre de la nota. Si no queremos que las indicaciones de número de cuerda aparezcan en la notación tradicional, podemos sobrescribir el sello correspondiente. Generalmente es mucho más cómodo definir la posición utilizando el valor de `minimumFret`. El valor predeterminado de es 0.

Incluso si `minimumFret` está establecido, se usan las cuerdas al aire siempre que sea posible. Este comportamiento se puede cambiar fijando `restrainOpenStrings` al valor `#t`.

```
\layout { \omit Voice.StringNumber }
\new StaffGroup <<
  \new Staff \relative {
    \clef "treble_8"
    \time 2/4
    c16 d e f g4
    c,16\5 d\5 e\4 f\4 g4\4
    c,16 d e f g4
  }
  \new TabStaff \relative {
    c16 d e f g4
    c,16\5 d\5 e\4 f\4 g4\4
    \set TabStaff.minimumFret = #5
    \set TabStaff.restrainOpenStrings = ##t
    c,16 d e f g4
  }
>>
```



Las construcciones de acorde se pueden repetir mediante el símbolo de repetición de acordes `q`. En combinación con las tablaturas, su comportamiento de eliminar números de cuerda y de dedos junto a otros eventos es fastidioso, por lo que quizá quiera ejecutar

```
\chordRepeats #'(string-number-event fingering-event)
```

explícitamente sobre expresiones musicales en las tablaturas que utilicen [Repetición de acordes], página 178. Esta instrucción en particular es tan común que está disponible como `\tabChordRepeats`.

```
guitar = \relative {
  r8 <gis-2 cis-3 b-0>~ q4 q8~ 8 q4
}
```

```
\new StaffGroup <<
  \new Staff {
    \clef "treble_8"
    \guitar
  }
  \new TabStaff {
    \tabChordRepeats \guitar
  }
>>
```

Las ligaduras de unión que cruzan saltos de línea llevan paréntesis de forma predeterminada. Lo mismo rige para la casilla de segunda vez de una repetición.

```
ties = \relative {
  \repeat volta 2 {
    e'2. f4~
    2 g2~
  }
  \alternative {
    { g4 f2. }
    { g4\repeatTie c,2. }
  }
  b1~
  \break
  b1
  \bar " | ."
}
```

```
\score {
  <<
    \new StaffGroup <<
      \new Staff {
        \clef "treble_8"
        \ties
      }
      \new TabStaff {
```

```

        \ties
      }
    >>
  >>
  \layout {
    indent = #0
    ragged-right = ##t
  }
}

```

La instrucción `\hideSplitTiedTabNotes` cancela el comportamiento de imprimir los números de traste entre paréntesis:

```

ties = \relative {
  \repeat volta 2 {
    e'2. f4~
    2 g2~ }
  \alternative {
    { g4 f2. }
    { g4\repeatTie c,2. }
  }
  b1~
  \break
  b1
  \bar "|"
}

\score {
  <<
  \new StaffGroup <<
  \new Staff {
    \clef "treble_8"
    \ties
  }
  \new TabStaff {

```

```

\hideSplitTiedTabNotes
\ties
}
>>
>>
\layout {
indent = #0
ragged-right = ##t
}
}

```

Se pueden añadir indicaciones de armónicos a la notación de tablatura como sus alturas de sonido:

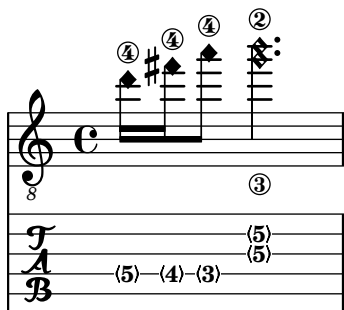
```

\layout { \omit Voice.StringNumber }
firstHarmonic = {
d'4\4\harmonic
g'4\3\harmonic
b'2\2\harmonic
}
\score {
<<
\new Staff {
\clef "treble_8"
\firstHarmonic
}
\new TabStaff { \firstHarmonic }
>>
}

```

Observe que la instrucción `\harmonic` se debe adjuntar siempre a notas individuales (posiblemente dentro de un acorde) en lugar de a acordes completos. Sólo tiene sentido para armónicos sobre una cuerda abierta en el duodécimo traste. Cualquier otro armónico debe estar calculado por parte de LilyPond. Se puede conseguir esto indicando el traste en que debe rozar la cuerda el dedo de la mano que pisa.

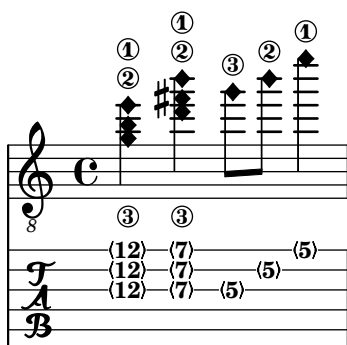
```
fretHarmonics = {
  \harmonicByFret #5 d16\4
  \harmonicByFret #4 d16\4
  \harmonicByFret #3 d8\4
  \harmonicByFret #5 <g\3 b\2>2.
}
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \fretHarmonics
    }
    \new TabStaff { \fretHarmonics }
  >>
}
```



De forma alternativa, se pueden calcular los armónicos definiendo la razón de las longitudes de cuerda por encima y por debajo de la digitación de armónico.

```
ratioHarmonics = {
  \harmonicByRatio #1/2 <g\3 b\2 e'\1>4
  \harmonicByRatio #1/3 <g\3 b\2 e'\1>4
  \harmonicByRatio #1/4 { g8\3 b8\2 e'4\1 }
}
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \ratioHarmonics
    }
    \new TabStaff { \ratioHarmonics }
  >>
}
```

}

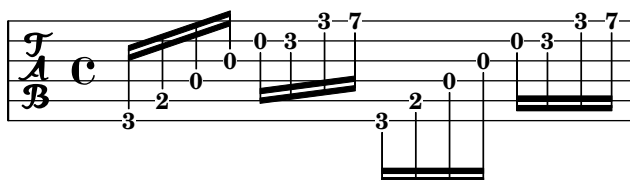


## Fragmentos de código seleccionados

### *Comportamiento de las plicas y las barras de corchea en tablaturas*

La dirección de las plicas se controla de la misma forma en la tablatura que en la notación tradicional. Las barras se pueden poner horizontales, como se muestra en este ejemplo.

```
\new TabStaff {
  \relative c {
    \tabFullNotation
    g16 b d g b d g b
    \stemDown
    \override Beam.concaveness = #10000
    g,,16 b d g b d g b
  }
}
```



### *Polifonía en tablaturas*

La polifonía se crea de la misma forma en un TabStaff que en una pauta normal.

```
upper = \relative c' {
  \time 12/8
  \key e \minor
  \voiceOne
  r4. r8 e, fis g16 b g e e' b c b a g fis e
}

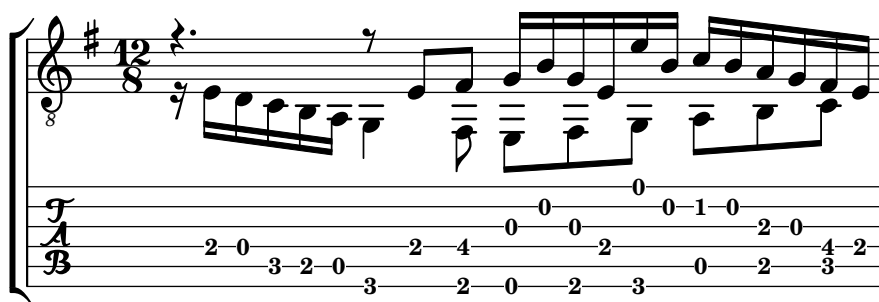
lower = \relative c {
  \key e \minor
  \voiceTwo
  r16 e d c b a g4 fis8 e fis g a b c
}

\score {
  <<
```

```

\new StaffGroup = "tab with traditional" <<
  \new Staff = "guitar traditional" <<
    \clef "treble_8"
    \new Voice = "upper" \upper
    \new Voice = "lower" \lower
  >>
  \new TabStaff = "guitar tab" <<
    \new TabVoice = "upper" \upper
    \new TabVoice = "lower" \lower
  >>
>>
}

```



### *Referencia para armónicos sobre cuerdas al aire*

Referencia para armónicos sobre cuerdas al aire (armónicos naturales):

```

openStringHarmonics = {
  \textSpannerDown
  \override TextSpanner.staff-padding = #3
  \override TextSpanner.dash-fraction = #0.3
  \override TextSpanner.dash-period = #1

  %first harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "1st harm. "
  \harmonicByFret #12 e,2\6\startTextSpan
  \harmonicByRatio #1/2 e,\6\stopTextSpan

  %second harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "2nd harm. "
  \harmonicByFret #7 e,\6\startTextSpan
  \harmonicByRatio #1/3 e,\6
  \harmonicByFret #19 e,\6
  \harmonicByRatio #2/3 e,\6\stopTextSpan
  %\harmonicByFret #19 < e,\6 a,\5 d\4 >
  %\harmonicByRatio #2/3 < e,\6 a,\5 d\4 >

  %third harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "3rd harm. "
  \harmonicByFret #5 e,\6\startTextSpan

```



```

\harmonicByRatio #1/4 e,\6
\harmonicByFret #24 e,\6
\harmonicByRatio #3/4 e,\6\stopTextSpan
\break

%fourth harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "4th harm. "
\harmonicByFret #4 e,\6\startTextSpan
\harmonicByRatio #1/5 e,\6
\harmonicByFret #9 e,\6
\harmonicByRatio #2/5 e,\6
\harmonicByFret #16 e,\6
\harmonicByRatio #3/5 e,\6\stopTextSpan

%fifth harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "5th harm. "
\harmonicByFret #3 e,\6\startTextSpan
\harmonicByRatio #1/6 e,\6\stopTextSpan
\break

%sixth harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "6th harm. "
\harmonicByFret #2.7 e,\6\startTextSpan
\harmonicByRatio #1/7 e,\6\stopTextSpan

%seventh harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "7th harm. "
\harmonicByFret #2.3 e,\6\startTextSpan
\harmonicByRatio #1/8 e,\6\stopTextSpan

%eighth harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "8th harm. "
\harmonicByFret #2 e,\6\startTextSpan
\harmonicByRatio #1/9 e,\6\stopTextSpan
}

\score {
  <<
    \new Staff
    \with { \omit StringNumber } {
      \new Voice {
        \clef "treble_8"
        \openStringHarmonics
      }
    }
  }
  \new TabStaff {
    \new TabVoice {

```

```

\openStringHarmonics
}
}
>>
}

```

1st harm. 2nd harm. 3rd harm.

4th harm. 5th harm.

6th harm. 7th harm. 8th harm.

### *Armónicos sobre cuerdas pisadas en tablatura*

Armónicos sobre cuerdas pisadas (armónicos artificiales):

```

pinchedHarmonics = {
  \textSpannerDown
  \override TextSpanner.bound-details.left.text =
    \markup { \halign #-0.5 \teeny "PH" }
  \override TextSpanner.style =
    #'dashed-line
  \override TextSpanner.dash-period = #0.6
  \override TextSpanner.bound-details.right.attach-dir = #1
  \override TextSpanner.bound-details.right.text =
    \markup { \draw-line #'(0 . 1) }
  \override TextSpanner.bound-details.right.padding = #-0.5
}

harmonics = {
  %artificial harmonics (AH)
  \textLengthOn
  <\parenthesize b b'\harmonic>4_\markup { \teeny "AH 16" }

```

```

<\parenthesize g g'\harmonic>4_\markup { \teeny "AH 17" }
<\parenthesize d' d''\harmonic>2_\markup { \teeny "AH 19" }
%pinched harmonics (PH)
\pinchedHarmonics
<a'\harmonic>2\startTextSpan
<d''\harmonic>4
<e'\harmonic>4\stopTextSpan
%tapped harmonics (TH)
<\parenthesize g\4 g'\harmonic>4_\markup { \teeny "TH 17" }
<\parenthesize a\4 a'\harmonic>4_\markup { \teeny "TH 19" }
<\parenthesize c'\3 c''\harmonic>2_\markup { \teeny "TH 17" }
%touch harmonics (TCH)
a4( <e'\harmonic>2. )_\markup { \teeny "TCH" }
}

frettedStrings = {
  %artificial harmonics (AH)
  \harmonicByFret #4 g4\3
  \harmonicByFret #5 d4\4
  \harmonicByFret #7 g2\3
  %pinched harmonics (PH)
  \harmonicByFret #7 d2\4
  \harmonicByFret #5 d4\4
  \harmonicByFret #7 a4\5
  %tapped harmonics (TH)
  \harmonicByFret #5 d4\4
  \harmonicByFret #7 d4\4
  \harmonicByFret #5 g2\3
  %touch harmonics (TCH)
  a4 \harmonicByFret #9 g2.\3
}

\score {
  <<
    \new Staff
    \with { \omit StringNumber } {
      \new Voice {
        \clef "treble_8"
        \harmonics
      }
    }
    \new TabStaff {
      \new TabVoice {
        \frettedStrings
      }
    }
  >>
}

```

}

### *Deslizamientos en tablatura*

Los deslizamientos se pueden componer tipográficamente tanto en los contextos de **Staff** como en los de **TabStaff**:

```
slides = {
  c'8\3(\glissando d'8\3)
  c'8\3\glissando d'8\3
  \hideNotes
  \grace { g16\glissando }
  \unhideNotes
  c'4\3
  \afterGrace d'4\3\glissando {
    \stemDown \hideNotes
    g16 }
  \unhideNotes
}

\score {
  <<
    \new Staff { \clef "treble_8" \slides }
    \new TabStaff { \slides }
  >>
  \layout {
    \context {
      \Score
      \override Glissando.minimum-length = #4
      \override Glissando.springs-and-rods =
        #ly:spanner::set-spacing-rods
      \override Glissando.thickness = #2
      \omit StringNumber
      % or:
      %\override StringNumber.stencil = ##f
    }
  }
}
```

}

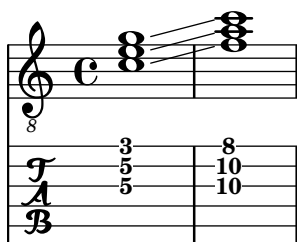
## Glissando de acordes en tablatura

Los deslizamientos para acordes se indican por omisión tanto en el contexto **Staff** como en **TabStaff**. Los números de cuerda son necesarios para **TabStaff** porque los cálculos de cuerda automáticos son diferentes para los acordes y para notas sueltas.

```
myMusic = \relative c' {
  <c e g>1 \glissando <f a c>
}

\score {
  <<
    \new Staff {
      \clef "treble_8"
      \myMusic
    }
    \new TabStaff \myMusic
  >>
}

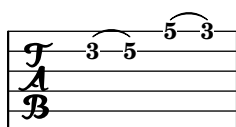
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \myMusic
    }
    \new TabStaff \with { \override Glissando.style = #'none } {
      \myMusic
    }
  >>
}
```



### *Hammer on and pull off*

Hammer-on and pull-off can be obtained using slurs.

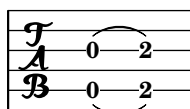
```
\new TabStaff {
  \relative c' {
    d4( e\2)
    a( g)
  }
}
```



### *Hammer on and pull off using voices*

The arc of hammer-on and pull-off is upwards in voices one and three and downwards in voices two and four:

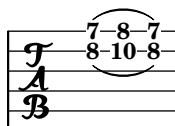
```
\new TabStaff {
  \relative c' {
    << { \voiceOne g2( a) }
    \\ { \voiceTwo a,( b) }
    >> \oneVoice
  }
}
```



### *Hammer on and pull off using chords*

When using hammer-on or pull-off with chorded notes, only a single arc is drawn. However “double arcs” are possible by setting the `doubleSlurs` property to `##t`.

```
\new TabStaff {
  \relative c' {
    % chord hammer-on and pull-off
    \set doubleSlurs = ##t
    <g' b>8( <a c> <g b>)
  }
}
```



## Véase también

Referencia de la notación: [Repetición de acordes], página 178, [Repeticiones explícitas], página 168, [Plicas], página 245, [Armónicos], página 366, [Glissando], página 147.

Fragmentos de código: Sección “Fretted strings” en *Fragmentos de código*.

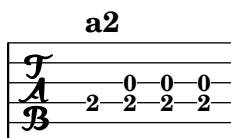
Referencia de funcionamiento interno: Sección “TabNoteHead” en *Referencia de Funcionamiento Interno*, Sección “TabStaff” en *Referencia de Funcionamiento Interno*, Sección “TabVoice” en *Referencia de Funcionamiento Interno*, Sección “Beam” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Los acordes no se tratan de una forma especial, y de aquí que el selector automático de la cuerda puede elegir fácilmente la misma cuerda para dos notas del acorde.

Para manejar `\partCombine`, es necesario que `TabStaff` utilice voces especialmente creadas:

```
melodia = \partCombine { e4 g g g } { e4 e e e }
<<
  \new TabStaff <<
    \new TabVoice = "one" s1
    \new TabVoice = "two" s1
    \new TabVoice = "shared" s1
    \new TabVoice = "solo" s1
    { \melodia }
  >>
>>
```



Los efectos especiales de guitarra se limitan a armónicos y *slides*.

## Tablaturas personalizadas

La tablatura en LilyPond calcula automáticamente el traste para cada nota, basándose en la cuerda a que está asignada la nota. Para hacerlo, es necesario especificar la afinación de las cuerdas. La afinación de las cuerdas se da en la propiedad `stringTunings`.

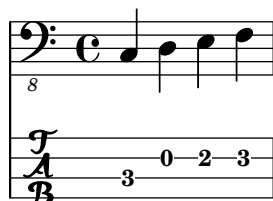
LilyPond trae afinaciones predefinidas para el banjo, la mandolina, la guitarra, el bajo, el ukelele, el violín, la viola, el violoncello y el contrabajo. LilyPond establece automáticamente la transposición correcta para las afinaciones predefinidas. El ejemplo siguiente es para bajo, que suena una octava por debajo de lo escrito.

```
<<
  \new Voice \with {
    \omit StringNumber
  } {
    \clef "bass_8"
    \relative {
      c,4 d e f
    }
  }
  \new TabStaff \with {
    stringTunings = #bass-tuning
```

```

} {
  \relative {
    c,4 d e f
  }
}
>>

```



La afinación por omisión es `guitar-tuning` (la afinación estándar Mi-La-Re-Sol-Si-Mi). Otras afinaciones predefinidas son `guitar-open-g-tuning` (Sol Mayor al aire, Re-Sol-Re-Sol-Si-Re), `mandolin-tuning` (mandolina) y `banjo-open-g-tuning` (banjo con Sol Mayor al aire). Las afinaciones predefinidas están en `ly/string-tunings-init.ly`.

Es posible crear cualquier afinación de cuerdas que se desee. Se puede usar la función `\stringTuning` para definir una afinación de cuerdas y que puede usarse para establecerla como el valor de `stringTunings` para el contexto actual.

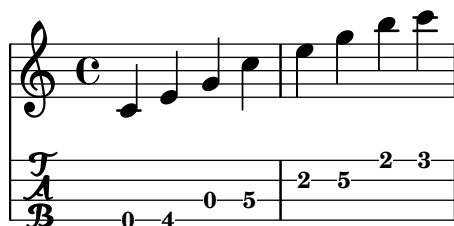
Su argumento es una construcción de acorde que define la altura de las notas de cada cuerda al aire en dicha afinación. La construcción de acorde debe estar en el modo de octava absoluto, véase [Escritura de octava absoluta], página 1. La cuerda con el número más alto (generalmente la más grave) debe ir primero en el acorde. Por ejemplo, podemos definir una afinación para un instrumento de cuatro cuerdas con las alturas `a''`, `d''`, `g'` y `c'`:

```

mynotes = {
  c'4 e' g' c'' |
  e''4 g'' b'' c'''
}

<<
\new Staff {
  \clef treble
  \mynotes
}
\new TabStaff {
  \set Staff.stringTunings = \stringTuning <c' g' d'' a''>
  \mynotes
}
>>

```



La propiedad `stringTunings` también se utiliza por parte de `FretBoards` para calcular los diagramas de posición automáticos.



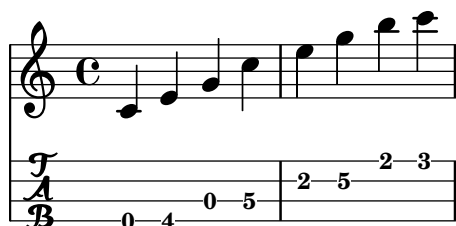
Las afinaciones de cuerdas se utilizan como parte de la clave de asociación para los diagramas de posición predefinidos (véase [Diagramas predefinidos de trastes], página 400).

El ejemplo anterior también se podría haber escrito como sigue:

```
custom-tuning = \stringTuning <c' g' d'' a''>

mynotes = {
  c'4 e' g' c'' |
  e''4 g'' b'' c'''
}

<<
\new Staff {
  \clef treble
  \mynotes
}
\new TabStaff {
  \set TabStaff.stringTunings = #custom-tuning
  \mynotes
}
>>
```



Internamente, una afinación de las cuerdas es una lista de Scheme de alturas, una por cada cuerda, ordenadas de 1 a N, donde la cuerda 1 está arriba en la pauta de tablatura y la cuerda N está abajo. Esto resulta generalmente en un orden de la nota más aguda a la más grave, pero algunos instrumentos (p.ej. el ukelele) no tienen las cuerdas dispuestas en orden de altura.

La altura de una cuerda dentro de una lista de alturas de cuerdas es un objeto altura de LilyPond. Los objetos altura se crean con la función de Scheme `ly:make-pitch` (véase Sección A.22 [Funciones de Scheme], página 859).

`\stringTuning` crea dicho objeto a partir de la entrada de acordes.

LilyPond calcula automáticamente el número de líneas dentro del `TabStaff` y el número de cuerdas dentro de un `FretBoard` calculado automáticamente como el número de elementos de `stringTunings`.

Para permitir a todos los contextos `TabStaff` usar la misma afinación personalizada por omisión, podemos usar

```
\layout {
  \context {
    \TabStaff
    stringTunings = \stringTuning <c' g' d'' a''>
  }
}
```

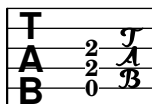
También se puede usar una clave moderna de tablatura.

```
\new TabStaff {
  \clef moderntab
```

```

<a, e a>1
\break
\clef tab
<a, e a>1
}

```



2



La clave moderna de tablatura contempla tablaturas de 4 a 7 cuerdas.

Una pauta de tablatura `TabStaff` puede contener intervallos microtonales como cuartos de tono, que se pueden reproducir mediante eventos de la rueda de cambio de tono. Es necesario fijar el valor `supportNonIntegerFret = ##t` dentro del contexto `Score`. Sin embargo, los microtonos no están contemplados en los diagramas `FretBoards`.

```

\layout {
  \context {
    \Score
    supportNonIntegerFret = ##t
  }
}

```

```

custom-tuning = \stringTuning <e, a, d ges beh eeh'>

```

```

mus = \relative {
  eeses'4
  eeseh
  ees
  eeh
  e
  eih
  eis
  eisih
  eisis
}

```

```

<<
  \new Staff << \clef "G_8" \mus >>
  \new TabStaff \with { stringTunings = \custom-tuning } \mus
>>

```



## Véase también

Referencia de la notación: [Escritura de octava absoluta], página 1, [Diagramas predefinidos de trastes], página 400, Sección A.22 [Funciones de Scheme], página 859.

Archivos instalados: `ly/string-tunings-init.ly`, `scm/tablatute.scm`.

Fragmentos de código: Sección “Fretted strings” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Tab\_note\_heads\_engraver” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

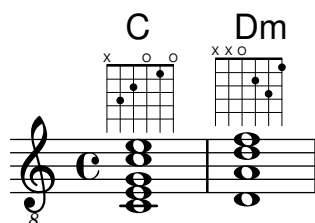
Los cálculos de tablatura automáticos no funcionan bien en la mayoría de los casos para instrumentos en que la afinación de las cuerdas no varía de forma monótona con el número de cuerda, como el ukelele.

## Marcas de diagramas de trastes

Se pueden añadir diagramas de trastes a la música como elementos de marcado sobre la nota deseada. El marcado contiene información sobre el diagrama de trastes deseado. Existen tres interfaces distintos de marcado de diagramas de trastes: `standard` (estándar), `terse` (escueto) y `verbose` (prolijo). Los tres interfaces producen marcados equivalentes, pero tienen cantidades variables de información en la cadena de marcado. Hay más detalles sobre la sintaxis de las distintas cadenas de marcado que se usan para definir diagramas de posiciones en Sección A.11.5 [Instrument Specific Markup], página 780.

La cadena de marcado de los diagramas estándar de trastes indica el número de la cuerda y el número de traste en que se coloca cada uno de los puntos sobre la cuerda. Además se pueden indicar cuerdas al aire y cuerdas mudas (que no se tocan).

```
<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram "6-x;5-3;4-2;3-o;2-1;1-o;"
    }
    <d a d' f'>1^\markup {
      \fret-diagram "6-x;5-x;4-o;3-2;2-3;1-1;"
    }
  }
  }
>>
```



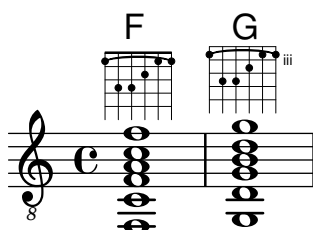
Las indicaciones de cejilla se pueden añadir al diagrama a partir de la cadena de marcado del diagrama de trastes.

```
<<
```

```

\new ChordNames {
  \chordmode {
    f1 g
  }
}
\new Staff {
  \clef "treble_8"
  <f, c f a c' f'>1^\markup {
    \fret-diagram "c:6-1-1;6-1;5-3;4-3;3-2;2-1;1-1;"
  }
  <g, d g b d' g'>1^\markup {
    \fret-diagram "c:6-1-3;6-3;5-5;4-5;3-4;2-3;1-3;"
  }
}
}
>>

```

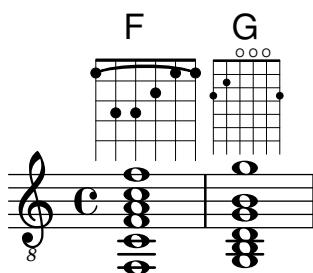


El tamaño del diagrama de trastes y el número de trastes del diagrama, se puede cambiar en la cadena de marcado `fret-diagram`.

```

<<
\new ChordNames {
  \chordmode {
    f1 g
  }
}
\new Staff {
  \clef "treble_8"
  <f, c f a c' f'>1^\markup {
    \fret-diagram "s:1.5;c:6-1-1;6-1;5-3;4-3;3-2;2-1;1-1;"
  }
  <g, b, d g b g'>1^\markup {
    \fret-diagram "h:6;6-3;5-2;4-o;3-o;2-o;1-3;"
  }
}
}
>>

```

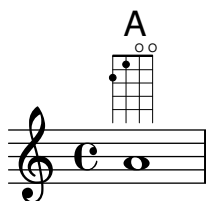


Se puede cambiar el número de cuerdas de un diagrama de trastes para que se adapte a distintos instrumentos como el bajo y el ukelele, con la cadena de marcado del diagrama.

```

<<
  \new ChordNames {
    \chordmode {
      a1
    }
  }
  \new Staff {
    % An 'A' chord for ukulele
    a'1^\markup {
      \fret-diagram "w:4;4-2-2;3-1-1;2-o;1-o;"
    }
  }
>>

```

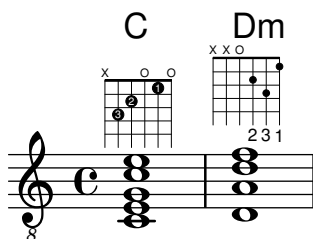


Se pueden añadir indicaciones de digitación, y la posición de las etiquetas de los dedos se puede controlar mediante la cadena de marcado del diagrama.

```

<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram "f:1;6-x;5-3-3;4-2-2;3-o;2-1-1;1-o;"
    }
    <d a d' f'>1^\markup {
      \fret-diagram "f:2;6-x;5-x;4-o;3-2-2;2-3-3;1-1-1;"
    }
  }
>>

```



Se puede controlar el radio y la posición de los puntos con la cadena de marcado fret-diagram.

```

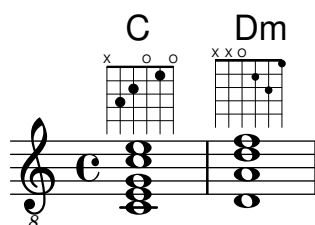
<<
  \new ChordNames {
    \chordmode {

```

```

    c1 d:m
  }
}
\new Staff {
  \clef "treble_8"
  <c e g c' e'>1^\markup {
    \fret-diagram "d:0.35;6-x;5-3;4-2;3-o;2-1;1-o;"
  }
  <d a d' f'>1^\markup {
    \fret-diagram "p:0.2;6-x;5-x;4-o;3-2;2-3;1-1;"
  }
}
}
>>

```



Se pueden imprimir diagramas de trastes de acordes invertidos para personas zurdas.

```

\markup
\center-column {
  "C"
  "(left-handed)"
  \override #`{fret-diagram-details . ((handedness . ,LEFT))}
  \fret-diagram "6-x;5-3-3;4-2-2;3-o;2-1;1-o;"
}

```

C  
(left-handed)



La cadena de marcado de diagrama de trastes escueta, **fret-diagram-terse**, omite los números de la cuerda; el número de cuerda viene implícito por la presencia del punto y coma. Hay un punto y coma por cada cuerda del diagrama. El primer punto y coma corresponde al número de cuerda más alto, y el último punto y coma corresponde a la primera cuerda. Se pueden indicar cuerdas mudas, al aire y números de traste.

```

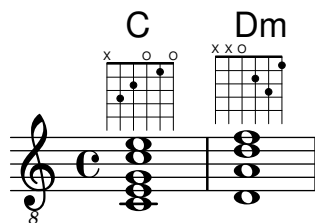
<<
\new ChordNames {
  \chordmode {
    c1 d:m
  }
}
\new Staff {
  \clef "treble_8"
  <c e g c' e'>1^\markup {
    \fret-diagram-terse "x;3;2;o;1;o;"
  }
}

```

```

<d a d' f'>1^\markup {
  \fret-diagram-terse "x;x;o;2;3;1;"
}
}
>>

```

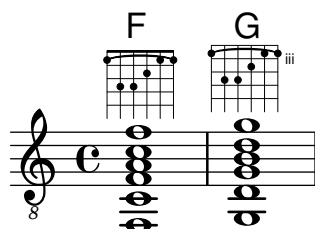


Se pueden incluir números de cejilla en la cadena de marcado escueta `fret-diagram-terse`.

```

<<
  \new ChordNames {
    \chordmode {
      f1 g
    }
  }
  \new Staff {
    \clef "treble_8"
    <f, c f a c' f'>1^\markup {
      \fret-diagram-terse "1-(;3;3;2;1;1-);"
    }
    <g, d g b d' g'>1^\markup {
      \fret-diagram-terse "3-(;5;5;4;3;3-);"
    }
  }
}
>>

```



Se pueden incluir indicaciones de digitación en la cadena de marcado escueta `fret-diagram-terse`

```

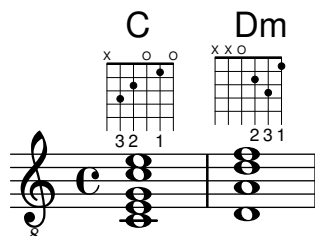
<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new Staff {
    \override Voice.TextScript.fret-diagram-details.finger-code = #'below-string
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram-terse "x;3-3;2-2;o;1-1;o;"
    }
  }
}

```

```

<d a d' f'>1^\markup {
  \fret-diagram-terse "x;x;o;2-2;3-3;1-1;"
}
}
>>

```



Otras propiedades de los diagramas de trastes se deben ajustar usando `\override` al utilizar el marcado escueto con `fret-diagram-terse`.

Solo se puede incluir una indicación por cada cuerda en un marcado del tipo `fret-diagram-terse`. Para tener más de una indicación por cuerda, utilice un marcado de diagrama normal o del tipo prolijo.

La cadena de marcado prolija `fret-diagram-verbose` está en el formato de una lista de Scheme. Cada elemento de la lista indica la colocación un elemento sobre el diagrama de trastes.

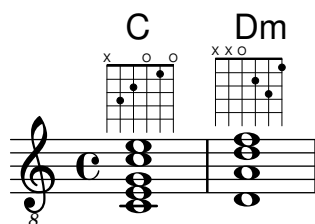
```

<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram-verbose #'(
        (mute 6)
        (place-fret 5 3)
        (place-fret 4 2)
        (open 3)
        (place-fret 2 1)
        (open 1)
      )
    }
    <d a d' f'>1^\markup {
      \fret-diagram-verbose #'(
        (mute 6)
        (mute 5)
        (open 4)
        (place-fret 3 2)
        (place-fret 2 3)
        (place-fret 1 1)
      )
    }
  }
}

```



&gt;&gt;



Se pueden incluir digitaciones y cejillas en una cadena de marcado prolija `fret-diagram-verbose`. Es exclusiva del interfaz de `fret-diagram-verbose` la indicación ‘capo’ que se puede colocar sobre el diagrama de posición. La indicación de capo es una barra gruesa que cubre todas las cuerdas. El traste que tiene el capo es el más bajo del diagrama de posición.

Los puntos que indican las digitaciones se pueden tanto colorear como imprimir entre paréntesis; asimismo, es posible alterar independientemente el color de los paréntesis.

También se pueden insertar elementos de marcado dentro de los puntos.

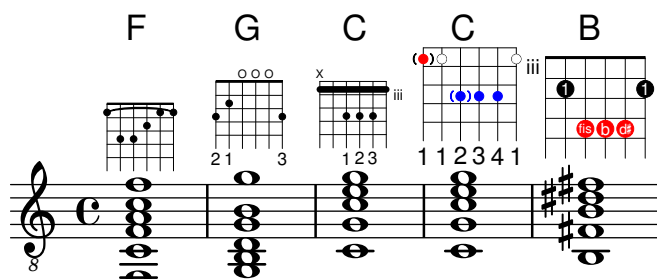
&lt;&lt;

```
\new ChordNames {
  \chordmode {
    f1 g c c b
  }
}
\new Staff {
  \clef "treble_8"
  \override Voice.TextScript
    .fret-diagram-details.finger-code = #'below-string
  <f, c f a c' f'>1^\markup {
    \fret-diagram-verbose #'(
      (place-fret 6 1)
      (place-fret 5 3)
      (place-fret 4 3)
      (place-fret 3 2)
      (place-fret 2 1)
      (place-fret 1 1)
      (barre 6 1 1)
    )
  }
  <g, b, d g b g'>1^\markup {
    \fret-diagram-verbose #'(
      (place-fret 6 3 2)
      (place-fret 5 2 1)
      (open 4)
      (open 3)
      (open 2)
      (place-fret 1 3 3)
    )
  }
  <c g c' e' g'>1^\markup {
    \fret-diagram-verbose #'(
      (capo 3)
      (mute 6)
    )
  }
}
```

```

        (place-fret 4 5 1)
        (place-fret 3 5 2)
        (place-fret 2 5 3)
    )
}
\override Voice.TextScript.size = 1.4
<c g c' e' g'>1^\markup {
  \fret-diagram-verbose #'(
    (place-fret 6 3 1 red parenthesized default-paren-color)
    (place-fret 5 3 1 inverted)
    (place-fret 4 5 2 blue parenthesized)
    (place-fret 3 5 3 blue)
    (place-fret 2 5 4 blue)
    (place-fret 1 3 1 inverted)
  )
}
\override Voice.TextScript.size = 1.5
<b, fis b dis' fis'>1^\markup
  \override #'(fret-diagram-details . ((finger-code . in-dot)))
  \fret-diagram-verbose #`(
    (place-fret 5 2 1)
    (place-fret 4 4 "fis" red)
    (place-fret 3 4 "b" red)
    (place-fret
      2 4
      ,#{ \markup
        \concat {
          \vcenter "d"
          \fontsize #-5
          \musicglyph "accidentals.sharp"} #}
      red)
    (place-fret 1 2 1)
  )
}
>>

```



Todas las otras propiedades de diagrama de trastes se deben ajustar utilizando `\override` cuando se usa el marcado prolijo con `fret-diagram-verbose`.

El aspecto gráfico de un diagrama de trastes se puede personalizar según las preferencias del usuario a través de las propiedades del interface `fret-diagram-interface`. Los detalles están en Sección “`fret-diagram-interface`” en *Referencia de Funcionamiento Interno*. Para un marcado de diagrama de trastes, las propiedades del interface pertenecen a `Voice.TextScript`.

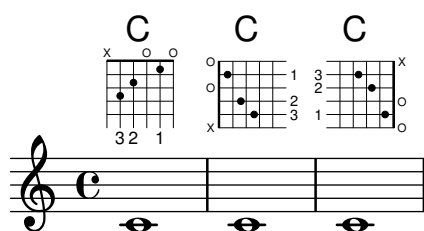
## Fragmentos de código seleccionados

### *Modificar la orientación de los trastes*

Los diagramas de posiciones de acordes se pueden orientar de tres formas. De manera predeterminada se alinea la cuerda o traste superior en las distintas alineaciones.

```
\include "predefined-guitar-fretboards.ly"

<<
\chords {
  c1
  c1
  c1
}
\new FretBoards {
  \chordmode {
    c1
    \override FretBoard.fret-diagram-details.orientation =
      #'landscape
    c1
    \override FretBoard.fret-diagram-details.orientation =
      #'opposing-landscape
    c1
  }
}
\new Voice {
  c'1
  c'1
  c'
}
>>
```



### *Personalizar diagramas de posiciones de marcado*

Se pueden establecer las propiedades de los diagramas de posiciones a través de 'fret-diagram-details. Para los diagramas de posiciones de marcado, se pueden aplicar overrides (sobrescrituras) al objeto Voice.TextScript o directamente al elemento de marcado.

```
<<
\chords { c1 | c | c | d }

\new Voice = "mel" {
  \textLengthOn
  % Set global properties of fret diagram
  \override TextScript.size = #'1.2
  \override TextScript.fret-diagram-details.finger-code = #'in-dot
}
```

```

\override TextScript.fret-diagram-details.dot-color = #'white

%% C major for guitar, no barre, using defaults
% terse style
c'1^\markup { \fret-diagram-terse "x;3-3;2-2;o;1-1;o;" }

%% C major for guitar, barred on third fret
% verbose style
% size 1.0
% roman fret label, finger labels below string, straight barre
c'1^\markup {
  % standard size
  \override #'(size . 1.0) {
    \override #'(fret-diagram-details . (
      (number-type . roman-lower)
      (finger-code . in-dot)
      (barre-type . straight))) {
      \fret-diagram-verbose #'((mute 6)
        (place-fret 5 3 1)
        (place-fret 4 5 2)
        (place-fret 3 5 3)
        (place-fret 2 5 4)
        (place-fret 1 3 1)
        (barre 5 1 3))
    }
  }
}

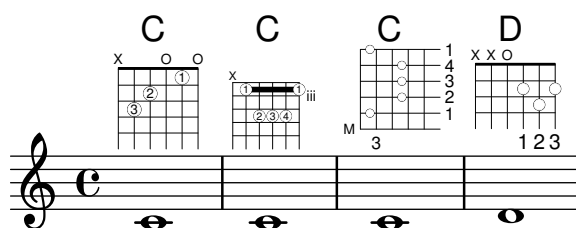
%% C major for guitar, barred on third fret
% verbose style
% landscape orientation, arabic numbers, M for mute string
% no barre, fret label down or left, small mute label font
c'1^\markup {
  \override #'(fret-diagram-details . (
    (finger-code . below-string)
    (number-type . arabic)
    (label-dir . -1)
    (mute-string . "M")
    (orientation . landscape)
    (barre-type . none)
    (xo-font-magnification . 0.4)
    (xo-padding . 0.3))) {
    \fret-diagram-verbose #'((mute 6)
      (place-fret 5 3 1)
      (place-fret 4 5 2)
      (place-fret 3 5 3)
      (place-fret 2 5 4)
      (place-fret 1 3 1)
      (barre 5 1 3))
    }
  }
}

```

```

%% simple D chord
% terse style
% larger dots, centered dots, fewer frets
% label below string
d'1^\markup {
  \override #'(fret-diagram-details . (
    (finger-code . below-string)
    (dot-radius . 0.35)
    (dot-position . 0.5)
    (fret-count . 3))) {
    \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
  }
}
}
}
>>

```



## Véase también

Referencia de la notación: Sección A.11.5 [Instrument Specific Markup], página 780.

Fragmentos de código: Sección “Fretted strings” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “fret-diagram-interface” en *Referencia de Funcionamiento Interno*.

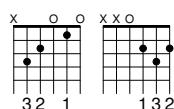
## Diagramas predefinidos de trastes

Se pueden imprimir los diagramas de trastes usando el contexto `FretBoards`. De forma predefinida, el contexto `FretBoards` imprime diagramas de trastes que están almacenados en una tabla de búsqueda:

```

\include "predefined-guitar-fretboards.ly"
\new FretBoards {
  \chordmode {
    c1 d
  }
}

```



Los diagramas de traste predefinidos están en el archivo `predefined-guitar-fretboards.ly`. Los diagramas de trastes se almacenan tomando como base las notas de un acorde y el valor de `stringTunings` que se está utilizando en cada momento. `predefined-guitar-fretboards.ly` contiene diagramas de traste predefinidos sólo para `guitar-tuning`. Los diagramas de traste predefinidos se pueden añadir para otros instrumentos u otras afinaciones siguiendo los ejemplos que aparecen en `predefined-guitar-fretboards.ly`.

Los diagramas de posiciones para el ukelele están en el archivo `predefined-ukulele-fretboards.ly`.

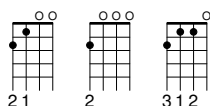
```
\include "predefined-ukulele-fretboards.ly"

myChords = \chordmode { a1 a:m a:aug }

\new ChordNames {
  \myChords
}

\new FretBoards {
  \set Staff.stringTunings = #ukulele-tuning
  \myChords
}
```

**A   Am   A+**



Los diagramas de posiciones para la mandolina están contenidos en el archivo `predefined-mandolin-fretboards.ly`.

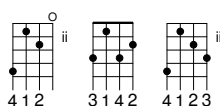
```
\include "predefined-mandolin-fretboards.ly"

myChords = \chordmode { c1 c:m7.5- c:aug }

\new ChordNames {
  \myChords
}

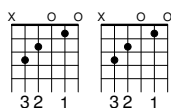
\new FretBoards {
  \set Staff.stringTunings = #mandolin-tuning
  \myChords
}
```

**C   C<sup>∅</sup>   C+**



Las notas de los acordes se pueden introducir como música simultánea o bien usando el modo de acordes (véase [Panorámica del modo de acordes], página 442).

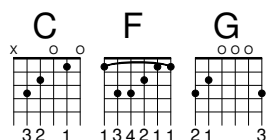
```
\include "predefined-guitar-fretboards.ly"
\new FretBoards {
  \chordmode { c1 }
  <c' e' g'>1
}
```



Es corriente que se impriman juntos los nombres de acorde y los diagramas de traste. Esto se puede hacer poniendo un contexto de nombres de acorde **ChordNames** en paralelo con un contexto de trastes **FretBoards** y dando a los dos contextos el mismo contenido musical.

```
\include "predefined-guitar-fretboards.ly"
mychords = \chordmode {
  c1 f g
}

<<
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
>>
```

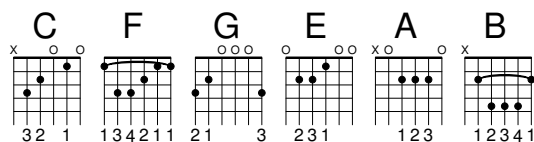


Los diagramas de traste predefinidos se pueden transportar, en la medida en que esté almacenado un diagrama para el acorde transportado, en la tabla de diagramas de traste.

```
\include "predefined-guitar-fretboards.ly"
mychords = \chordmode{
  c1 f g
}

mychordlist = {
  \mychords
  \transpose c e { \mychords }
}

<<
  \new ChordNames {
    \mychordlist
  }
  \new FretBoards {
    \mychordlist
  }
>>
```

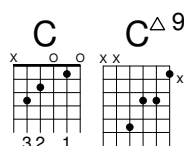


La tabla de diagramas de traste predefinidos para la guitarra contiene ocho acordes (mayor, menor, aumentado, disminuido, séptima dominante, séptima mayor, menor séptima y novena dominante) para 17 tonalidades distintas. La tabla de diagramas predefinidos para el ukelele contiene estos acordes y además otros tres (sexta mayor, segunda suspendida y cuarta suspendida). Se puede ver una lista completa de los diagramas de traste predefinidos en [Diagramas predefinidos de trastes], página 400. Si no hay una entrada en la tabla para un acorde, el grabador

FretBoards calcula un cadena de diagrama fret-diagram usando la funcionalidad de diagramas automáticos que se describe en [Diagramas de traste automáticos], página 411.

```
\include "predefined-guitar-fretboards.ly"
mychords = \chordmode {
  c1 c:maj9
}
```

```
<<
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
>>
```



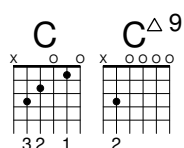
Se pueden añadir diagramas de posiciones a la tabla de diagramas de posiciones. Para añadir un diagrama debemos especificar la tabla de correspondencias para el diagrama, el acorde del diagrama, la afinación utilizada y una definición del diagrama. Normalmente la tabla de correspondencias será *default-fret-table*. La definición del diagrama puede ser una cadena de definición escueta fret-diagram-terse o una lista de marcados prolija fret-diagram-verbose.

```
\include "predefined-guitar-fretboards.ly"
```

```
\storePredefinedDiagram #default-fret-table
  \chordmode { c:maj9 }
  #guitar-tuning
  "x;3-2;o;o;o;o;"
```

```
mychords = \chordmode {
  c1 c:maj9
}
```

```
<<
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
>>
```



Se pueden almacenar distintos diagramas de traste para el mismo acorde usando distintas octavas para las notas. Las octavas de diferencia deben ser al menos dos por encima o por



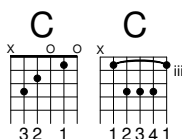
debajo de la octava predeterminada, porque las octavas inmediatas superior e inferior de la predeterminada se utilizan para los trasteros transpositores.

```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram #default-fret-table
    \chordmode { c'' }
    #guitar-tuning
    #(offset-fret 2 (chord-shape 'bes guitar-tuning))

mychords = \chordmode {
    c1 c''
}

<<
    \new ChordNames {
        \mychords
    }
    \new FretBoards {
        \mychords
    }
>>
```



Además de los diagramas de traste, LilyPond almacena una lista interna de formas de acorde. Las formas de acorde son diagramas de traste que se pueden desplazar por el mástil para dar acordes distintos. Se pueden añadir formas de acorde a la lista interna y luego usarlas para definir diagramas de posición predefinidos. Dado que se pueden mover a distintas posiciones dentro del mástil, las formas de acorde normalmente no contienen cuerdas al aire. Como los diagramas de posiciones, las formas de acorde se pueden introducir como cadenas escuetas `fret-diagram-terse` o como listas de marcado prolijas `fret-diagram-verbose`.

```
\include "predefined-guitar-fretboards.ly"

% Add a new chord shape

\addChordShape #'powerf #guitar-tuning "1-1;3-3;3-4;x;x;x;"

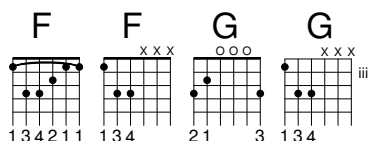
% add some new chords based on the power chord shape

\storePredefinedDiagram #default-fret-table
    \chordmode { f'' }
    #guitar-tuning
    #(chord-shape 'powerf guitar-tuning)
\storePredefinedDiagram #default-fret-table
    \chordmode { g'' }
    #guitar-tuning
    #(offset-fret 2 (chord-shape 'powerf guitar-tuning))

mychords = \chordmode {
```

```
f1 f'' g g''
}
```

```
<<
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
>>
```



El aspecto gráfico de un diagrama de traste se puede personalizar según las preferencias del usuario a través de las propiedades del interface `fret-diagram-interface`. Los detalles están en Sección “fret-diagram-interface” en *Referencia de Funcionamiento Interno*. Para un diagrama de trastes predefinido, las propiedades de interface pertenecen a `FretBoards.FretBoard`.

## Fragmentos de código seleccionados

### *Personalizar los diagramas de posiciones*

Se pueden establecer las propiedades de los diagramas de posiciones de acordes por medio de '`fret-diagram-details`'. Para los diagramas de posiciones de `FretBoard`, se aplican los overrides (sobrescrituras) al objeto `FretBoards.FretBoard`. Como `Voice`, `FretBoards` es un contexto del nivel inferior, y por tanto se puede omitir su nombre en la sobrescritura de propiedades.

```
\include "predefined-guitar-fretboards.ly"
\storePredefinedDiagram #default-fret-table \chordmode { c' }
    #guitar-tuning
    #"x;1-1-(;3-2;3-3;3-4;1-1-);"

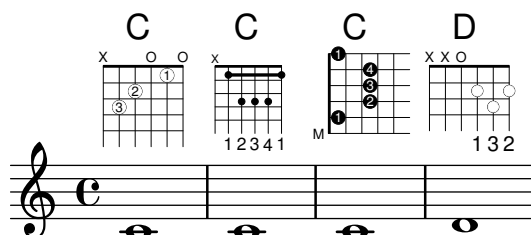
% shorthand
oo = #(define-music-function
  (grob-path value)
  (list? scheme?)
  #{ \once \override $grob-path = #value #})

<<
  \new ChordNames {
    \chordmode { c1 | c | c | d }
  }
  \new FretBoards {
    % Set global properties of fret diagram
    \override FretBoards.FretBoard.size = #'1.2
    \override FretBoard.fret-diagram-details.finger-code = #'in-dot
    \override FretBoard.fret-diagram-details.dot-color = #'white
    \chordmode {
      c
      \oo FretBoard.size #'1.0
    }
  }
>>
```

```

\oo FretBoard.fret-diagram-details.barre-type #'straight
\oo FretBoard.fret-diagram-details.dot-color #'black
\oo FretBoard.fret-diagram-details.finger-code #'below-string
c'
\oo FretBoard.fret-diagram-details.barre-type #'none
\oo FretBoard.fret-diagram-details.number-type #'arabic
\oo FretBoard.fret-diagram-details.orientation #'landscape
\oo FretBoard.fret-diagram-details.mute-string #"M"
\oo FretBoard.fret-diagram-details.label-dir #LEFT
\oo FretBoard.fret-diagram-details.dot-color #'black
c'
\oo FretBoard.fret-diagram-details.finger-code #'below-string
\oo FretBoard.fret-diagram-details.dot-radius #0.35
\oo FretBoard.fret-diagram-details.dot-position #0.5
\oo FretBoard.fret-diagram-details.fret-count #3
d
}
}
\new Voice {
  c'1 | c' | c' | d'
}
>>

```



### *Definición de posiciones predefinidas para otros instrumentos*

Se pueden añadir diagramas de posiciones predefinidas para instrumentos nuevos además de los estándar que se usan para la guitarra. Este archivo muestra cómo se hace, definiendo una afinación nueva y unas cuantas posiciones para el cuatro venezolano.

Este archivo también muestra cómo se pueden incluir las digitaciones en los acordes que se usan como puntos de referencia para la búsqueda de acordes en la tabla, y mostrarse en el diagrama de posiciones y la tablatura `TabStaff`, pero no en la música.

Estas posiciones no se pueden transportar porque contienen información de las cuerdas. Hay planes para corregir esto en un futuro.

```

% add FretBoards for the Cuatro
% Note: This section could be put into a separate file
% predefined-cuatro-fretboards.ly
% and \included into each of your compositions

cuatroTuning = #`((ly:make-pitch 0 6 0)
                  ,(ly:make-pitch 1 3 SHARP)
                  ,(ly:make-pitch 1 1 0)
                  ,(ly:make-pitch 0 5 0))

dSix = { <a\4 b\1 d\3 fis\2> }
dMajor = { <a\4 d\1 d\3 fis \2> }

```

```

aMajSeven = { <a\4 cis\1 e\3 g\2> }
dMajSeven = { <a\4 c\1 d\3 fis\2> }
gMajor = { <b\4 b\1 d\3 g\2> }

\storePredefinedDiagram #default-fret-table \dSix
    #cuatroTuning
    #"o;o;o;o;"
\storePredefinedDiagram #default-fret-table \dMajor
    #cuatroTuning
    #"o;o;o;3-3;"
\storePredefinedDiagram #default-fret-table \aMajSeven
    #cuatroTuning
    #"o;2-2;1-1;2-3;"
\storePredefinedDiagram #default-fret-table \dMajSeven
    #cuatroTuning
    #"o;o;o;1-1;"
\storePredefinedDiagram #default-fret-table \gMajor
    #cuatroTuning
    #"2-2;o;1-1;o;"

% end of potential include file /predefined-cuatro-fretboards.ly

#(set-global-staff-size 16)

primerosNames = \chordmode {
  d:6 d a:maj7 d:maj7
  g
}
primeros = {
  \dSix \dMajor \aMajSeven \dMajSeven
  \gMajor
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \primerosNames
    }

    \new Staff {
      \new Voice \with {
        \remove "New_fingering_engraver"
      }
      \relative c'' {
        \primeros
      }
    }
  }

  \new FretBoards {
    \set Staff.stringTunings = #cuatroTuning
  }
}

```

```

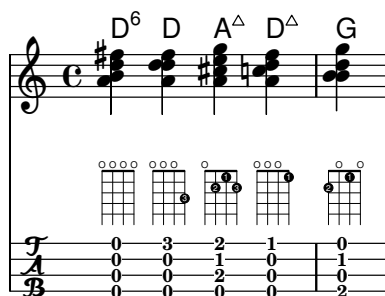
%      \override FretBoard
%      #'(fret-diagram-details string-count) = 4
%      \override FretBoard.fret-diagram-details.finger-code = #'in-dot
%      \primeros
%    }

%    \new TabStaff \relative c'' {
%      \set TabStaff.stringTunings = #cuatroTuning
%      \primeros
%    }

>>

\layout {
  \context {
    \Score
    \override SpacingSpanner.base-shortest-duration =
      #(ly:make-moment 1 16)
  }
}
\midi { }
}

```



### *Cambios de acorde de posiciones de trastes*

Se puede hacer que los diagramas de posiciones se muestren sólo cuando el acorde cambia o al comienzo de una nueva línea.

```

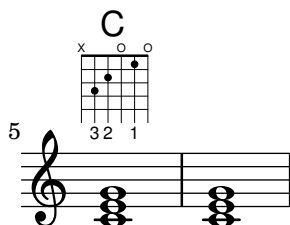
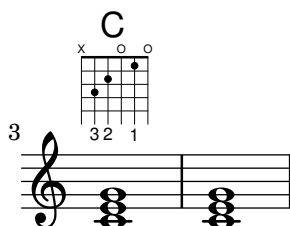
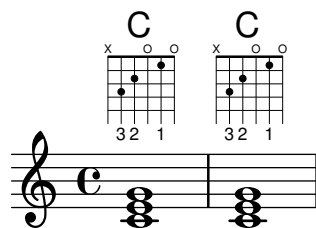
\include "predefined-guitar-fretboards.ly"

myChords = \chordmode {
  c1 c1 \break
  \set chordChanges = ##t
  c1 c1 \break
  c1 c1
}

<<
\new ChordNames { \myChords }
\new FretBoards { \myChords }
\new Staff { \myChords }

```

&gt;&gt;



### *Tablas alternativas de diagramas de posiciones*

Se pueden crear tablas alternativas de diagramas de posiciones. Se utilizarían para tener diagramas alternativos para un acorde dado.

Para usar una tabla alternativa de diagramas de posiciones, se debe crear la tabla primero. Después se añaden los diagramas a la tabla.

La tabla de diagramas de posiciones que se crea puede estar vacía o se puede copiar a partir de una tabla existente.

La tabla a usar en la impresión de los diagramas predefinidos se selecciona por medio de la propiedad `\predefinedDiagramTable`.

```
\include "predefined-guitar-fretboards.ly"

% Make a blank new fretboard table
\define custom-fretboard-table-one
  (make-fretboard-table)

% Make a new fretboard table as a copy of default-fret-table
\define custom-fretboard-table-two
  (make-fretboard-table default-fret-table)

% Add a chord to custom-fretboard-table-one
\storePredefinedDiagram #custom-fretboard-table-one
  \chordmode {c}
  #guitar-tuning
  "3-(;3;5;5;5;3-);"

% Add a chord to custom-fretboard-table-two
```

```

\storePredefinedDiagram #custom-fretboard-table-two
\chordmode {c}
#guitar-tuning
"x;3;5;5;5;o;"

<<
\chords {
  c1 | d1 |
  c1 | d1 |
  c1 | d1 |
}
\new FretBoards {
  \chordmode {
    \set predefinedDiagramTable = #default-fret-table
    c1 | d1 |
    \set predefinedDiagramTable = #custom-fretboard-table-one
    c1 | d1 |
    \set predefinedDiagramTable = #custom-fretboard-table-two
    c1 | d1 |
  }
}
\new Staff {
  \clef "treble_8"
  <<
    \chordmode {
      c1 | d1 |
      c1 | d1 |
      c1 | d1 |
    }
    {
      s1_\markup "Default table" | s1 |
      s1_\markup \column {"New table" "from empty"} | s1 |
      s1_\markup \column {"New table" "from default"} | s1 |
    }
  >>
}
>>

```

Diagram 1: C chord, fingering 32 1

Diagram 2: D chord, fingering 132

Diagram 3: C chord, fingering iii

Diagram 4: D chord, fingering v

Diagram 5: C chord, fingering iii

Diagram 6: D chord, fingering 132

Staff notation: C, D, C, D, C, D

Labels: Default table, New table from empty, New table from default

## Véase también

Referencia de la notación: [Tablaturas personalizadas], página 386, [Diagramas de traste automáticos], página 411, [Panorámica del modo de acordes], página 442, [Diagramas predefinidos de trastes], página 400.

Archivos de inicio: `ly/predefined-guitar-fretboards.ly`,  
`ly/predefined-guitar-ninth-fretboards.ly`,  
`ly/predefined-ukulele-fretboards.ly`,  
`ly/predefined-mandolin-fretboards.ly`.

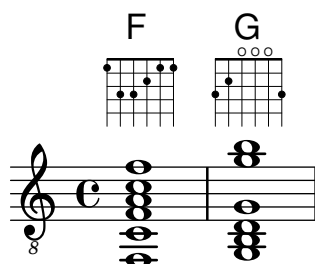
Fragmentos de código: Sección “Fretted strings” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “fret-diagram-interface” en *Referencia de Funcionamiento Interno*.

## Diagramas de traste automáticos

Se pueden crear diagramas de traste automáticamente a partir de las notas introducidas usando el contexto `FretBoards`. Si no está disponible ningún diagrama predefinido para las notas introducidas en la afinación `stringTunings` activa, este contexto calcula las cuerdas y los trastes que se pueden usar para tocar las notas.

```
<<
  \new ChordNames {
    \chordmode {
      f1 g
    }
  }
  \new FretBoards {
    <f, c f a c' f'>1
    <g,\6 b, d g b g'>1
  }
  \new Staff {
    \clef "treble_8"
    <f, c f a c' f'>1
    <g, b, d g b' g'>1
  }
>>
```



Puesto que de forma predeterminada no se carga ningún diagrama predefinido, el comportamiento predeterminado es el cálculo automático de los diagramas de traste. Una vez que los diagramas predeterminados se han cargado, se puede habilitar e inhabilitar el cálculo automático con instrucciones predefinidas:

```
\storePredefinedDiagram #default-fret-table
  <c e g c' e'>
  #guitar-tuning
  "x;3-1-(;5-2;5-3;5-4;3-1-1-);"
```

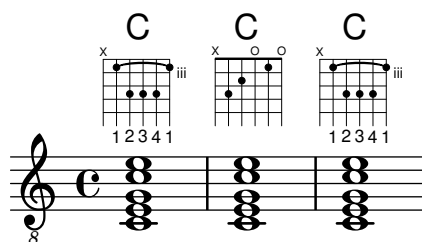
```
<<
  \new ChordNames {
    \chordmode {
      c1 c c
    }
  }
```



```

    }
  }
  \new FretBoards {
    <c e g c' e'>1
    \predefinedFretboardsOff
    <c e g c' e'>1
    \predefinedFretboardsOn
    <c e g c' e'>1
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1
    <c e g c' e'>1
    <c e g c' e'>1
  }
  >>

```



A veces el calculador de trastes es incapaz de encontrar un diagrama aceptable. Esto se puede remediar generalmente mediante la asignación manual de una nota a una cuerda. En muchos casos sólo hay que colocar manualmente una nota sobre una cuerda; el resto de las notas se situará en el lugar adecuado por parte del contexto **FretBoards**.

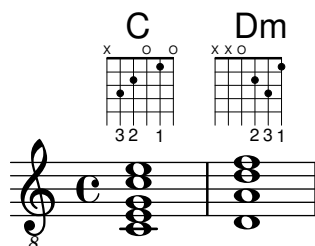
Se pueden añadir digitaciones a los diagramas de traste del contexto **FretBoard**.

```

<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new FretBoards {
    <c-3 e-2 g c'-1 e'>1
    <d a-2 d'-3 f'-1>1
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1
    <d a d' f'>1
  }

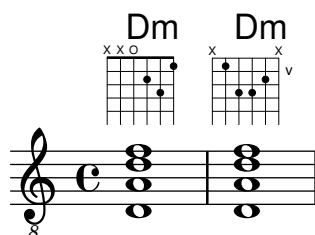
```

&gt;&gt;



El traste mínimo que usar en el cálculo de cuerdas y trastes para el contexto `FretBoard` se puede fijar con la propiedad `minimumFret`.

```
<<
  \new ChordNames {
    \chordmode {
      d1:m d:m
    }
  }
  \new FretBoards {
    <d a d' f'>1
    \set FretBoards.minimumFret = #5
    <d a d' f'>1
  }
  \new Staff {
    \clef "treble_8"
    <d a d' f'>1
    <d a d' f'>1
  }
>>
```



Las cuerdas y los trastes para el contexto `FretBoards` dependen de la propiedad `stringTunings`, que tiene el mismo significado que en el contexto de tablatura `TabStaff`. Consulte [Tablaturas personalizadas], página 386, para ver más información sobre la propiedad `stringTunings`.

El aspecto gráfico de un diagrama de trastes se puede personalizar de acuerdo a las preferencias del usuario a través de las propiedades del interface `fret-diagram-interface`. Los detalles están en Sección “fret-diagram-interface” en *Referencia de Funcionamiento Interno*. Para un diagrama de `FretBoards`, las propiedades del interface pertenecen a `FretBoards.FretBoard`.

## Instrucciones predefinidas

`\predefinedFretboardsOff`, `\predefinedFretboardsOn`.

## Véase también

Referencia de la notación: [Tablaturas personalizadas], página 386.

Fragmentos de código: Sección “Fretted strings” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “fret-diagram-interface” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

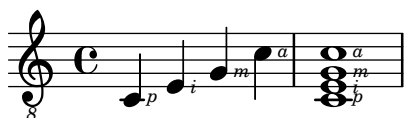
Los cálculos de tablatura automáticos no funcionan bien para instrumentos con afinaciones no monótonas.

## Digitaciones de la mano derecha

Las digitaciones de la mano derecha *p-i-m-a* se deben introducir usando `\rightHandFinger` seguido de un número.

**Nota:** Si el número se introduce en notación de Scheme, recuerde escribir al final un espacio seguido de un ángulo de cierre `>` o similar.

```
\clef "treble_8"
c4\rightHandFinger #1
e\rightHandFinger #2
g\rightHandFinger #3
c'\rightHandFinger #4
<c\rightHandFinger #1 e\rightHandFinger #2
g\rightHandFinger #3 c'\rightHandFinger #4 >1
```



Por brevedad, se puede escribir `\rightHandFinger` como algo más corto, como por ejemplo RH:

```
RH=#rightHandFinger
```

## Fragmentos de código seleccionados

### *Posicionamiento de digitaciones de mano derecha*

Es posible ejercer un mayor control sobre la colocación de las digitaciones de la mano derecha estableciendo el valor de una propiedad específica, como se muestra en el ejemplo siguiente.

```
#(define RH rightHandFinger)

\relative c {
  \clef "treble_8"

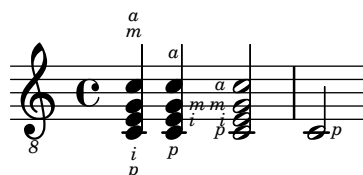
  \set strokeFingerOrientations = #'(up down)
  <c\RH #1 e\RH #2 g\RH #3 c\RH #4 >4

  \set strokeFingerOrientations = #'(up right down)
  <c\RH #1 e\RH #2 g\RH #3 c\RH #4 >4

  \set strokeFingerOrientations = #'(left)
  <c\RH #1 e\RH #2 g\RH #3 c\RH #4 >2

  \set strokeFingerOrientations = #'(right)
```

```
c\RH #1
}
```

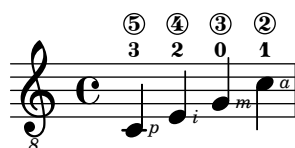


### *Digitaciones - indicación del número de cuerda y digitaciones de mano derecha*

En este ejemplo se combinan las digitaciones de la mano izquierda, indicaciones del número de cuerda y digitaciones de la mano derecha.

```
#(define RH rightHandFinger)
```

```
\relative c {
  \clef "treble_8"
  <c-3\5\RH #1 >4
  <e-2\4\RH #2 >4
  <g-0\3\RH #3 >4
  <c-1\2\RH #4 >4
}
```



### Véase también

Fragmentos de código: Sección “Fretted strings” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “StrokeFinger” en *Referencia de Funcionamiento Interno*.

### 2.4.2 Guitarra

Casi todos los asuntos de notación relacionados con la música para guitarra están suficientemente cubiertos en la sección general sobre ajustes de instrumentos de trastes, pero hay algunos más de los que merece la pena hablar aquí. De forma ocasional, los usuarios quieren crear documentos del tipo de cancioneros que tengan sólo la letra de las canciones con indicaciones de acordes sobre ella. Dado que LilyPond es un tipografiador de música, no se recomienda para documentos que no tienen notación musical. Una alternativa mejor es un procesador de texto, editor de texto o, para usuarios con experiencia, un tipografiador como GuitarTeX.

### Indicar la posición y la cejilla

Este ejemplo muestra cómo incluir indicaciones de posición y cejilla de guitarra:

```
\relative {
  \clef "treble_8"
  b,16 d g b e
  \textSpannerDown
  \override TextSpanner.bound-details.left.text = "XII "
  g16\startTextSpan
  b16 e g e b g\stopTextSpan
```

```
e16 b g d
}
```



Véase también

Referencia de la notación: [Extensiones de texto], página 258.

Fragmentos de código: Sección “Fretted strings” en *Fragmentos de código*, Sección “Expressive marks” en *Fragmentos de código*.

## Indicar armónicos y notas tapadas

Se pueden usar cabezas de nota especiales para indicar notas apagadas o armónicos. Los armónicos se suelen explicar de forma más completa con un marcado de texto.

```
\relative {
  \clef "treble_8"
  \override NoteHead.style = #'harmonic-mixed
  d'8~\markup { \italic \fontsize #-2 "harm. 12" } <g b>4
}
```

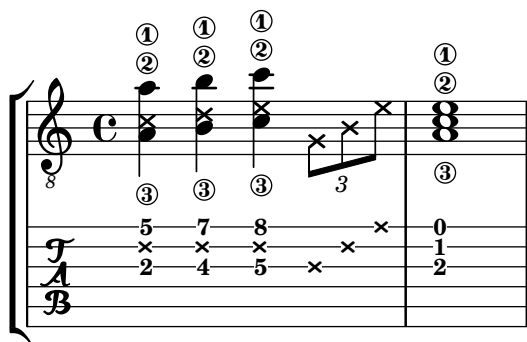


Las notas tapadas (llamadas también *notas muertas*) están contempladas dentro de los pentagramas normales y de tablatura:

```
music = \relative {
  < a\3 \deadNote c\2 a'\1 >4
  < b\3 \deadNote d\2 b'\1 >
  < c\3 \deadNote e\2 c'\1 >
  \deadNotesOn
  \tuplet 3/2 { g8 b e }
  \deadNotesOff
  < a,\3 c\2 e\1 >1
}

\new StaffGroup <<
  \new Staff {
    \clef "treble_8"
    \music
  }
  \new TabStaff {
    \music
  }
}
```

&gt;&gt;



Otra técnica de ejecución (utilizada especialmente en la guitarra eléctrica) se llama *tapado de palma*. La cuerda es parcialmente tapada por la palma de la mano que pulsa (de ahí el nombre). LilyPond contempla la notación de las notas de estilo de tapado de palma mediante el cambio de la forma de la cabeza de nota a un triángulo.

```
\new Voice { % Warning: explicit Voice instantiation is
              % required to have palmMuteOff work properly
              % when palmMuteOn comes at the beginning of
              % the piece.
\relative c, {
  \clef "G_8"
  \palmMuteOn
  e8^\markup { \musicglyph "noteheads.s2do" = palm mute }
  < e b' e > e
  \palmMuteOff
  e e \palmMute e e e |
  e8 \palmMute { e e e } e e e e |
  < \palmMute e b' e >8 \palmMute { e e e } < \palmMute e b' e >2
}
}
```



## Véase también

Fragmentos de código: Sección “Fretted strings” en *Fragmentos de código*.

Referencia de la notación: [Cabezas de nota especiales], página 41, Sección A.9 [Estilos de cabezas de nota], página 734.

## Indicación de acordes de potencia o de quinta vacía

Los acordes de quinta vacía o acordes de potencia (*power chords*) y sus símbolos se pueden tipografiar en el modo de acordes o como construcciones de acorde normales. Como excepción, se especifica la quinta en estos nombres de acorde, mientras que habitualmente se omite en otros acordes (p. ej. en las tríadas mayores o menores).

```
ChordsAndSymbols = {
  \chordmode {
    e,,1:5
```

```

a,, :5.8
\set TabStaff.restrainOpenStrings = ##t
\set minimumFret = #8
c, :5
f, :5.8
}
\set minimumFret = #2
\set restrainOpenStrings = ##f
<a, e> <a cis' e'>
<g d' g'>
}
\score {
  <<
    \new ChordNames {
      \ChordsAndSymbols
    }
    \new Staff {
      \clef "treble_8"
      \ChordsAndSymbols
    }
    \new TabStaff {
      \ChordsAndSymbols
    }
  >>
}

```

## Véase también

Glosario musical: Sección “power chord” en *Glosario Musical*.

Referencia de la notación: [Acordes extendidos y alterados], página 444, [Impresión de los nombres de acorde], página 448.

Fragmentos de código: Sección “Fretted strings” en *Fragmentos de código*.

### 2.4.3 Banjo

#### Tablaturas de banjo

LilyPond contempla el banjo de cinco cuerdas de una manera básica. Cuando haga tablaturas para banjo de cinco cuerdas, utilice la función de formato de tablatura de banjo para obtener los números correctos de los trastes para la quinta cuerda:

```

music = {
  g8 d' g'\5 a b g e d' |
  g4 d''8\5 b' a'\2 g'\5 e'\2 d' |
  g4
}

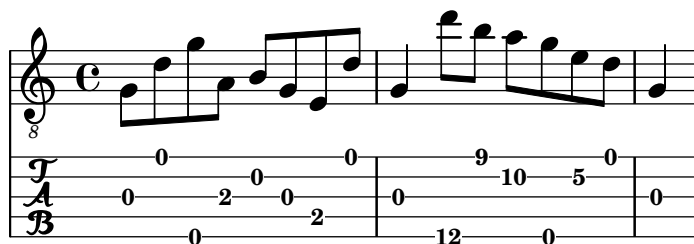
```

```

}

<<
\new Staff \with { \omit StringNumber }
{ \clef "treble_8" \music }
\new TabStaff \with {
  tablatureFormat = #fret-number-tablature-format-banjo
  stringTunings = #banjo-open-g-tuning
}
{ \music }
>>

```



Se encuentran predefinidas varias afinaciones comunes para el banjo de cinco cuerdas: `banjo-c-tuning` (sol-Do-Sol-Si-Re), `banjo-modal-tuning` (sol-Re-Sol-Do-Re), `banjo-open-d-tuning` (Re Mayor al aire, la-Re-Fa $\sharp$ -La-Re) y `banjo-open-dm-tuning` (Re menor al aire, la-Re-Fa-La-Re).

Estas afinaciones se pueden convertir a afinaciones para banjo de cuatro cuerdas utilizando la función `four-string-banjo`:

```
\set TabStaff.stringTunings = #(four-string-banjo banjo-c-tuning)
```

## Véase también

Archivos de inicio: `ly/string-tunings-init.ly`

Fragmentos de código: Sección “Fretted strings” en *Fragmentos de código*.

### 2.4.4 Laúd

#### Tablaturas de laúd

LilyPond contempla las tablaturas para laúd.

Para obtener las cuerdas graves adicionales, utilice `additionalBassStrings`, donde se determinan las notas correspondientes a dichas cuerdas. Se imprimen por debajo de la línea inferior como: a, /a, //a, ///a, 4, 5, etc.

Se debe usar el valor `fret-letter-tablature-format` para la propiedad de contexto `tablatureFormat` y probablemente también ajustar la propiedad `fretLabels` para una mayor personalización.

```

m = { f'4 d' a f d a, g, fis, e, d, c, \bar "|" }

\score {
  <<
    \new Staff { \clef bass \cadenzaOn \m }
    \new TabStaff \m
  >>
  \layout {
    \context {

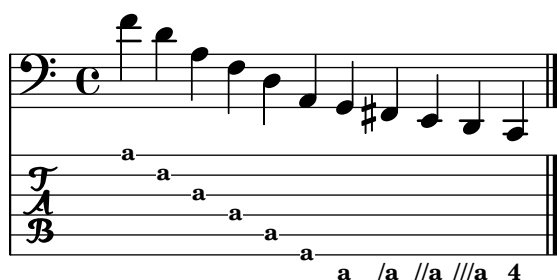
```



```

\Score
  tablatureFormat = #fret-letter-tablature-format
}
\context {
  \TabStaff
  stringTunings = \stringTuning <a, d f a d' f'>
  additionalBassStrings = \stringTuning <c, d, e, fis, g,>
  fretLabels = #'("a" "b" "r" "d" "e" "f" "g" "h" "i" "k")
}
}
}

```



## Advertencias y problemas conocidos

No se contempla el uso de `FretBoards` con `additionalBassStrings`; de utilizarse, puede dar lugar a un resultado poco satisfactorio.

## 2.5 Percusión

### 2.5.1 Notación común para percusión

La notación rítmica se emplea primordialmente para la percusión y la batería, pero también se puede utilizar para mostrar los valores rítmicos una melodía.

### Referencias para percusión

- La notación de algunas percusiones se puede hacer sobre una pauta de ritmo; esto se estudia en [Mostrar los ritmos de la melodía], página 86, y [Crear instancias de pentagramas nuevos], página 201.
- La salida MIDI se trata en una sección aparte; véase Sección 3.5 [Creación de salida MIDI], página 552.

### Véase también

Referencia de la notación: [Mostrar los ritmos de la melodía], página 86, [Crear instancias de pentagramas nuevos], página 201. Sección 3.5 [Creación de salida MIDI], página 552.

Fragmentos de código: Sección “Percussion” en *Fragmentos de código*.

### Notación básica de percusión

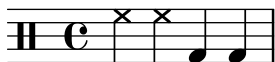
Las notas de percusión se pueden escribir en el modo `\drummode`, que es similar al modo estándar para introducir notas. La manera más fácil de escribir notas de percusión es utilizar la instrucción `\drums`, que crea el contexto y el modo de entrada apropiados para percusión:

```

\drums {
  hihat4 hh bassdrum bd
}

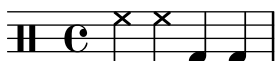
```

}



Esto es una abreviatura de

```
\new DrumStaff \drummode {
  hihat4 hh bassdrum bd
}
```



Cada elemento de un set de percusión lleva un nombre completo y un nombre abreviado, y los dos se pueden usar en la entrada. La lista completa de nombre de instrumentos de percusión se encuentra en Sección A.15 [Notas de percusión], página 802.

Observe que la notación normal de las notas con altura determinada (como `cis4`) en un contexto `DrumStaff` producen un mensaje de error. Las claves de percusión se añaden automáticamente al contexto `DrumStaff` pero también se pueden establecer explícitamente o utilizar otras claves.

```
\drums {
  \clef percussion
  bd4 4 4 4
  \clef treble
  hh4 4 4 4
}
```



Hay ciertos detalles respecto a cómo está contemplado el MIDI para los instrumentos de percusión; para ver los detalles consulte Sección 3.5 [Creación de salida MIDI], página 552.

## Véase también

Referencia de la notación: Sección 3.5 [Creación de salida MIDI], página 552, Sección A.15 [Notas de percusión], página 802.

Archivos de inicio: `ly/drumpitch-init.ly`.

Fragmentos de código: Sección “Percussion” en *Fragmentos de código*.

## Redobles

Los redobles se indican mediante tres barras cruzadas en la plica. Para las negras o notas más largas, las tres barras se muestran explícitamente, las corcheas se presentan con dos barras cruzadas (siendo la barra de corchea la tercera), y los redobles más breves que las corcheas tienen una barra cruzada para complementar las barras normales. Esto se consigue mediante la notación de trémolo, véase [Repeticiones de trémolo], página 173.

```
\drums {
  \time 2/4
  sn16 8 16 8 8:32 ~
  8 8 4:32 ~
  4 8 16 16
  4 r4
```

}



Los golpes de baqueta se pueden indicar mediante la colocación de los elementos de marcado "R" o "L" encima o debajo de las notas, como se estudia en Sección 5.4.2 [Dirección y posición], página 661. La propiedad `staff-padding` se puede sobrescribir para conseguir una línea de base satisfactoria.

```
\drums {
  \repeat unfold 2 {
    sn16^"L" 16^"R" 16^"L" 16^"L" 16^"R" 16^"L" 16^"R" 16^"R"
    \stemUp
    sn16_"L" 16_"R" 16_"L" 16_"L" 16_"R" 16_"L" 16_"R" 16_"R"
  }
}
```



## Véase también

Referencia de la notación: [Repeticiones de trémolo], página 173.

Fragmentos de código: Sección “Percussion” en *Fragmentos de código*.

## Percusión afinada

Ciertos instrumentos de percusión de altura determinada (p.ej.: el xilófono, el vibráfono y los timbales) se escriben usando pentagramas normales. Esto se estudia en otras secciones del manual.

## Véase también

Referencia de la notación: Sección A.6 [Instrumentos MIDI], página 709.

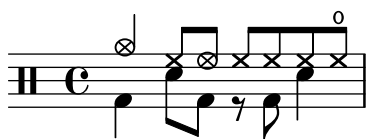
Fragmentos de código: Sección “Percussion” en *Fragmentos de código*.

## Pautas de percusión

Una parte de percusión para más de un instrumento, normalmente utiliza una pauta de varias líneas donde cada posición dentro de la pauta se refiere a un elemento de percusión. Para tipografiar la música, se deben interpretar las notas dentro de los contextos Sección “DrumStaff” en *Referencia de Funcionamiento Interno* y Sección “DrumVoice” en *Referencia de Funcionamiento Interno*.

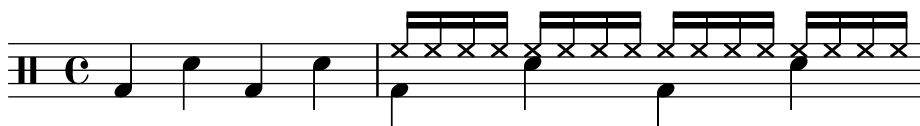
```
up = \drummode {
  crashcymbal4 hihat8 halfopenhihat hh hh hh openhihat
}
down = \drummode {
  bassdrum4 snare8 bd r bd sn4
}
\new DrumStaff <<
  \new DrumVoice { \voiceOne \up }
  \new DrumVoice { \voiceTwo \down }
```

&gt;&gt;



El ejemplo anterior muestra una notación polifónica prolija. La notación polifónica abreviada, descrita en Sección “Oigo voces” en *Manual de Aprendizaje*, también se puede usar. Por ejemplo,

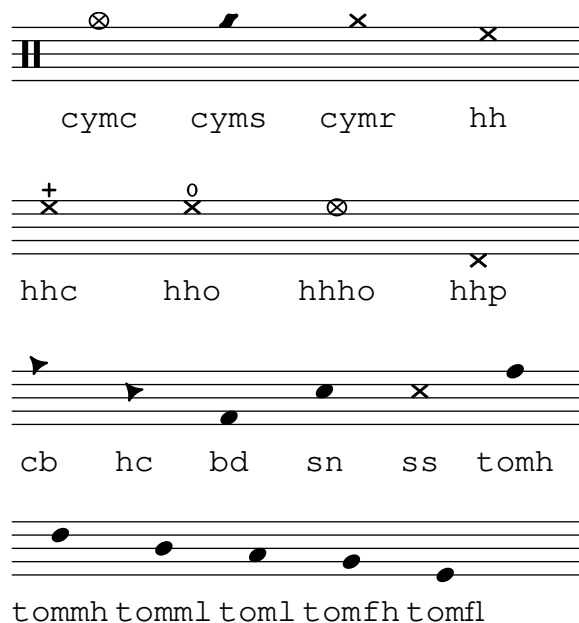
```
\new DrumStaff <<
  \drummode {
    bd4 sn4 bd4 sn4
    << {
      \repeat unfold 16 hh16
    } \ {
      bd4 sn4 bd4 sn4
    } >>
  }
>>
```



También existen otras posibilidades en lo referente a la disposición. Para usarlas, establezca la propiedad `drumStyleTable` en el contexto Sección “DrumVoice” en *Referencia de Funcionamiento Interno*. Se han predefinido las siguientes variables:

`drums-style`

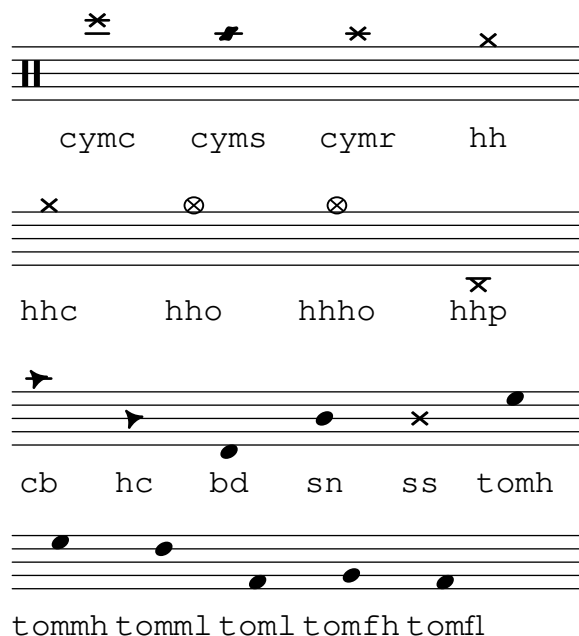
Es la opción por defecto. Se usa para tipografiar una batería típica sobre pentagrama:



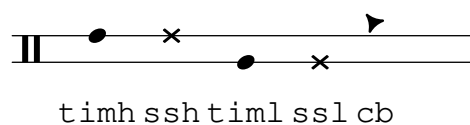
El esquema de percusión contempla seis tambores graves (toms) distintos. Cuando haya menos toms, sencillamente seleccione aquellos que producen el resultado deseado, es decir, para tener toms en las tres líneas centrales utilizará `tommh`, `tomml` y `tomfh`.

**agostini-drums-style**

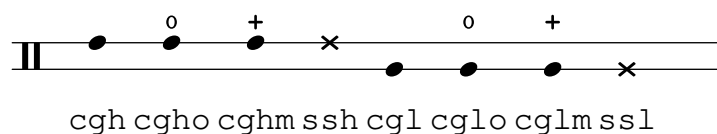
Inventado por el percusionista francés Dante Agostini en 1965, esta notación se emplea normalmente en Francia pero también en muchos otros lugares.

**timbales-style**

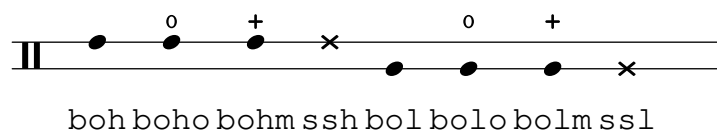
Se usa para tipografiar timbales en una pauta de dos líneas:

**congas-style**

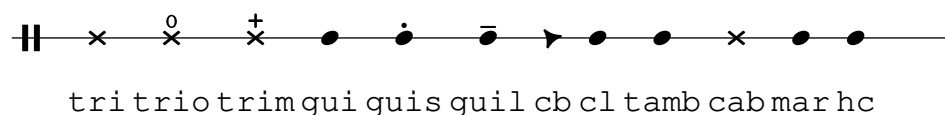
Se usa para tipografiar congas en una pauta de dos líneas:

**bongos-style**

Se usa para bongos sobre una pauta de dos líneas:

**percussion-style**

Para tipografiar toda clase de percusiones simples sobre pautas de una línea:



También se pueden definir estilos de percusión personalizados como se explica en [Pautas de percusión personalizadas], página 425.

## Véase también

Manual de aprendizaje: Sección “Oigo voces” en *Manual de Aprendizaje*.

Referencia de la notación: [Pautas de percusión personalizadas], página 425.

Archivos instalados: `ly/drumpitch-init.ly`.

Fragmentos de código: Sección “Percussion” en *Fragmentos de código*.

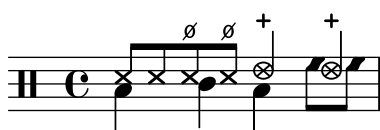
## Pautas de percusión personalizadas

Se pueden definir estilos de percusión personalizados, a los que después puede establecerse la propiedad `drumStyleTable`. Las notaciones existentes se pueden redefinir como una lista de asociación en la que cada entrada debe estar compuesta de cuatro elementos: un nombre, el estilo de cabezas de nota (o `default`), un símbolo de articulación si es necesario (o `#f` en caso contrario), y la posición de la cabeza sobre la pauta. Dicha lista se debe entonces convertir a una tabla de hash de Scheme usando la función `alist->hash-table`.

```
#(define mydrums '(
  (bassdrum      default  #f      -1)
  (snare         default  #f      0)
  (hihat         cross    #f      1)
  (halfopenhihat cross    "halfopen" 1)
  (pedalhihat    xcircle  "stopped" 2)
  (lowtom        diamond  #f      3)))

up = \drummode { hh8 hh hhho hhho hhp4 hhp }
down = \drummode { bd4 sn bd toml8 toml }

\new DrumStaff <<
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \new DrumVoice { \voiceOne \up }
  \new DrumVoice { \voiceTwo \down }
>>
```



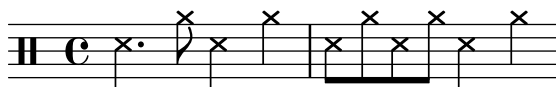
También se pueden añadir nombres nuevos a dichas notaciones personalizadas mediante la variable `drumPitchNames`, que se puede redefinir como una lista de asociación (o aumentada añadiendo al final una lista nueva a su valor existente, como se muestra más abajo), pero también a través de sus entradas individuales. Esto también hace posible definir nombres alternativos o alias.

```
drumPitchNames =
  #(append
    '((leftsnap . sidestick)
      (rightsnap . ridecymbal))
    drumPitchNames)

drumPitchNames.ls = #'sidestick
drumPitchNames.rs = #'ridecymbal

\drums {
  leftsnap4. rightsnap8 leftsnap4 rightsnap
```

```
ls8 rs ls rs ls4 rs
}
```



De manera similar, la propiedad `drumPitchTable` asocia una altura específica (con el significado de un sonido instrumental distinto, tal y como lo proveen las fuentes de sonido MIDI) a cada notación. Dicha propiedad necesita definirse como una tabla de hash, que es de nuevo convertida a partir de una lista de asociación (almacenada de forma predeterminada como la variable `midiDrumPitches`). La redefinición de estas asociaciones se consigue como se ha explicado más arriba, ya sea definiendo una lista de asociación completa o mediante las entradas individuales. El siguiente ejemplo muestra cómo crear un conjunto completo de notaciones con su propia sintaxis de entrada, su propia notación personalizada y la salida MIDI correspondiente.

```
drumPitchNames.dbass      = #'dbass
drumPitchNames.dba        = #'dbass  % 'db is in use already
drumPitchNames.dbassmute  = #'dbassmute
drumPitchNames.dbm        = #'dbassmute
drumPitchNames.do         = #'dopen
drumPitchNames.dopenmute  = #'dopenmute
drumPitchNames.dom        = #'dopenmute
drumPitchNames.dslap      = #'dslap
drumPitchNames.ds         = #'dslap
drumPitchNames.dslapmute  = #'dslapmute
drumPitchNames.dsm        = #'dslapmute
```

```
#(define djembe-style
  '((dbass      default  #f      -2)
    (dbassmute  default  "stopped" -2)
    (dopen      default  #f      0)
    (dopenmute  default  "stopped" 0)
    (dslap      default  #f      2)
    (dslapmute  default  "stopped" 2)))
```

```
midiDrumPitches.dbass      = g
midiDrumPitches.dbassmute  = fis
midiDrumPitches.dopen      = a
midiDrumPitches.dopenmute  = gis
midiDrumPitches.dslap      = b
midiDrumPitches.dslapmute  = ais
```

```
test = \drummode { dba4 do ds dbm dom dsm }
```

```
\score {
  \new DrumStaff \with {
    \override StaffSymbol.line-count = #3
    instrumentName = "Djembe "
    drumStyleTable = #(alist->hash-table djembe-style)
    drumPitchTable = #(alist->hash-table midiDrumPitches)
  } {
    \time 3/4
    \test
```

```

}
\layout {}
\midi {}
}

```



## Véase también

Fragmentos de código: Sección “Percussion” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “DrumStaff” en *Referencia de Funcionamiento Interno*, Sección “DrumVoice” en *Referencia de Funcionamiento Interno*.

## Notas fantasma

Las notas fantasma para la batería e instrumentos de percusión, también conocidas como notas muertas, tapadas, silenciadas o falsas, se pueden crear utilizando la instrucción `\parenthesize`, véase [Paréntesis], página 245.

```

\new DrumStaff <<
  \new DrumVoice = "1" { s1 }
  \new DrumVoice = "2" { s1 }
  \drummode {
    <<
      {
        hh8[ 8] <hh sn> hh16
        \parenthesize sn hh
        \parenthesize sn hh8 <hh sn> hh
      } \\\
      {
        bd4 r4 bd8 8 r8 bd
      }
    >>
  }
>>

```



## Véase también

Referencia de la notación: [Paréntesis], página 245.

Fragmentos de código: Sección “Percussion” en *Fragmentos de código*.



## 2.6 Instrumentos de viento

**Moderato assai**

The image shows a musical score for two flutes. The top staff is for Flauto I, II and the bottom staff is for Flauto III (Gr. Fl.). The time signature is 2/4 and the tempo is Moderato assai. The key signature has two sharps (F# and C#). The score includes various musical notations such as slurs, accents, and dynamic markings (p, mf, sf, mf).

Esta sección incluye elementos de notación musical que afloran al escribir para instrumentos de viento.

### 2.6.1 Notación común para instrumentos de viento

Esta sección trata la notación común para casi todos los instrumentos de viento.

#### Referencias para instrumentos de viento

Muchas cuestiones de la notación para instrumentos de viento tienen que ver con las respiraciones y los golpes de lengua:

- Las respiraciones se pueden especificar mediante silencios o marcas de respiración, véase [Marcas de respiración], página 144.
- La ejecución ligada se indica mediante ligaduras de expresión, véase [Ligaduras de expresión], página 140.
- Los distintos golpes de lengua, desde legato hasta staccato pasando por non legato, se presentan por lo general mediante articulaciones, en ocasiones combinadas con ligaduras de expresión, véase [Articulaciones y ornamentos], página 128, y Sección A.14 [Lista de articulaciones], página 800.
- El frullato se indica generalmente mediante una indicación de trémolo y una marca textual sobre la nota. Véase [Repeticiones de trémolo], página 173.

También existen otros aspectos de la notación musical que son de aplicación a los instrumentos de viento:

- Muchos instrumentos de viento son transpositores, véase [Transposición de los instrumentos], página 28.
- El portamento es característico del trombón, pero otros instrumentos de viento pueden realizar glissandos con llaves o válvulas. Véase [Glissando], página 147.
- Los glissandos de serie armónica, que son posibles en todos los metales pero bastante específicos de las trompas, se escriben en general como notas de adorno, véase [Notas de adorno], página 120.
- Las inflexiones de tono al final de una nota se tratan en [Caídas y elevaciones], página 146.
- Los golpes de llave o de válvula se suelen indicar con el estilo **cross** (aspas) de cabezas de nota, véase [Cabezas de nota especiales], página 41.
- Los instrumentos de viento-madera pueden sobreinflar las notas graves para producir armónicos. Éstos se muestran por medio de la articulación **flageolet**. Véase Sección A.14 [Lista de articulaciones], página 800.
- El uso de sordinas para los metales se suele indicar mediante marcas de texto, pero en los lugares donde se suceden muchos cambios rápidos es mejor utilizar las articulaciones **stopped** (tapado) y **open** (abierto). Véase [Articulaciones y ornamentos], página 128, y Sección A.14 [Lista de articulaciones], página 800.

- Las trompas tapadas se indican por medio de la articulación **stopped**. Véase [Articulaciones y ornamentos], página 128.

## Fragmentos de código seleccionados

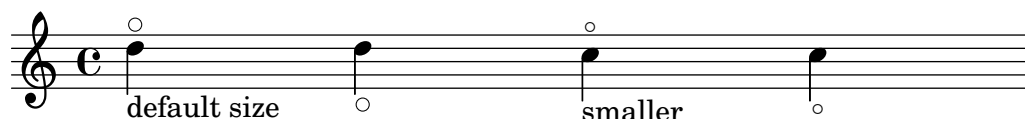
### *Cambiar el tamaño de la marca de \flageolet*

Para hacer más pequeño el círculo de `\flageolet` (armónico) utilice el siguiente truco.

```
smallFlageolet = \tweak font-size -3 \flageolet

\layout { ragged-right = ##f }

\relative c' {
  d4^\flageolet_\markup { default size } d_\flageolet
  c4^\smallFlageolet_\markup { smaller } c_\smallFlageolet
}
```



## Véase también

Referencia de la notación: [Marcas de respiración], página 144, [Ligaduras de expresión], página 140, [Articulaciones y ornamentos], página 128, Sección A.14 [Lista de articulaciones], página 800, [Repeticiones de trémolo], página 173, [Transposición de los instrumentos], página 28, [Glissando], página 147, [Notas de adorno], página 120, [Caídas y elevaciones], página 146, [Cabezas de nota especiales], página 41.

Fragmentos de código: Sección “Winds” en *Fragmentos de código*.

## Digitaciones

Todos los instrumentos de viento aparte del trombón requieren el uso de los distintos dedos para producir cada nota. En los fragmentos de código que aparecen a continuación pueden verse algunos ejemplos de digitaciones.

Se pueden producir diagramas de posiciones para instrumentos de viento madera, como se describe en Sección 2.6.3.1 [Diagramas de posiciones para viento madera], página 433.

## Fragmentos de código seleccionados

### *Símbolos de digitación para instrumentos de viento*

Se pueden conseguir símbolos especiales combinando glifos existentes, lo que es de utilidad para la notación de instrumentos de viento.

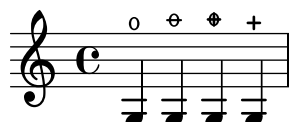
```
centermarkup = {
  \once \override TextScript.self-alignment-X = #CENTER
  \once \override TextScript.X-offset =#(lambda (g)
    (+ (ly:self-alignment-interface::centered-on-x-parent g)
      (ly:self-alignment-interface::x-aligned-on-self g)))
}

\score {
  \relative c'{
    g\open
```

```

\once \override TextScript.staff-padding = #-1.0
\centermarkup
g^\markup {
  \combine
    \musicglyph "scripts.open"
    \musicglyph "scripts.tenuto"
}
\centermarkup
g^\markup {
  \combine
    \musicglyph "scripts.open"
    \musicglyph "scripts.stopped"
}
g\stopped
}
}

```



### *Diagramas de digitación para la flauta dulce*

El ejemplo siguiente muestra cómo se pueden realizar diagramas de digitación para instrumentos de viento.

```

% range chart for paetzold contrabass recorder

centermarkup = {
  \once \override TextScript.self-alignment-X = #CENTER
  \once \override TextScript.X-offset = #(\lambda (g)
    (+ (ly:self-alignment-interface::centered-on-x-parent g)
      (ly:self-alignment-interface::x-aligned-on-self g)))
}

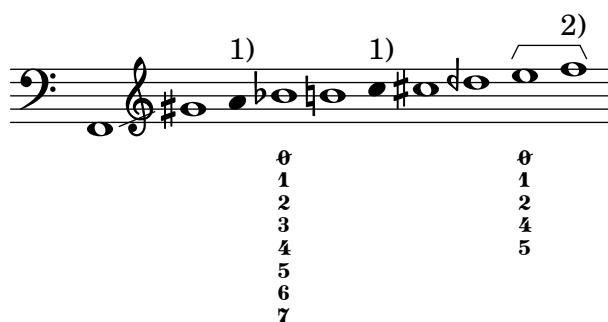
\score {
  \new Staff \with {
    \remove "Time_signature_engraver"
    \omit Stem
    \omit Flag
    \consists "Horizontal_bracket_engraver"
  }
  {
    \clef bass
    \set Score.timing = ##f
    f,1*1/4 \glissando
    \clef violin
    gis'1*1/4
    \stemDown a'4^\markup {1)}
    \centermarkup
    \once \override TextScript.padding = #2
    bes'1*1/4_\markup {\override #'(baseline-skip . 1.7) \column
      { \fontsize #-5 \slashed-digit #0 \finger 1 \finger 2

```

```

        \finger 3 \finger 4 \finger 5 \finger 6 \finger 7} }
b'1*1/4
c''4^\markup {1)}
\centermarkup
\once \override TextScript.padding = #2
cis''1*1/4
deh''1*1/4
\centermarkup
\once \override TextScript.padding = #2
\once \override Staff.HorizontalBracket.direction = #UP
e''1*1/4_\markup {\override #'(baseline-skip . 1.7) \column
  { \fontsize #-5 \slashed-digit #0 \finger 1 \finger 2
    \finger 4 \finger 5} }\startGroup
f''1*1/4^\markup {2)}\stopGroup
}
}

```



## Véase también

Referencia de la notación: Sección 2.6.3.1 [Diagramas de posiciones para viento madera], página 433.

Fragmentos de código: Sección “Winds” en *Fragmentos de código*.

## 2.6.2 Gaita

Esta sección trata de la notación común para gaita.

### Definiciones para la gaita

LilyPond contiene definiciones especiales para la música de gaita escocesa de las tierras altas; para usarlas, escriba

```
\include "bagpipe.ly"
```

al principio del archivo de entrada. De esta forma podrá escribir las notas de adorno especiales que son usuales en la música de gaita, mediante instrucciones breves. Por ejemplo, puede escribir `\taor` en lugar de

```
\grace { \small G32[ d G e] }
```

`bagpipe.ly` también contiene definiciones de alturas para las notas de la gaita en las octavas adecuadas, de forma que no se tenga que preocupar por `\relative` o por `\transpose`.

```
\include "bagpipe.ly"
```

```
{ \grg G4 \grg a \grg b \grg c \grg d \grg e \grg f \grA g A }
```



La música de gaita utiliza siempre la tonalidad de Re mayor (aunque esto no es completamente cierto). Sin embargo, puesto que es la única tonalidad que se puede utilizar, normalmente la armadura no se escribe. Para configurar esto de forma correcta, comience la música siempre con `\hideKeySignature` (ocultar armadura). Si, por algún motivo, quiere mostrar la armadura, puede usar `\showKeySignature` en su lugar.

Alguna música moderna utiliza digitaciones cruzadas sobre el Do y el Fa para bemolizar estas notas. Se puede indicar esto mediante `c-flat` o `f-flat`. De forma parecida, el sol agudo piobaireachd se puede escribir `g-flat` cuando aparece en música ligera.

## Véase también

Fragmentos de código: Sección “Winds” en *Fragmentos de código*.

## Ejemplo de música de gaita

La conocida melodía Amazing Grace tiene este aspecto en notación de gaita.

```
\include "bagpipe.ly"
\layout {
  indent = 0.0\cm
  \context { \Score \remove "Bar_number_engraver" }
}

\header {
  title = "Amazing Grace"
  meter = "Hymn"
  arranger = "Trad. arr."
}

{
  \hideKeySignature
  \time 3/4
  \grg \partial 4 a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 \grg f8 e
  \thrwd d2 \grg b4
  \grG a2 \grg a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 \grg e8. f16
  \dblA A2 \grg A4
  \grg A2 f8. A16
  \grg A2 \hdbl f8[ e32 d16.]
  \grg f2 \grg f8 e
  \thrwd d2 \grg b4
  \grG a2 \grg a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 e4
}
```

```

\thrwd d2.
\slurd d2
\bar "|."
}

```

## Amazing Grace

Hymn

Trad. arr.



### Véase también

Fragmentos de código: Sección “Winds” en *Fragmentos de código*.

### 2.6.3 Instrumentos de viento madera

Esta sección estudia la notación específica para instrumentos de viento madera.

#### 2.6.3.1 Diagramas de posiciones para viento madera

Se pueden utilizar diagramas de posiciones para los instrumentos de viento madera para indicar la digitación que usar al tocar ciertas notas, y están disponibles para los siguientes instrumentos:

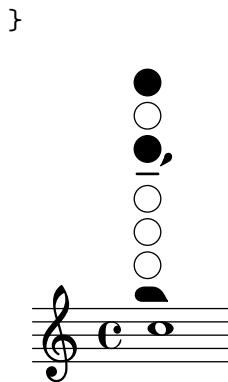
- flautín
- flauta
- oboe
- clarinete
- clarinete bajo
- saxofón
- fagot
- contrafagot

Los diagramas se crean como elementos de marcado:

```

c''1^\markup {
  \woodwind-diagram #'piccolo #'((lh . (gis))
                                (cc . (one three))
                                (rh . (ees)))

```



Las llaves pueden estar abiertas, parcialmente cubiertas, con el anillo pulsado o totalmente apretadas:

```
\textLength0n
c''1^\markup {
  \center-column {
    "one quarter"
    \woodwind-diagram #'flute #'((cc . (one1q))
                                (lh . ()))
                                (rh . ()))
  }
}
```

```
c''1^\markup {
  \center-column {
    "one half"
    \woodwind-diagram #'flute #'((cc . (one1h))
                                (lh . ()))
                                (rh . ()))
  }
}
```

```
c''1^\markup {
  \center-column {
    "three quarter"
    \woodwind-diagram #'flute #'((cc . (one3q))
                                (lh . ()))
                                (rh . ()))
  }
}
```

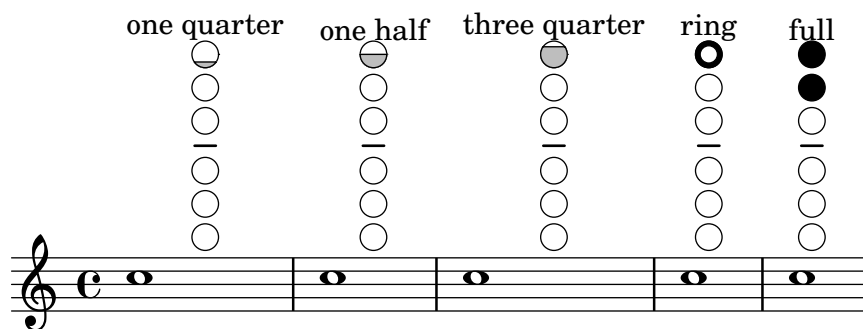
```
c''1^\markup {
  \center-column {
    "ring"
    \woodwind-diagram #'flute #'((cc . (oneR))
                                (lh . ()))
                                (rh . ()))
  }
}
```

```
c''1^\markup {
```

```

\center-column {
  "full"
  \woodwind-diagram #'flute #'((cc . (oneF two))
                        (lh . ()))
                        (rh . ()))
}

```

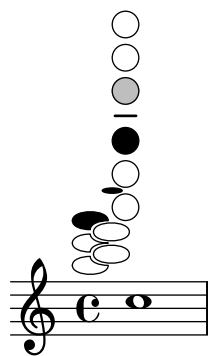


Los trinos se indican mediante llaves sombreadas:

```

c''1^\markup {
  \woodwind-diagram #'bass-clarinet
                    #'((cc . (threeT four))
                      (lh . ()))
                      (rh . (b fis)))
}

```



Es posible imprimir una amplia variedad de trinos:

```

\textLengthOn
c''1^\markup {
  \center-column {
    "one quarter to ring"
    \woodwind-diagram #'flute #'((cc . (one1qTR))
                                  (lh . ()))
                                  (rh . ()))
  }
}

```

```

c''1^\markup {
  \center-column {
    "ring to shut"
    \woodwind-diagram #'flute #'((cc . (oneTR))

```



```

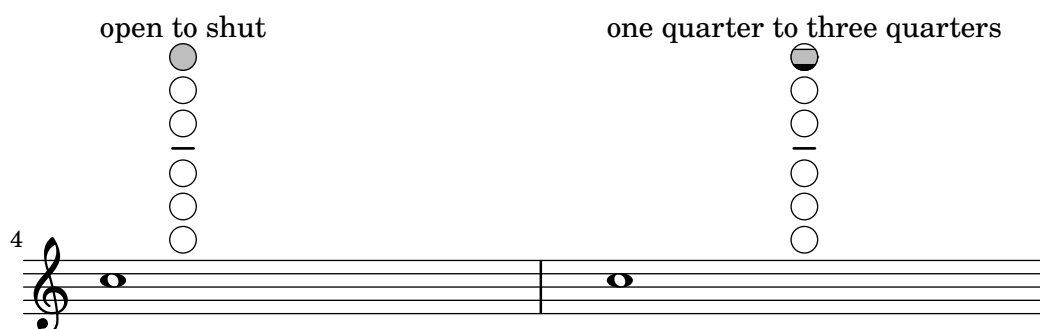
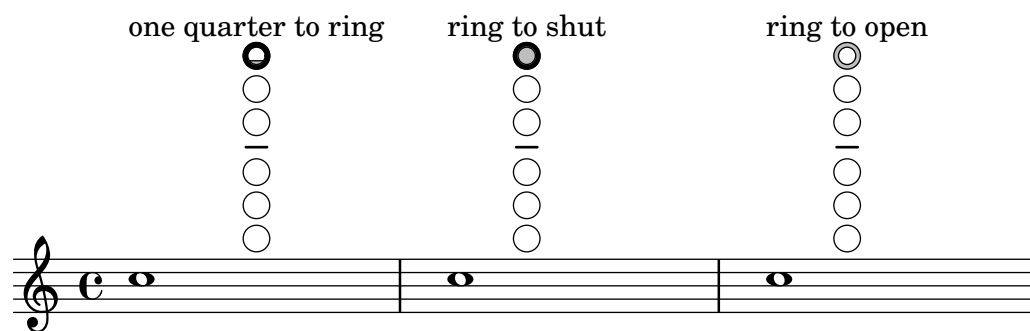
                                (lh . ())
                                (rh . ()))
    }
}

c''1^\markup {
  \center-column {
    "ring to open"
    \woodwind-diagram #'flute #'((cc . (oneRT))
                                (lh . ())
                                (rh . ()))
  }
}

c''1^\markup {
  \center-column {
    "open to shut"
    \woodwind-diagram #'flute #'((cc . (oneT))
                                (lh . ())
                                (rh . ()))
  }
}

c''1^\markup {
  \center-column {
    "one quarter to three quarters"
    \woodwind-diagram #'flute #'((cc . (one1qT3q))
                                (lh . ())
                                (rh . ()))
  }
}

```



La lista de todas las tonalidades y ajustes posibles para un instrumento dado se puede imprimir en la consola usando `$(print-keys-verbose 'flute)` o en el archivo de registro usando `$(print-keys-verbose 'flute (current-error-port))`, aunque no se pueden mostrar en la salida de música impresa.

Es posible crear diagramas nuevos, aunque requiere conocimientos de Scheme y quizá no esté al alcance de todos los usuarios. Los patrones de los diagramas están en `scm/define-woodwind-diagrams.scm` y en `scm/display-woodwind-diagrams.scm`.

## Instrucciones predefinidas

### Fragmentos de código seleccionados

#### *Listado de los diagramas para viento madera*

El fragmento de música que aparece a continuación presenta todos los diagramas de viento madera que se encuentran definidos en LilyPond por el momento.

```
\layout {
  indent = 0
}

\relative c' {
  \textLengthOn
  c1~
  \markup {
    \center-column {
      'tin-whistle
      " "
      \woodwind-diagram
      #'tin-whistle
      #'()
    }
  }
}

c1~
\markup {
  \center-column {
    'piccolo
    " "
    \woodwind-diagram
    #'piccolo
    #'()
  }
}

c1~
\markup {
  \center-column {
    'flute
    " "
    \woodwind-diagram
    #'flute
    #'()
  }
}
```

```

    }
  }
  c1~\markup {
    \center-column {
      'oboe
      " "

      \woodwind-diagram
      #'oboe
      #'()
    }
  }
}

```

```

  c1~\markup {
    \center-column {
      'clarinet
      " "

      \woodwind-diagram
      #'clarinet
      #'()
    }
  }
}

```

```

  c1~\markup {
    \center-column {
      'bass-clarinet
      " "

      \woodwind-diagram
      #'bass-clarinet
      #'()
    }
  }
}

```

```

  c1~\markup {
    \center-column {
      'saxophone
      " "

      \woodwind-diagram
      #'saxophone
      #'()
    }
  }
}

```

```

  c1~\markup {
    \center-column {
      'bassoon
      " "

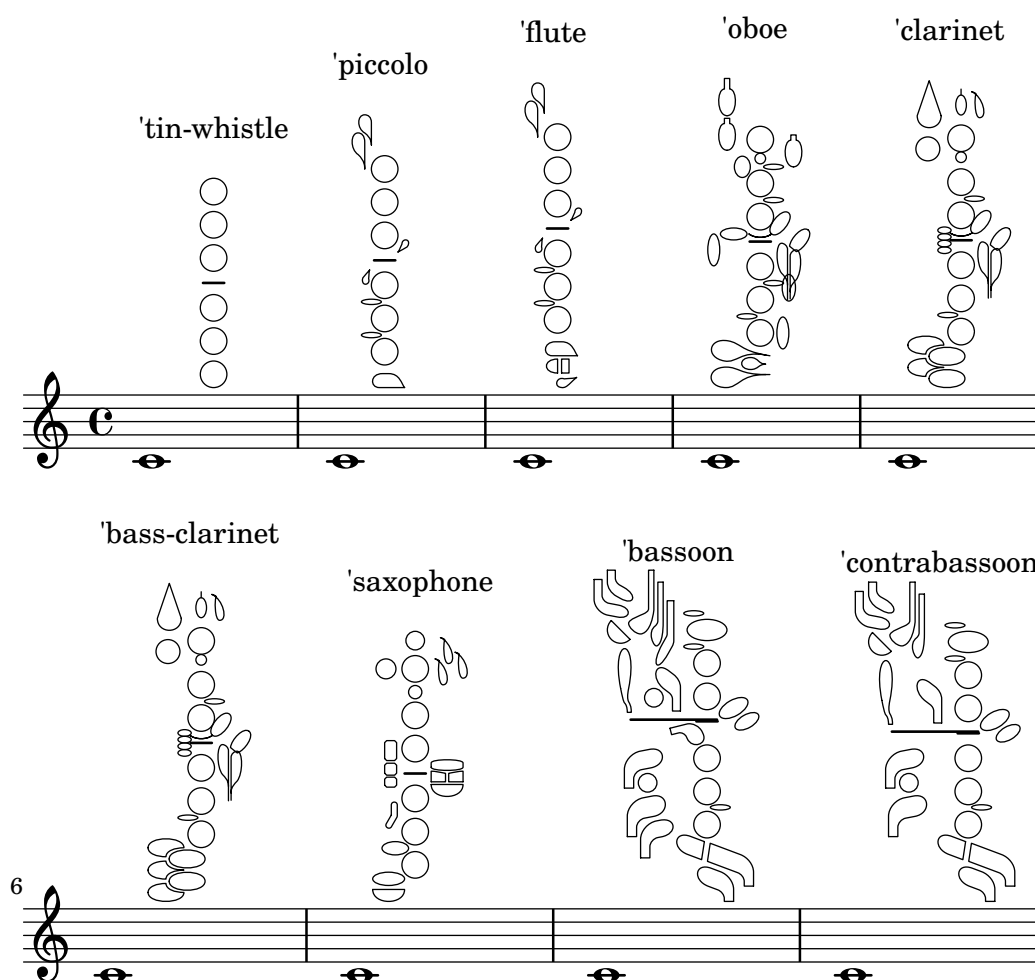
      \woodwind-diagram
      #'bassoon
      #'()
    }
  }
}

```

```

c1^\markup {
  \center-column {
    'contrabassoon
    " "
    \woodwind-diagram
    #'contrabassoon
    #'()
  }
}

```



### *Diagramas para viento madera gráficos y textuales*

En muchos casos, las llaves que no están en la columna central se pueden presentar por el nombre de la llave así como de forma gráfica.

```

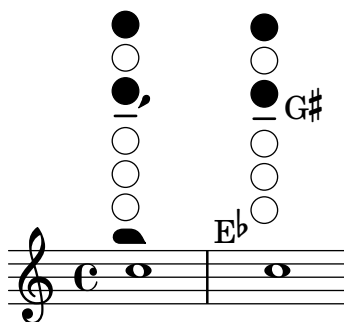
\relative c'' {
  \textLength0n
  c1^\markup
  \woodwind-diagram
  #'piccolo
  #'((cc . (one three))
    (lh . (gis))
    (rh . (ees)))

```

```

c^\markup
  \override #'(graphical . #f) {
    \woodwind-diagram
      #'piccolo
      #'((cc . (one three))
        (lh . (gis))
        (rh . (ees)))
  }
}

```



### *Modificar el tamaño de los diagramas de viento madera*

Se puede cambiar el tamaño y grosor de las líneas de los diagramas de posiciones para instrumentos de viento madera.

```

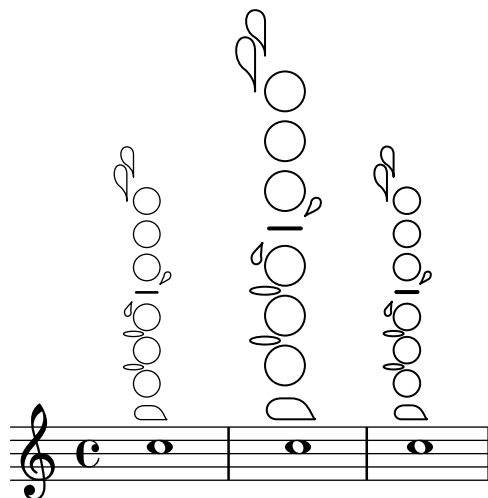
\relative c'' {
  \textLength0n
  c1^\markup
    \woodwind-diagram
      #'piccolo
      #'()

  c^\markup
    \override #'(size . 1.5) {
      \woodwind-diagram
        #'piccolo
        #'()
    }

  c^\markup
    \override #'(thickness . 0.15) {
      \woodwind-diagram
        #'piccolo
        #'()
    }
}

```

}



### Listas de llaves para los diagramas de viento madera

El fragmento de código que aparece a continuación produce una lista de todas las llaves y disposiciones de llave posibles para los diagramas de posiciones de instrumentos de viento madera, tal y como están definidos en `scm/define-woodwind-diagrams.scm`. La lista se muestra en el archivo de registro, pero no en la música. Si desea una salida por la consola, omita el `(current-error-port)` de las instrucciones.

```
#(print-keys-verbose 'piccolo (current-error-port))
#(print-keys-verbose 'flute (current-error-port))
#(print-keys-verbose 'flute-b-extension (current-error-port))
#(print-keys-verbose 'tin-whistle (current-error-port))
#(print-keys-verbose 'oboe (current-error-port))
#(print-keys-verbose 'clarinet (current-error-port))
#(print-keys-verbose 'bass-clarinet (current-error-port))
#(print-keys-verbose 'low-bass-clarinet (current-error-port))
#(print-keys-verbose 'saxophone (current-error-port))
#(print-keys-verbose 'soprano-saxophone (current-error-port))
#(print-keys-verbose 'alto-saxophone (current-error-port))
#(print-keys-verbose 'tenor-saxophone (current-error-port))
#(print-keys-verbose 'baritone-saxophone (current-error-port))
#(print-keys-verbose 'bassoon (current-error-port))
#(print-keys-verbose 'contrabassoon (current-error-port))
```

```
\score {c'1}
```



### Véase también

Archivos de inicio: `scm/define-woodwind-diagrams.scm`,  
`scm/display-woodwind-diagrams.scm`.

Fragmentos de código: Sección “Winds” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “TextScript” en *Referencia de Funcionamiento Interno*, Sección “instrument-specific-markup-interface” en *Referencia de Funcionamiento Interno*.

## 2.7 Notación de acordes

1. Fair is the sun - shine, Fair - er the moon - light  
 2. Fair are the mead - ows, Fair - er the wood - land,

And all the stars in heav'n a - bove;  
 Robed in the flow - ers of bloom - ing spring;

Los acordes se pueden escribir en el modo de acordes, que reconoce ciertas convenciones europeas tradicionales de nomenclatura de acordes. También se pueden imprimir los nombres de los acordes. Además se puede imprimir notación de bajo cifrado.

### 2.7.1 Modo de acordes

Para introducir acordes se utiliza el modo de acordes. Se usa un indicador de la estructura del acorde en lugar de la altura de sus notas.

#### Panorámica del modo de acordes

Los acordes se pueden escribir como música simultánea, como se explica en [Notas en acorde], página 176.

También se pueden escribir los acordes en el “modo de acordes”, que es un modo de entrada que se centra en las estructuras de los acordes dentro de la música europea tradicional (donde se conoce como «cifrado americano») más que en las notas concretas. Esto es muy práctico para los que están familiarizados con la utilización de nombres para describir los acordes. Hay más información sobre los distintos modos de entrada en Sección 5.4.1 [Modos de entrada], página 660.

```
\chordmode { c1 g a g c }
```

Los acordes escritos con el modo de acordes son elementos musicales, y se pueden transportar igual que los acordes escritos mediante construcciones de música simultánea. `\chordmode` es absoluto, pues `\relative` no tiene ningún efecto sobre los bloques `chordmode`. Sin embargo, dentro de `\chordmode` las alturas absolutas son una octava más agudas que en el modo de notas.

Se pueden mezclar los modos de acorde y de nota en la música secuencial:

```
\relative {
  <c' e g>2 <g b d>
  \chordmode { c2 f }
  <c e g>2 <g' b d>
  \chordmode { f2 g }
}
```



## Véase también

Glosario musical: Sección “chord” en *Glosario Musical*.

Referencia de la notación: [Notas en acorde], página 176, Sección 5.4.1 [Modos de entrada], página 660.

Fragmentos de código: Sección “Chords” en *Fragmentos de código*.

## Advertencias y problemas conocidos

No se pueden usar las abreviaturas predefinidas de articulaciones y adornos sobre notas dentro del modo de acordes; véase [Articulaciones y ornamentos], página 128.

## Acordes más usuales

Las tríadas mayores se introducen escribiendo la fundamental y una duración opcional:

```
\chordmode { c2 f4 g }
```



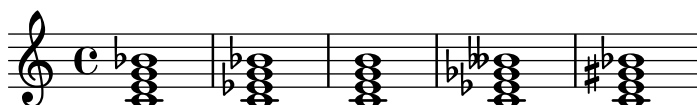
Las tríadas menores, aumentadas y disminuidas se escriben poniendo : y una cadena modificadora de variante después de la duración:

```
\chordmode { c2:m f4:aug g:dim }
```



Se pueden crear acordes de séptima:

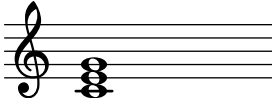


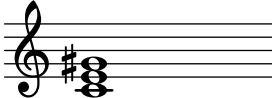
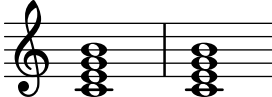
```
\chordmode { c1:7 c:m7 c:maj7 c:dim7 c:aug7 }
```



La tabla que aparece más abajo muestra el efecto de los modificadores de variante sobre los acordes de tríada y de séptima. La séptima añadida de forma predeterminada a los acordes es menor, lo que hace del acorde de séptima dominante el acorde básico de séptima. Todas las alteraciones se consideran en relación con la séptima dominante. Hay una tabla más completa



donde aparecen los usos de los modificadores en Sección A.2 [Modificadores de acorde más usuales], página 696.

Modificador	Acción	Ejemplo
Ninguno	Acción predeterminada; produce una tríada mayor.	
m, m7	Acorde menor. Este modificador baja la tercera.	
dim, dim7	Acorde disminuido. Este modificador baja la tercera, la quinta y, si existe, la séptima.	
aug	Acorde aumentado. Este modificador eleva la quinta.	
maj, maj7	Acorde de séptima mayor. Este modificador añade una séptima elevada. El 7 que sigue a maj es opcional. NO utilice este modificador para crear una tríada mayor.	

## Véase también

Referencia de la notación: Sección A.2 [Modificadores de acorde más usuales], página 696, [Acordes extendidos y alterados], página 444.

Fragmentos de código: Sección “Chords” en *Fragmentos de código*.

## Advertencias y problemas conocidos

Sólo se puede usar un modificador por cada acorde, normalmente sobre la nota más aguda del mismo. Los acordes con más de un modificador se analizan sin producir errores ni advertencias, pero el resultado es impredecible. Los acordes que no se pueden conseguir con un solo modificador se deben alterar en sus notas individuales como se describe en [Acordes extendidos y alterados], página 444.

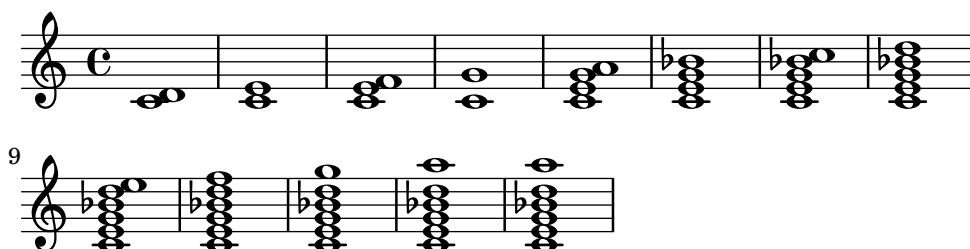
## Acordes extendidos y alterados

Se pueden crear estructuras de acorde de complejidad arbitraria dentro del modo de acordes. Se puede usar la cadena modificadora para extender un acorde, añadir o quitar notas, elevar o bajar notas del acorde y añadir un bajo distinto o crear una inversión.

El primer número que sigue al : se considera que es el ámbito del acorde. El acorde se construye secuencialmente añadiendo terceras a la fundamental hasta que se alcanza el número

especificado. Observe que la séptima añadida como parte de un acorde extendido es la séptima menor, no mayor. Si el ámbito no es una tercera (p.ej. 6), se añaden terceras hasta la tercera más alta inferior al ámbito, y después se añade la nota del ámbito. El mayor valor posible para el ámbito es 13. Cualquier valor mayor se interpreta como 13.

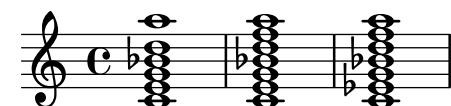
```
\chordmode {
  c1:2 c:3 c:4 c:5
  c1:6 c:7 c:8 c:9
  c1:10 c:11 c:12 c:13
  c1:14
}
```



Como excepción especial, c:5 produce un ‘power chord’, que tiene solamente la fundamental y la quinta.

Puesto que una onцена sin alteración no suena bien cuando se combina con una trecena sin alteración, se elimina la onцена de los acordes de :13 (a no ser que se añada explícitamente).

```
\chordmode {
  c1:13 c:13.11 c:m13
}
```



Se pueden añadir notas individuales a un acorde. Las adiciones siguen el ámbito y van prefijadas por un punto (.). La séptima normal que se añade a un acorde es la séptima menor, no mayor.

```
\chordmode {
  c1:3.5.6 c:3.7.8 c:3.6.13
}
```



Las notas añadidas pueden ser tan agudas como se desee.

```
\chordmode {
  c4:3.5.15 c:3.5.20 c:3.5.25 c:3.5.30
}
```



Las notas añadidas a los acordes se pueden alterar mediante la adición de los sufijos `-` o `+` al número. Para alterar una nota que se ha incluido automáticamente como parte de la estructura básica del acorde, añádala como una nota alterada.

```
\chordmode {
  c1:7+ c:5+.3- c:3-.5-.7-
}
```



Una nota que se quiere suprimir de un acorde se indica dentro de la cadena modificadora prefijándola por un acento circunflejo `^`. Sólo se permite una supresión con `^` dentro de una cadena modificadora.

```
\chordmode {
  c1^3 c:7^5 c:9^3 c:9^3.5 c:13.11^3.7
}
```



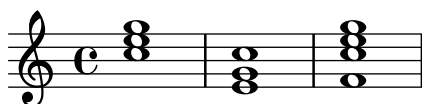
Se puede añadir el modificador `sus` a la cadena modificadora para crear acordes suspendidos. Esto elimina la tercera del acorde. Escriba bien 2 o bien 4 para añadir la segunda o la cuarta al acorde, respectivamente. Cuando `sus` va seguido de una segunda o una cuarta, es equivalente a `^3`, y en caso contrario a `sus4`, o sea, 5.4.

```
\chordmode {
  c1:sus c:sus2 c:sus4 c:5.4
}
```



Se pueden especificar notas añadidas en el bajo (colocar en el bajo una nota distinta de la fundamental del acorde) mediante la adición de `/nota` al acorde.

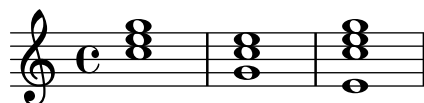
```
\chordmode {
  c'1 c'/e c'/f
}
```



Si la nota añadida ya forma parte del acorde, esto se puede aprovechar para imprimir inversiones de acorde, en cuyo caso la nota no se añade realmente sino que se desplaza a la base del acorde. Sin embargo, se puede tratar como una nota añadida (imprimiéndose así dos veces), mediante la sintaxis `/+nota`.

```
\chordmode {
  c'1 c'/g c'/+e
}
```

}



Las inversiones y realizaciones de acorde automáticas se explican en [Inversiones de acorde y realizaciones especiales], página 447.

Los modificadores de acorde que se pueden utilizar para producir una amplia variedad de acordes estándar se muestran en Sección A.2 [Modificadores de acorde más usuales], página 696.

### Véase también

Referencia de la notación: [Inversiones de acorde y realizaciones especiales], página 447, Sección A.2 [Modificadores de acorde más usuales], página 696.

Fragmentos de código: Sección “Chords” en *Fragmentos de código*.

### Advertencias y problemas conocidos

Cada nota sólo puede estar presente en el acorde una sola vez. Lo que sigue produce simplemente el acorde aumentado, porque 5+ se interpreta en último lugar.

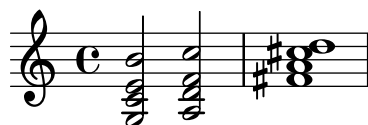
```
\chordmode { c1:3.5.5-.5+ }
```



### Inversiones de acorde y realizaciones especiales

Además de los modificadores de acorde y las notas añadidas en el bajo, se pueden usar varias funciones para imprimir automáticamente acordes en una inversión o realización especial (por ejemplo la realización conocida como ‘drop 2’ que se usa habitualmente en el jazz).

```
\chordmode {
  \dropNote 2 {
    c2:maj7 d:m7
  }
  \invertChords 1 d1:maj7
}
```



A diferencia de las notas añadidas en el bajo que se ven en [Acordes extendidos y alterados], página 444, esto afecta solamente a la manera en que los acordes se imprimen en un pentagrama, y no a los nombres de acorde escritos con letras. Es más, estas funciones se pueden usar no solo en el modo de acordes sino también con la construcción de acorde habitual <...> que se explica en [Notas en acorde], página 176.

### Véase también

Referencia de la notación: [Acordes extendidos y alterados], página 444, [Notas en acorde], página 176.

Fragmentos de código: Sección “Chords” en *Fragmentos de código*.

### 2.7.2 Imprimir los acordes

Los acordes se pueden imprimir por su nombre, además de la impresión estándar como notas sobre un pentagrama.

#### Impresión de los nombres de acorde

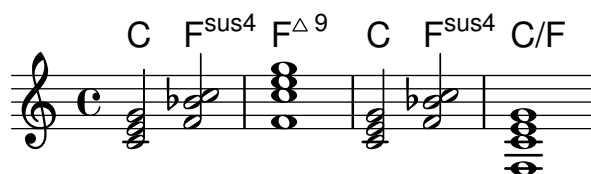
Los nombres de acorde se imprimen dentro del contexto **ChordNames**:

```
\new ChordNames {
  \chordmode {
    c2 f4. g8
  }
}
```

C F G

Los acordes se pueden escribir como notas simultáneas o a través del uso del modo de acordes. El nombre de acorde que se imprime es el mismo independientemente del modo de entrada, a no ser que existan inversiones o bajos añadidos:

```
chordmusic = \relative {
  <c' e g>2 <f bes c>
  <f c' e g>1
  \chordmode {
    c2 f:sus4 c1:/f
  }
}
<<
  \new ChordNames {
    \chordmusic
  }
  {
    \chordmusic
  }
>>
```



Los silencios pasados a un contexto **ChordNames** hacen que se imprima el elemento de marcado **noChordSymbol** (símbolo de N.C., Sin Acorde).

```
<<
  \new ChordNames \chordmode {
    c1
    r1
    g1
    c1
  }
  \chordmode {
    c1
    r1
    g1
  }
```



`\chords { ... }` es una forma abreviada de escribir `\new ChordNames \chordmode { ... }`.

```

\chords {
  c2 f4.:m g8:maj7
}

```

C Fm G<sup>Δ</sup>

```

\new ChordNames {
  \chordmode {
    c2 f4.:m g8:maj7
  }
}

```

C Fm G<sup>Δ</sup>

## Fragmentos de código seleccionados

### *Imprimir los acordes cuando se produce un cambio*

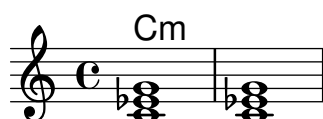
De forma predeterminada se imprimen todos los acordes que aparecen en el código de entrada; se puede modificar este comportamiento de tal manera que los acordes se impriman exclusivamente al comienzo de las líneas y cuando cambia el acorde.

```

harmonies = \chordmode {
  c1:m c:m \break c:m c:m d
}

<<
\new ChordNames {
  \set chordChanges = ##t
  \harmonies
}
\new Staff {
  \relative c' { \harmonies }
}
>>

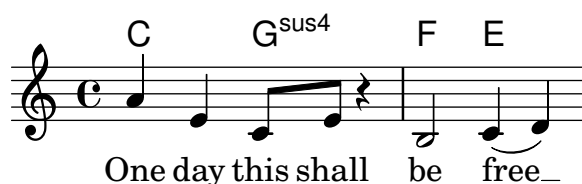
```



### ***Hoja guía de acordes o «lead sheet» sencilla***

Al juntar nombres de acorde en cifrado americano, melodía y letra, obtenemos una hoja guía de acordes o «lead sheet»:

```
<<
\chords { c2 g:sus4 f e }
\new Staff \relative c'' {
  a4 e c8 e r4
  b2 c4( d)
}
\addlyrics { One day this shall be free __ }
>>
```



### **Véase también**

Glosario musical: Sección “chord” en *Glosario Musical*.

Referencia de la notación: [Escribir música en paralelo], página 197.

Fragmentos de código: Sección “Chords” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “ChordNames” en *Referencia de Funcionamiento Interno*, Sección “ChordName” en *Referencia de Funcionamiento Interno*, Sección “Chord\_name\_engraver” en *Referencia de Funcionamiento Interno*, Sección “Volta\_engraver” en *Referencia de Funcionamiento Interno*, Sección “Bar\_engraver” en *Referencia de Funcionamiento Interno*.

### **Advertencias y problemas conocidos**

Los acordes que contienen inversiones o bajos alterados no reciben un nombre adecuado si se escriben usando música simultánea.

### **Nombres de acorde personalizados**


No hay un sistema único para nombrar acordes. Las distintas tradiciones musicales usan distintos nombres para el mismo conjunto de acordes. También hay distintos símbolos impresos para un nombre de acorde dado. Los nombres y símbolos que se imprimen para los nombres de acorde dados se pueden personalizar.

El esquema de nomenclatura de acordes predeterminado es un sistema para música de Jazz, propuesto por Klaus Ignatzek (véase Sección “Lista de referencias bibliográficas” en *Ensayo*. Se pueden implementar otros sistemas de nomenclatura de acordes mediante funciones de Scheme, como se muestra en el fragmento de código “Chord names alternative” dentro de la sección Sección “Chords” en *Fragmentos de código*.)). Hay una lista de notaciones comunes para acordes de jazz en la tabla de la sección Sección A.1 [Cuadro de nombres de acordes], página 695.

Se puede trucar fácilmente el sistema de nomenclatura de acordes predeterminado. Para empezar, ciertas órdenes predefinidas permiten la utilización de distintos idiomas para la

nota fundamental. Entre ellos están `\germanChords`, `\semiGermanChords`, `\italianChords` y `\frenchChords`:

default	E/D	Cm	B/B	B <sup>#</sup> /B <sup>#</sup>	B <sup>b</sup> /B <sup>b</sup>
german	E/d	Cm	H/h	H <sup>#</sup> /his	B/b
semi-german	E/d	Cm	H/h	H <sup>#</sup> /his	B <sup>b</sup> /b
italian	Mi/Re	Do m	Si/Si	Si <sup>#</sup> /Si <sup>#</sup>	Si <sup>b</sup> /Si <sup>b</sup>
french	Mi/Ré	Do m	Si/Si	Si <sup>#</sup> /Si <sup>#</sup>	Si <sup>b</sup> /Si <sup>b</sup>



Algunos cancioneros alemanes indican los acordes menores como letras minúsculas, sin el sufijo *m*. Esto se puede conseguir estableciendo la propiedad `chordNameLowercaseMinor`:

```
\chords {
  \set chordNameLowercaseMinor = ##t
  c2 d:m e:m f
}
```

C d e F

También se puede afinar la presentación de los nombres de acorde a través de las siguientes propiedades.

#### `chordRootNamer`

El nombre de acorde se imprime normalmente como una letra para la fundamental con una alteración opcional. La transformación de la nota en la letra se realiza por parte de esta función. Los nombres de nota especiales (por ejemplo, la ‘H’ alemana para un acorde de Si) se pueden producir almacenando una función nueva en esta propiedad.

#### `majorSevenSymbol`

Esta propiedad contiene el objeto de marcado que se usa para hacer el seguimiento de la salida de `chordRootNamer` para identificar un acorde de séptima mayor. Las opciones predefinidas son `whiteTriangleMarkup` (triángulo blanco) y `blackTriangleMarkup` (triángulo negro).

#### `additionalPitchPrefix`

Cuando el acorde contiene notas añadidas, se pueden prefijar opcionalmente por un texto. Por omisión no hay ningún prefijo, para evitar la sobrecarga visual, pero para pequeñas cantidades de notas añadidas puede ser visualmente efectivo.

```
\new ChordNames {
  <c e g d'> % add9
  \set additionalPitchPrefix = "add"
  <c e g d'> % add9
}
```

C<sup>9</sup> C<sup>add9</sup>

#### `chordNoteNamer`

Cuando el nombre del acorde contiene notas adicionales aparte de la fundamental (p.ej., un bajo añadido), se utiliza esta función para imprimir la nota adicional. De



forma predeterminada se imprime la nota usando `chordRootNamer`. La propiedad `chordNoteNamer` se puede establecer a una función especializada para cambiar este comportamiento. Por ejemplo, el bajo se puede imprimir en minúscula.

#### `chordNameSeparator`

Las diferentes partes del nombre de un acorde se separan normalmente mediante un pequeño espacio en blanco. Mediante el ajuste de `chordNameSeparator`, podemos usar cualquier marcado que deseemos como separador. Esto no afecta al separador entre un acorde y la nota del bajo: para personalizar eso, utilice `slashChordSeparator`.

```
\chords {
  c4:7.9- c:7.9-/g
  \set chordNameSeparator = \markup { "/" }
  \break
  c4:7.9- c:7.9-/g
}
```

$C^7 \flat^9 C^7 \flat^9 / G$

$C^{7/\flat^9} C^{7/\flat^9} / G$

#### `slashChordSeparator`

Los acordes se pueden tocar sobre una nota del bajo distinta a la fundamental. Se llaman "inversiones" o "acordes de barra inclinada" porque la manera predeterminada de efectuar su notación es con una barra inclinada entre el acorde y la nota del bajo. Por tanto, el valor de `slashChordSeparator` es, por omisión, una barra inclinada, pero podemos cambiarlo a cualquier elemento de marcado que queramos.

```
\chords {
  c4:7.9- c:7.9-/g
  \set slashChordSeparator = \markup { " over " }
  \break
  c4:7.9- c:7.9-/g
}
```

$C^7 \flat^9 C^7 \flat^9 / G$

$C^7 \flat^9 C^7 \flat^9 \text{ over } G$

#### `chordNameExceptions`

Esta propiedad es una lista de parejas. El primer elemento de cada pareja es un conjunto de notas utilizadas para identificar los elementos presentes en el acorde. El segundo elemento es un elemento de marcado que sigue a la salida de `chordRootNamer` para crear el nombre del acorde.

#### `minorChordModifier`

La notación de los acordes menores se suele hacer por medio del sufijo 'm' a la derecha de la letra que indica la fundamental del acorde. Sin embargo, otras convenciones dan preferencia a sufijos diferentes, como un signo menos.

```
\chords {
  c4:min f:min7
  \set minorChordModifier = \markup { "-" }
  \break
}
```

```
c4:min f:min7
}
```

Cm Fm<sup>7</sup>

C- F-<sup>7</sup>

#### chordPrefixSpacer

El modificador de los acordes menores tal y como viene determinado por `minorChordModifier` se imprime por lo general inmediatamente a la derecha de la fundamental. Se puede insertar un símbolo separador entre la fundamental y el modificador estableciendo `chordPrefixSpacer`. El símbolo separador no se usa cuando el acorde está alterado.

### Instrucciones predefinidas

`\whiteTriangleMarkup`, `\blackTriangleMarkup`, `\germanChords`, `\semiGermanChords`, `\italianChords`, `\frenchChords`.

### Fragmentos de código seleccionados

#### *Excepciones para los nombres de acorde*

Se puede usar la propiedad `chordNameExceptions` para almacenar una lista de notaciones espaciales para acordes específicos.

```
% modify maj9 and 6(add9)
% Exception music is chords with markups
chExceptionMusic = {
  <c e g b d'>1-\markup { \super "maj9" }
  <c e g a d'>1-\markup { \super "6(add9)" }
}

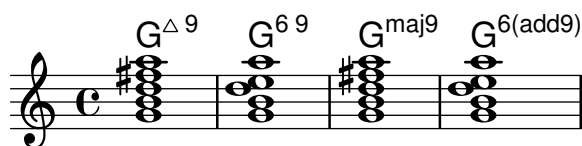
% Convert music to list and prepend to existing exceptions.
chExceptions = #(append
  (sequential-music-to-chord-exceptions chExceptionMusic #t)
  ignatzekExceptions)

theMusic = \chordmode {
  g1:maj9 g1:6.9
  \set chordNameExceptions = #chExceptions
  g1:maj9 g1:6.9
}

\layout {
  ragged-right = ##t
}

<<
  \new ChordNames \theMusic
  \new Voice \theMusic
```

&gt;&gt;



### *Nombre de acorde maj7*

La presentación del acorde de séptima mayor se puede ajustar mediante `majorSevenSymbol`.

```
\chords {
  c:7+
  \set majorSevenSymbol = \markup { j7 }
  c:7+
}
```

$C^{\Delta} C^{j7}$

### *Añadir barras de compás al contexto de nombres de acorde (ChordNames)*

Para añadir indicaciones de línea divisoria dentro del contexto de los nombres de acorde `ChordNames`, incluya el grabador `Bar_engraver`.

```
\new ChordNames \with {
  \override BarLine.bar-extent = #'(-2 . 2)
  \consists "Bar_engraver"
}

\chordmode {
  f1:maj7 f:7 bes:7
}
```

$F^{\Delta} \mid F^7 \mid Bb^7 \mid$

### *Corchetes de primera y segunda vez debajo de los acordes*

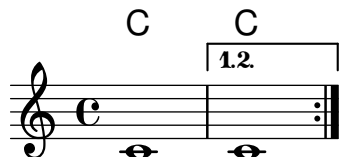
Mediante la adición del grabador `Volta_engraver` al pentagrama pertinente, se pueden poner los corchetes de primera y segunda vez debajo de los acordes.

```
\score {
  <<
  \chords {
    c1
    c1
  }
  \new Staff \with {
    \consists "Volta_engraver"
  }
  {
    \repeat volta 2 { c'1 }
    \alternative { c' }
  }
  >>
  \layout {
```

```

\context {
  \Score
  \remove "Volta_engraver"
}
}
}

```



### Modificación del separador de acordes

Se puede establecer el separador entre las distintas partes del nombre de un acorde para que sea cualquier elemento de marcado.

```

\chords {
  c:7sus4
  \set chordNameSeparator
    = \markup { \typewriter | }
  c:7sus4
}

```

$C^7 \text{ sus4}$   $C^7 | \text{ sus4}$

### Véase también

Referencia de la notación: Sección A.1 [Cuadro de nombres de acordes], página 695, Sección A.2 [Modificadores de acorde más usuales], página 696.

El ensayo sobre grabado musical automatizado: Sección “Lista de referencias bibliográficas” en *Ensayo*.

Archivos instalados: `scm/chords-ignatzek-names.scm`, `scm/chord-entry.scm`, `ly/chord-modifiers-init.ly`.

Fragmentos de código: Sección “Chords” en *Fragmentos de código*.

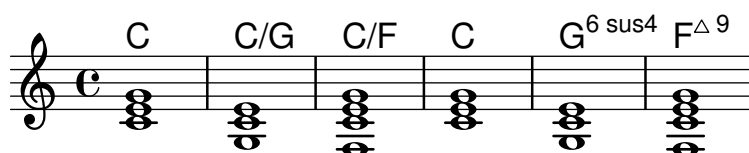
### Advertencias y problemas conocidos

Los nombres de acorde se determinan a partir de las notas que están presentes en el mismo y de la información acerca de la estructura del acorde que ha podido ser introducida en `\chordmode`. Si el método de las notas simultáneas se usa para introducir los acordes, pueden resultar nombres no deseados a partir de las inversiones o de las notas del bajo.

```

myChords = \relative c' {
  \chordmode { c1 c/g c/f }
  <c e g>1 <g c e> <f c' e g>
}
<<
  \new ChordNames { \myChords }
  \new Staff { \myChords }
>>

```







El soporte para bajo cifrado se compone de dos partes: hay un modo de entrada, presentado por `\figuremode`, que admite la entrada de las cifras del bajo, y hay un contexto llamado `FiguredBass` que se ocupa de la impresión de los objetos del tipo `BassFigure`. El bajo cifrado también se puede imprimir dentro de contextos de `Staff`.

`\figures { ... }` es una notación abreviada de `\new FiguredBass \figuremode { ... }`.

Aunque el soporte para bajo cifrado puede parecerse superficialmente al de los acordes, realmente es mucho más simple. El modo `\figuremode` simplemente almacena las cifras y el contexto `FiguredBass` las imprime tal y como se han introducido. No hay ninguna conversión a notas.

## Véase también

Glosario musical: Sección “figured bass” en *Glosario Musical*.

Fragmentos de código: Sección “Chords” en *Fragmentos de código*.

## Introducir el bajo cifrado

Se usa `\figuremode` para cambiar el modo de entrada al modo de cifras. Hay más información sobre los distintos modos de entrada en Sección 5.4.1 [Modos de entrada], página 660.

En el modo de cifras, un grupo de cifras del bajo está delimitado mediante ángulos simples, `< y >`. La duración se escribe después del `>`.

```
\new FiguredBass {
  \figuremode {
    <6 4>2
  }
}
```

**6**  
**4**

Se pueden usar alteraciones accidentales (incluso becuadros) para modificar grados de la escala. Se escriben añadiendo + (para los sostenidos), - (para los bemoles) o ! (para los becuadros) después del número. Para obtener alteraciones dobles, se aplica el modificador dos veces seguidas. Con frecuencia, para la modificación del tercer grado se omite el número, lo que puede obtenerse utilizando `_` en lugar de un número.

```
\figures {
  <7! 6+ 4-> <5++> <3--> < _+ > < 7 _!>
}
```

**7** **x5** **b3** **#** **7**  
**#6**  
**b4**

Se pueden indicar elementos del acorde aumentados y disminuidos:

```
\figures {
```

```
<6\+ 5/> <7/>
}
```

**+6 7**  
**5**

Se puede crear una barra invertida cruzando una cifra (usada normalmente para sextas elevadas):

```
\figures {
  <6> <6\\>
}
```

**6 6**

Se pueden incluir corchetes en las cifras:

```
\figures {
  <[12] 8 [6 4]>
}
```

**[12]**  
**8**  
**[6]**  
**[4]**

Se puede insertar cualquier elemento de marcado de texto como una cifra:

```
\figures {
  <\markup { \tiny \number 6 \super (1) } 5>
}
```

**6<sup>(1)</sup>**  
**5**

Se pueden usar líneas de continuación para indicar cifras que se repiten:

```
<<
{
  \clef bass
  e4 d c b,
  e4 d c b,
}
\figures {
  \bassFigureExtendersOn
  <6 4>4 <6 3> <7 3> <7 3>
  \bassFigureExtendersOff
  <6 4>4 <6 3> <7 3> <7 3>
}
>>
```



En este caso, las líneas de extensión sustituyen a las cifras existentes, a no ser que las líneas de continuación hayan sido terminadas explícitamente.

```
<<
```

```

\figures {
  \bassFigureExtendersOn
  <6 4>4 <6 4> <6\! 4\!> <6 4>
}
{
  \clef bass
  d4 d c c
}
>>

```



La siguiente tabla resume los modificadores de cifrado disponibles.

### ModificadorPropósito

### Ejemplo

**+**, **-**, **!** Alteraciones

**b7 x5 b3**  
**#6**  
**b4**

**\+**, **/** Aumentaciones y disminuciones

**+6 7**  
**5**

**\** Sexta elevada

**6**

**\!** Fin de línea de continuación



### Instrucciones predefinidas

`\bassFigureExtendersOn`, `\bassFigureExtendersOff`.

### Fragmentos de código seleccionados

#### *Cambiar las posiciones de las alteraciones del bajo cifrado*

Las alteraciones y los signos «más» pueden aparecer antes o después de los números, según el valor de las propiedades `figuredBassAlterationDirection` y `figuredBassPlusDirection`.

```

\figures {
  <6\+> <5+> <6 4-> r
  \set figuredBassAlterationDirection = #RIGHT
  <6\+> <5+> <6 4-> r
  \set figuredBassPlusDirection = #RIGHT
}

```



```

<6\+> <5+> <6 4-> r
\set figuredBassAlterationDirection = #LEFT
<6\+> <5+> <6 4-> r
}

+6 #5 6      +6 5# 6      6+ 5# 6      6+ #5 6
  ♭4          4♭          4♭          ♭4

```

## Véase también

Fragmentos de código: Sección “Chords” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “BassFigure” en *Referencia de Funcionamiento Interno*, Sección “BassFigureAlignment” en *Referencia de Funcionamiento Interno*, Sección “BassFigureLine” en *Referencia de Funcionamiento Interno*, Sección “BassFigureBracket” en *Referencia de Funcionamiento Interno*, Sección “BassFigureContinuation” en *Referencia de Funcionamiento Interno*, Sección “FiguredBass” en *Referencia de Funcionamiento Interno*.

## Imprimir el bajo cifrado

El bajo cifrado se puede imprimir usando el contexto `FiguredBass`, o dentro de la mayoría de los contextos de pentagrama.

Cuando se presenta dentro de un contexto de `FiguredBass`, la ubicación vertical de las cifras es independiente de las notas que están en el pentagrama.

```

<<
\relative {
  c' '4 c'8 r8 c,4 c'
}
\new FiguredBass {
  \figuremode {
    <4>4 <10 6>8 s8
    <6 4>4 <6 4>
  }
}
>>

```



En el ejemplo de arriba, el contexto `FiguredBass` se debe instanciar explícitamente para evitar crear un segundo pentagrama vacío.

El bajo cifrado se puede añadir también a contextos de `Staff` directamente. En este caso, la posición vertical de las cifras se ajusta automáticamente.

```

<<
\new Staff = "myStaff"
\figuremode {
  <4>4 <10 6>8 s8
  <6 4>4 <6 4>
}
%% Put notes on same Staff as figures
\context Staff = "myStaff" {
  \clef bass
}
>>

```

```

    c4 c'8 r8 c4 c'
  }
>>

```

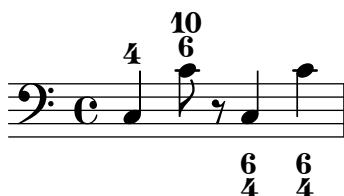


Cuando se escribe dentro de un contexto de pentagrama, el bajo cifrado se puede presentar encima o debajo del mismo.

```

<<
  \new Staff = "myStaff"
  \figuremode {
    <4>4 <10 6>8 s8
    \bassFigureStaffAlignmentDown
    <6 4>4 <6 4>
  }
  %% Put notes on same Staff as figures
  \context Staff = "myStaff" {
    \clef bass
    c4 c'8 r8 c4 c'
  }
>>

```



## Instrucciones predefinidas

`\bassFigureStaffAlignmentDown`, `\bassFigureStaffAlignmentUp`,  
`\bassFigureStaffAlignmentNeutral`.

## Véase también

Fragmentos de código: Sección “Chords” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “BassFigure” en *Referencia de Funcionamiento Interno*, Sección “BassFigureAlignment” en *Referencia de Funcionamiento Interno*, Sección “BassFigureLine” en *Referencia de Funcionamiento Interno*, Sección “BassFigureBracket” en *Referencia de Funcionamiento Interno*, Sección “BassFigureContinuation” en *Referencia de Funcionamiento Interno*, Sección “FiguredBass” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Para asegurar que las líneas extensoras funcionan adecuadamente, lo más seguro es utilizar las mismas duraciones en la línea de cifras y en la línea del bajo.

```

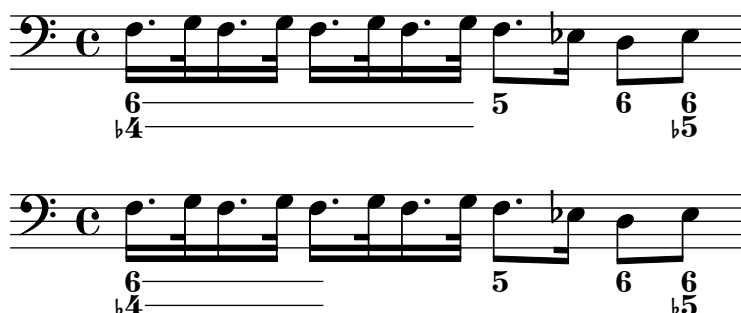
<<
{
  \clef bass
  \repeat unfold 4 { f16. g32 } f8. es16 d8 es
}

```

```

\figures {
  \bassFigureExtendersOn
  % The extenders are correct here, with the same rhythm as the bass
  \repeat unfold 4 { <6 4->16. <6 4->32 }
  <5>8. r16 <6>8 <6\! 5->
}
>>
<<
{
  \clef bass
  \repeat unfold 4 { f16. g32 } f8. es16 d8 es
}
\figures {
  \bassFigureExtendersOn
  % The extenders are incorrect here, even though the timing is the same
  <6 4->4 <6 4->4
  <5>8. r16 <6>8 <6\! 5->
}
>>

```



## 2.8 Música contemporánea

A partir de comienzos del s.XX se ha producido una expansión masiva de los estilos y técnicas composicionales. Los nuevos desarrollos armónicos y rítmicos, una expansión del espectro de alturas y el desarrollo de un amplio abanico de nuevas técnicas instrumentales han venido acompañadas de una evolución y expansión paralelas de la notación musical. El objetivo de esta sección es proporcionar referencias e información relevante para el trabajo con estas nuevas técnicas notacionales.

### 2.8.1 Altura y armonía en la música contemporánea

Esta sección destaca los problemas de la notación de las alturas y la armonía en la música contemporánea.

#### Referencias para alturas y armonía en música contemporánea

- La notación estándar de cuartos de tono se estudia en [Nombres de las notas en otros idiomas], página 8.
- Las armaduras de tonalidad no estándar se estudian en [Armadura de la tonalidad], página 23.
- Las prácticas contemporáneas en la presentación de las alteraciones accidentales se estudian en [Alteraciones accidentales automáticas], página 30.

#### Notación microtonal

## Armonía y armaduras de tonalidad contemporáneas

### 2.8.2 Enfoques contemporáneos del ritmo

Esta sección destaca los problemas de la notación de las duraciones en la música contemporánea.

#### Referencias para enfoque contemporáneos del ritmo

- Los tipos de compás de amalgama se estudian en [Indicación de compás], página 70.
- La notación polimétrica básica se estudia en [Notación polimétrica], página 81.
- Las barras en ángulo progresivas se estudian en [Barras progresivas], página 103.
- Las líneas divisorias de estilo Mensurstriche (líneas que están solamente entre los pentagramas) se estudian en [Agrupar pentagramas], página 202.

## Grupos de valoración especial en música contemporánea

### Compases contemporáneos

### Notación polimétrica extendida

### Barrado en música contemporánea

### Líneas divisorias en música contemporánea

### 2.8.3 Notación gráfica

### 2.8.4 Técnicas de partitura contemporáneas

### 2.8.5 Nuevas técnicas instrumentales

### 2.8.6 Lecturas adicionales y partituras de interés

Esta sección sugiere libros, ejemplos musicales y otros recursos de utilidad en el estudio de la notación musical contemporánea.

#### Libros y artículos sobre notación musical contemporánea

- *Music Notation in the Twentieth Century: A Practical Guidebook* de Kurt Stone [W. W. Norton, 1980]
- *Music Notation: A Manual of Modern Practice* de Gardner Read [Taplinger, 1979]
- *Instrumentation and Orchestration* de Alfred Blatter [Schirmer, 2nd ed. 1997]

### Partituras y ejemplos musicales

## 2.9 Notación antigua





rá-mus, ge-mén-tes et flen-tes in hac la-cri-má-rum val-le.

E-ia er-go, Ad-vo-cá-ta no-stra, il-los tu-os mi-se-ri-cór-

des ó-cu-los ad nos con-vér-te. Et Je-sum, be-ne-díc-tum fruc-

tum ven-tris tu-i, no-bis post hoc ex-sí-li-um os-tén-de.

O cle-mens: O pi-a: O dul-cis Vir-go Ma-rí-a.

Entre las formas en que está contemplada la notación antigua se encuentra la posibilidad de tipografiar notación mensural, canto gregoriano y canto kievano en notación cuadrada. Se puede acceder a estas funcionalidades mediante la modificación de propiedades de estilo de objetos gráficos como la cabeza de las notas o los silencios, o bien mediante la utilización de uno de los contextos predefinidos para estos estilos.

Muchos objetos gráficos, como la cabeza y el corchete de las figuras, las alteraciones, la indicación de compás y los silencios ofrecen una propiedad **style** (estilo), que se puede cambiar para emular varios estilos diferentes de notación antigua. Véase

- [Cabezas de nota de la música mensural], página 471,
- [Alteraciones y armaduras de la música mensural], página 473,
- [Silencios de la música mensural], página 472,
- [Claves de la música mensural], página 469,
- [Claves de canto gregoriano], página 476,
- [Corchetes de la música mensural], página 471,
- [Indicaciones de compás de la música mensural], página 470.

Ciertos conceptos de notación se incluyen específicamente para la notación antigua:

- [Custos], página 466,
- [Divisiones], página 478,
- [Ligaduras], página 466.

## Véase también

Glosario musical: Sección “custos” en *Glosario Musical*, Sección “ligature” en *Glosario Musical*, Sección “mensural notation” en *Glosario Musical*.

Referencia de la notación: [Cabezas de nota de la música mensural], página 471, [Alteraciones y armaduras de la música mensural], página 473, [Silencios de la música mensural], página 472, [Claves de canto gregoriano], página 476, [Corchetes de la música mensural], página 471, [Indicaciones de compás de la música mensural], página 470, [Custos], página 466, [Divisiones], página 478, [Ligaduras], página 466.

### 2.9.1 Panorámica de los estilos contemplados

Están disponibles tres estilos para la composición tipográfica del canto gregoriano:

- *Editio Vaticana* es un estilo completo para canto gregoriano, que sigue la apariencia de las ediciones de Solesmes, los libros de canto oficiales del Vaticano desde 1904. LilyPond contempla todos los signos de notación que se utilizan en este estilo, entre ellos las ligaduras, *custodias* y símbolos especiales como el quilisma y el oriscus.
- El estilo *Editio Medicaea* ofrece ciertas funcionalidades que se usan en las ediciones Medicaea (o de Ratisbona) que se usaban con anterioridad a las ediciones de Solesmes. Las diferencias más significativas con respecto al estilo *Vaticana* son las claves, que tienen trazos descendentes, y la cabeza de las figuras, que son cuadradas y regulares.
- El estilo *Hufnagel* (en “uña de herradura”) o *gótico* reproduce el estilo de escritura en los manuscritos de canto medievales de Alemania y Centro Europa. Recibe el nombre de la forma de la nota básica (la *virga*), que parece una pequeña uña.

Tres estilos emulan la apariencia de los manuscritos y ediciones impresas de música mensural tardomedievales y renacentistas:

- El estilo *Mensural* se asemeja bastante al estilo de escritura utilizado en los manuscritos tardomedievales y renacentistas tempranos, con la cabeza de sus figuras en forma de rombo pequeñas y estrechas, y los silencios que imitan un estilo dibujado a mano.
- El estilo *Neomensural* es una versión modernizada y estilizada del anterior: la cabeza de las figuras es más ancha y los silencios están formados por trazos rectilíneos. Este estilo está particularmente indicado, p.ej., para los *incipits* de transcripciones de fragmentos de música mensural.
- El estilo *Petrucchi* recibe su nombre de Ottaviano Petrucci (1466-1539), el primer impresor en utilizar tipos móviles para la música (en su *Harmonice musices odhecaton*, 1501). Este estilo utiliza para las figuras una cabeza más grande que los otros estilos mensurales.

*Baroque* y *Classical* no son estilos completos, sino que difieren del estilo predeterminado solamente en ciertos detalles: algunas cabezas de nota (Baroque) y el silencio de negra (Classical).

Sólo el estilo mensural tiene alternativas para todos los aspectos de la notación. Así, no existen silencios ni corchetes en los estilos gregorianos, pues estos símbolos no se usan en la notación del canto llano, y el estilo Petrucci no posee corchetes ni alteraciones por sí mismo.

Cada elemento de la notación se puede cambiar independientemente de los otros, y así podemos usar corchetes mensurales, cabezas de petrucci, silencios de classical y claves de vaticana en la misma pieza, si queremos.

### Véase también

Glosario musical: Sección “mensural notation” en *Glosario Musical*, Sección “flag” en *Glosario Musical*.

### 2.9.2 Notación antigua - funcionalidades comunes

#### Contextos predefinidos

Para el canto gregoriano y la notación mensural, están a nuestra disposición contextos predefinidos de voz y de pauta, que establecen todos los símbolos de notación a unos valores adecuados para estos estilos. Si nos satisfacen estos valores predeterminados, podemos a continuación introducir directamente las notas sin la necesidad de preocuparnos de los detalles sobre cómo personalizar un contexto. Véanse uno de los contextos predefinidos *VaticanaVoice*, *VaticanaStaff*, *MensuralVoice* y *MensuralStaff*.

- [Contextos del canto gregoriano], página 476,
- [Contextos de la música mensural], página 468.

## Véase también

Glosario musical: Sección “mensural notation” en *Glosario Musical*.

Referencia de la notación: [Contextos del canto gregoriano], página 476, [Contextos de la música mensural], página 468.

## Ligaduras

Una ligadura es un símbolo gráfico que representa al menos dos notas distintas. Las ligaduras aparecieron originalmente en los manuscritos de la notación del canto gregoriano para denotar secuencias de notas ascendentes o descendentes en la misma sílaba. Se usan también en notación mensural.

Las ligaduras se escriben *encerrándolas* dentro de un par `\[` y `\]`. Ciertos estilos de ligadura podrían necesitar una sintaxis de entrada adicional específica para este tipo de ligadura en particular. De forma predeterminada, el grabador `LigatureBracket` se limita a colocar un corchete recto sobre la ligadura:

```
\relative {
  \[ g' c, a' f d' \]
  a g f
  \[ e f a g \]
}
```



Están disponibles otros dos estilos de ligaduras: el Vaticano para canto gregoriano, y el Mensural para música mensural (sólo están contempladas las ligaduras mensurales blancas para la música mensural, y con ciertas limitaciones). Para usar cualquiera de estos estilos, se debe sustituir el grabador predeterminado `Ligature_bracket_engraver` por uno de los grabadores especializados en ligaduras del contexto `Voice`, como se explica en [Ligaduras mensurales blancas], página 474, y [Ligaduras de neumas cuadrados gregorianos], página 480.

## Véase también

Glosario musical: Sección “ligature” en *Glosario Musical*.

Referencia de la notación: [Ligaduras mensurales blancas], página 474, [Ligaduras de neumas cuadrados gregorianos], página 480.

## Advertencias y problemas conocidos

El espacio que las ligaduras necesitan no está implementado actualmente, y como resultado, puede producirse una separación excesiva entre ellas. Los saltos de línea podrían también ser poco satisfactorios.

La estra podría no alinearse con la música de la forma esperada cuando se utilizan ligaduras.

No se deben imprimir alteraciones accidentales dentro de una ligadura, sino acumularse e imprimirse delante de ella.

La sintaxis aún utiliza el estilo obsoleto “infijo” `\[ expresión musical \]`. Por motivos de consistencia, se cambiará tarde o temprano al estilo “postfijo” `nota\[ ... nota\]`.

## Custos

Un *custos* (en plural: *custodias*, que en latín significa “guarda”) es un símbolo que aparece al final de una pauta. Anticipa la altura de la primera o primeras notas de la línea siguiente, ayudando al intérprete con los saltos de línea durante la ejecución.

Las custodias se utilizaron mucho en la notación musical hasta el siglo XVII. En nuestros días, sobreviven sólo en algunas formas particulares de notación musical como las ediciones contemporáneas de canto gregoriano como la *Editio Vaticana*. Hay distintos glifos para el custos que se utilizan en las diferentes variaciones de estilo notacional.

Para tipografiar las custodias, simplemente coloque un grabador `Custos_engraver` dentro del contexto `Staff` (el contexto del pentagrama) al declarar el bloque de `\layout` (el bloque de disposición de la página), y modifique el estilo del custos con una instrucción `\override` si lo desea, como se muestra en el ejemplo siguiente:

```
\score {
  \relative {
    a'1
    \break
    g
  }
  \layout {
    \context {
      \Staff
      \consists "Custos_engraver"
      \override Custos.style = #'mensural
    }
  }
}
```



El glifo del custos se selecciona mediante la propiedad `style`. Los estilos contemplados son `vaticana`, `medicaea`, `hufnagel` y `mensural`.

```
\new Lyrics \lyricmode {
  \markup { \column {
    \typewriter "vaticana "
    \line { " " \musicglyph "custodes.vaticana.u0" }
  } }
  \markup { \column {
    \typewriter "medicaea "
    \line { " " \musicglyph "custodes.medicaea.u0" }
  } }
  \markup { \column {
    \typewriter "hufnagel "
    \line { " " \musicglyph "custodes.hufnagel.u0" }
  } }
  \markup { \column {
    \typewriter "mensural "
    \line { " " \musicglyph "custodes.mensural.u0" }
  } }
}
```



}

vaticana medicaea hufnagel mensural



## Véase también

Glosario musical: Sección “custos” en *Glosario Musical*.

Fragmentos de código: Sección “Ancient notation” en *Fragmentos de código*.

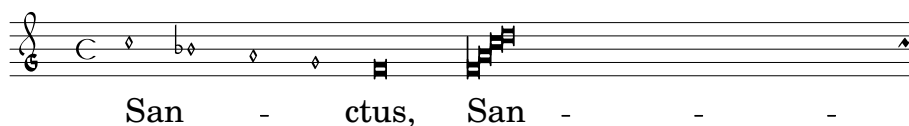
Referencia de funcionamiento interno: Sección “Custos” en *Referencia de Funcionamiento Interno*.

## 2.9.3 Tipografiar música mensural

### Contextos de la música mensural

Los contextos predefinidos de voz `MensuralVoice` y de pauta `MensuralStaff` se pueden usar para grabar una pieza en estilo mensural. Estos contextos establecen a unos valores iniciales adecuados todas las propiedades relevantes de los contextos y de los objetos gráficos, de manera que pueda comenzar inmediatamente a escribir el canto, como lo demuestra el siguiente fragmento:

```
\score {
  <<
    \new MensuralVoice = "discantus" \relative {
      \hide Score.BarNumber {
        c'1\melisma bes a g\melismaEnd
        f\breve
        \[ f1\melisma a c\breve d\melismaEnd \]
        c\longa
        c\breve\melisma a1 g1\melismaEnd
        fis\longa^\signumcongruentiae
      }
    }
    \new Lyrics \lyricsto "discantus" {
      San -- ctus, San -- ctus, San -- ctus
    }
  >>
}
```



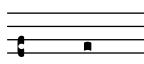
## Véase también

Glosario musical: Sección “mensural notation” en *Glosario Musical*.

## Claves de la música mensural

Están contempladas las claves mensurales usando la instrucción `\clef`. Algunas de las claves utilizan el mismo glifo, pero se diferencian solamente respecto a la línea en la que se imprimen. En tales casos, se utiliza un sufijo numérico en el nombre para denominar dichas claves, contando las líneas desde la inferior hasta la superior.

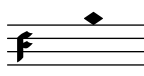
```
\override NoteHead.style = #'vaticana.punctum
\clef "vaticana-do1"
c'1
```



```
\override NoteHead.style = #'medicaea.punctum
\clef "medicaea-do3"
c'1
```



```
\override NoteHead.style = #'hufnagel.punctum
\clef "hufnagel-fa2"
c'1
```



```
\override NoteHead.style = #'neomensural
\clef "neomensural-c4"
c'1
```



Se puede forzar manualmente la composición tipográfica de un glifo de clave sobre cualquier línea seleccionada arbitrariamente, según se describe en [Clave], página 18. Para ver el abanico completo de las claves posibles, consulte Sección A.10 [Estilos de clave], página 734.

## Véase también

Glosario musical: Sección “mensural notation” en *Glosario Musical*, Sección “clef” en *Glosario Musical*.

Referencia de la notación: [Claves de canto gregoriano], página 476, [Clave], página 18.

Archivos instalados: `scm/parser-clef.scm`.

Fragmentos de código: Sección “Pitches” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Clef\_engraver” en *Referencia de Funcionamiento Interno*, Sección “Clef” en *Referencia de Funcionamiento Interno*, Sección “ClefModifier” en *Referencia de Funcionamiento Interno*, Sección “clef-interface” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

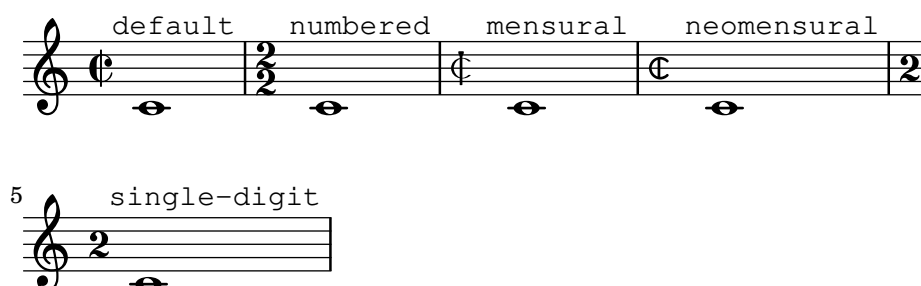
La clave de Sol Mensural está asignada a la clave de Sol de Petrucci.

## Indicaciones de compás de la música mensural

Los símbolos de mensuración (que son algo similar, pero no exactamente igual que las indicaciones de compás) están contemplados de forma limitada. Los glifos están configurados de forma fija para determinadas fracciones de tiempo. En otras palabras, para obtener un símbolo de mensuración determinado con la instrucción `\time n/m`, se tienen que escoger  $n$  y  $m$  según la siguiente tabla

<code>\time 4/4</code>	<code>\time 2/2</code>	<code>\time 6/4</code>	<code>\time 6/8</code>
<code>\time 3/2</code>	<code>\time 3/4</code>	<code>\time 9/4</code>	<code>\time 9/8</code>
<code>\time 4/8</code>	<code>\time 2/4</code>		

Utilice la propiedad `style` del elemento gráfico `TimeSignature` para seleccionar las indicaciones de compás de la música antigua. Los estilos contemplados son `neomensural` y `mensural`. La tabla anterior utiliza el estilo `neomensural`. Los ejemplos siguientes muestran las diferencias que existen entre los estilos:



[Indicación de compás], página 70, ofrece una introducción general al uso de las indicaciones de compás.

## Véase también

Glosario musical: Sección “mensural notation” en *Glosario Musical*.

Referencia de la notación: [Indicación de compás], página 70.

## Advertencias y problemas conocidos

Las relaciones de duración de las notas no pueden cambiar con la indicación de compás, porque no son constantes. Por ejemplo, la razón de 1 breve = 3 semibreves (*tempus perfectum*) se puede hacer a mano mediante el establecimiento de

```
breveTP = #(ly:make-duration -1 0 3/2)
...
{ c\breveTP f1 }
```

Esto fija un valor para `breveTP` de  $3/2$  multiplicado por  $2 = 3$  veces una redonda.

Los símbolos `mensural68alt` y `neomensural68alt` (alternativos al símbolo de  $6/8$ ) no se puede seleccionar directamente con `\time`. En su lugar, utilice `\markup {\musicglyph "timesig.mensural68alt" }`.

## Cabezas de nota de la música mensural

Para la notación antigua, se puede elegir un estilo de cabezas de notas distinto al `default` (estilo predeterminado). Esto se consigue estableciendo la propiedad `style` del objeto `NoteHead` a los valores `baroque`, `neomensural`, `mensural`, `petrucci`, `blackpetrucci` o `semipetrucci`.

El estilo `baroque` se diferencia del estilo predeterminado `default` en que:

- Proporciona una cabeza de nota para la `maxima`, y
- Usa una forma cuadrada para las cabezas de `\breve`.

Los estilos `neomensural`, `mensural` y `petrucci` se diferencian del estilo `baroque` en que:

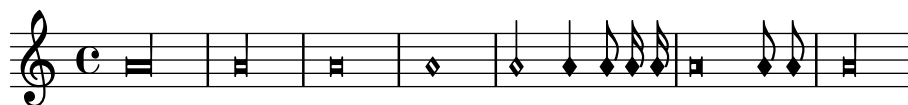
- Usan cabezas romboidales para las semibreves y todas las notas más cortas, y
- Centran las plicas sobre las cabezas.

El estilo `blackpetrucci` produce notas con una cabeza que puede usarse en notación mensural negra o en secciones de coloratio dentro de la notación mensural blanca. A causa de que el estilo de la cabeza de las notas no influye en el número de corchetes, en este estilo la notación de la semimínima debe hacer como `a8*2`, no como `a4`, pues de lo contrario tendrá el aspecto de una mínima. El multiplicador puede ser distinto si se usa la coloratio, es decir, para la notación de tresillos.

Utilice el estilo `semipetrucci` para dibujar notas con cabeza semi-rellena (breves, longas y máximas).

El ejemplo siguiente muestra el estilo `petrucci`:

```
\compressEmptyMeasures
\autoBeamOff
\override NoteHead.style = #'petrucci
a'\maxima a'\longa a'\breve a'1 a'2 a'4 a'8 a'16 a'
\override NoteHead.style = #'semipetrucci
a'\breve*5/6
\override NoteHead.style = #'blackpetrucci
a'8*4/3 a'
\override NoteHead.style = #'petrucci
a'\longa
```



Sección A.9 [Estilos de cabezas de nota], página 734, ofrece una visión de conjunto sobre la totalidad de los estilos de cabeza disponibles.

## Véase también

Glosario musical: Sección “mensural notation” en *Glosario Musical*, Sección “note head” en *Glosario Musical*.

Referencia de la notación: Sección A.9 [Estilos de cabezas de nota], página 734.

## Corchetes de la música mensural

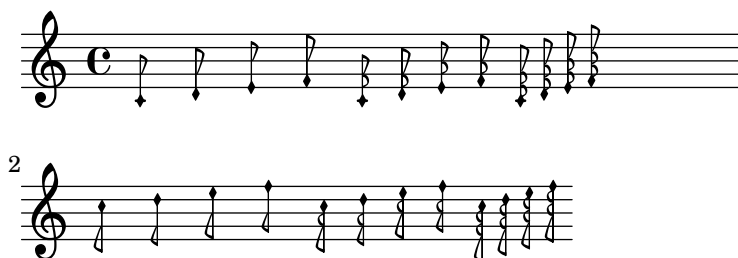
Utilice la propiedad `flag-style` del elemento gráfico `Stem` para seleccionar los corchetes de estilo antiguo. Aparte del estilo de corchete por defecto `default`, sólo está contemplado el estilo `mensural`

```
\relative c' {
  \override Flag.style = #'mensural
```

```

\override Stem.thickness = #1.0
\override NoteHead.style = #'mensural
\autoBeamOff
c8 d e f c16 d e f c32 d e f s8
c'8 d e f c16 d e f c32 d e f
}

```



Observe que la bandera más interna de cada uno de los corchetes mensurales se alinea verticalmente con una línea de la pauta.

No existe un estilo particular de corchete para la notación neo-mensural ni Petrucci. No existen corchetes en la notación del canto gregoriano.

## Véase también

Glosario musical: Sección “mensural notation” en *Glosario Musical*, Sección “flag” en *Glosario Musical*.

## Advertencias y problemas conocidos

La alineación vertical de cada uno de los corchetes con una línea de la pauta da por supuesto que las plicas siempre terminan exactamente encima o bien exactamente en el medio de dos líneas de la pauta. Esto puede no ser siempre cierto cuando se utilizan las posibilidades de disposición avanzadas de la notación clásica (que, de todas formas, por lo común se encuentra fuera del ámbito de la notación mensural).

## Silencios de la música mensural

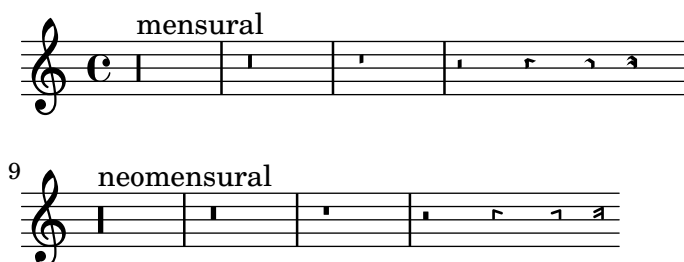
Utilice la propiedad `style` del elemento gráfico `Rest` para seleccionar silencios antiguos. Los estilos de música antigua contemplados son `classical`, `neomensural` y `mensural`.

El ejemplo siguiente muestra estos estilos:

```

\compressEmptyMeasures
\override Rest.style = #'mensural
r\longa^"mensural" r\breve r1 r2 r4 r8 r16 s \break
\override Rest.style = #'neomensural
r\longa^"neomensural" r\breve r1 r2 r4 r8 r16

```



No existen silencios de fusa y semifusa específicos para los estilos mensural ni neo-mensural. Se usan los silencios del estilo predeterminado.

## Véase también

Glosario musical: Sección “mensural notation” en *Glosario Musical*.

Referencia de la notación: [Silencios], página 62.

Fragmentos de código: Sección “Ancient notation” en *Fragmentos de código*.

## Advertencias y problemas conocidos

El glifo para el silencio de máxima en el estilo mensural es en realidad un silencio de longa perfecta; use dos (o tres) silencios de longa para imprimir un silencio de máxima. Los silencios de longa no se agrupan automáticamente, por lo que debe hacerse manualmente utilizando silencios con altura determinada.

## Alteraciones y armaduras de la música mensural

El estilo `mensural` ofrece unos símbolos de sostenido y bemol distintos de los del estilo predeterminado. La notación mensural rara vez utilizaba el becuadro: en su lugar, se usaba el sostenido o bemol correspondiente. Por ejemplo, un Si natural en la tonalidad de Fa mayor se indicaría con un sostenido. Sin embargo, si se solicita explícitamente, el símbolo de becuadro se extrae del estilo `vaticana`.

### mensural

♭ ✕

El estilo de las alteraciones y de la armadura de la tonalidad se controlan a través de la propiedad `alteration-glyph-name-alist` de los elementos gráficos `Accidental` y `KeySignature`, respectivamente; p.ej.:

```
\override Staff.Accidental.alteration-glyph-name-alist =
  #alteration-mensural-glyph-name-alist
```

## Véase también

Glosario musical: Sección “mensural notation” en *Glosario Musical*, Sección “Pitch names” en *Glosario Musical*, Sección “accidental” en *Glosario Musical*, Sección “key signature” en *Glosario Musical*.

Referencia de la notación: Sección 1.1 [Alturas], página 1, [Alteraciones accidentales], página 6, [Alteraciones accidentales automáticas], página 30, [Armadura de la tonalidad], página 23.

Referencia de funcionamiento interno: Sección “KeySignature” en *Referencia de Funcionamiento Interno*.

## Alteraciones de anotación (*musica ficta*)

En la música europea de hasta aprox. 1600, se esperaba de los cantantes que alterasen cromáticamente ciertas notas según su propia iniciativa y de acuerdo a ciertas reglas. Esto recibe el nombre de *musica ficta*. En las transcripciones modernas, estas alteraciones accidentales se imprimen normalmente encima de la nota.

Están contempladas estas alteraciones sugeridas, y se pueden activar estableciendo `suggestAccidentals` al valor verdadero.

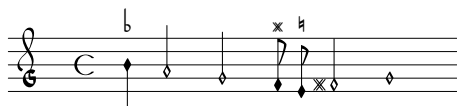
```
\relative {
  fis' gis
  \set suggestAccidentals = ##t
  ais bis
```

}



Esto hará que se traten *todas* las alteraciones siguientes como *musica ficta* hasta que se desactive con `\set suggestAccidentals = ##f`. Una forma más práctica es usar `\once \set suggestAccidentals = ##t`, que puede incluso definirse como una forma breve muy conveniente:

```
ficta = { \once \set suggestAccidentals = ##t }
\score { \relative
  \new MensuralVoice {
    \once \set suggestAccidentals = ##t
    bes'4 a2 g2 \ficta fis8 \ficta e! fis2 g1
  }
}
```



## Véase también

Referencia de funcionamiento interno: Sección “Accidental\_engraver” en *Referencia de Funcionamiento Interno*, Sección “AccidentalSuggestion” en *Referencia de Funcionamiento Interno*.

## Ligaduras mensurales blancas

Las ligaduras mensurales blancas están contempladas de forma limitada.

Para grabar ligaduras mensurales blancas, en el bloque layout o de diseño de página reemplace el grabador `Ligature_bracket_engraver` por el grabador `Mensural_ligature_engraver` en el contexto `Voice`:

```
\layout {
  \context {
    \Voice
    \remove "Ligature_bracket_engraver"
    \consists "Mensural_ligature_engraver"
  }
}
```

No existe ningún lenguaje de entrada adicional para describir la forma de una ligadura mensural blanca. Más bien la forma se determina exclusivamente a partir de la altura y duración de las notas comprendidas. Si bien este enfoque podría hacer que un usuario con poca experiencia tardase un tiempo en acostumbrarse a él, tiene la gran ventaja de que toda la información musical de la ligadura se conoce internamente. Esto no se requiere solamente para una salida MIDI correcta, sino que también permite la transcripción automática de las ligaduras.

En algunos lugares, dos notas consecutivas se pueden representar bien como dos cuadrados, bien como un paralelogramo oblicuo (forma de flexa). En tales casos la forma por omisión es dos cuadrados, pero puede requerirse una flexa fijando la propiedad `ligature-flexa` de la *segunda* cabeza. La longitud de una flexa se puede fijar mediante la propiedad `flexa-width` de la cabeza.

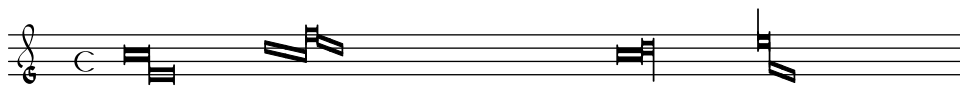
Por ejemplo:

```
\score {
  \relative {
```

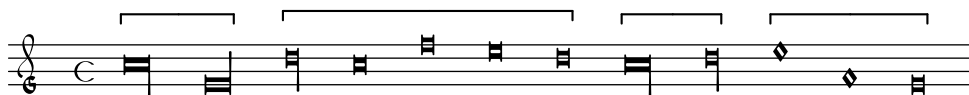
```

\set Score.timing = ##f
\set Score.defaultBarType = "-"
\override NoteHead.style = #'petrucci
\override Staff.TimeSignature.style = #'mensural
\clef "petrucci-g"
\[ c'\maxima g \]
\[ d'\longa
  \override NoteHead.ligature-flexa = ##t
  \once \override NoteHead.flexa-width = #3.2
  c\breve f e d \]
\[ c\maxima d\longa \]
\[ e1 a, g\breve \]
}
\layout {
  \context {
    \Voice
    \remove "Ligature_bracket_engraver"
    \consists "Mensural_ligature_engraver"
  }
}
}

```



Si no se sustituye el grabador `Ligature_bracket_engraver` por el `Mensural_ligature_engraver`, la misma música presenta la siguiente apariencia:



## Véase también

Glosario musical: Sección “ligature” en *Glosario Musical*.

Referencia de la notación: [Ligaduras de neumas cuadrados gregorianos], página 480, [Ligaduras], página 466.

## Advertencias y problemas conocidos

El espaciado horizontal de las ligaduras puede ser algo pobre. Las alteraciones accidentales pueden colisionar con las notas previas.

### 2.9.4 Tipografiado del canto gregoriano

Al componer tipográficamente una pieza en notación de canto gregoriano, el grabador `Vaticana_ligature_engraver` selecciona automáticamente la cabeza adecuada de las figuras, de manera que no es necesario establecer explícitamente el estilo de las cabezas. A pesar de ello se puede establecer el estilo de las cabezas, p.ej.: a `vaticana_punctum` para producir neumas de punctum. De forma similar, el grabador `Mensural_ligature_engraver` construye ligaduras mensurales automáticamente. Consulte [Ligaduras], página 466, para ver cómo funcionan los grabadores de ligaduras.



## Véase también

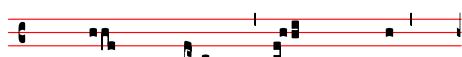
Glosario musical: Sección “ligature” en *Glosario Musical*.

Referencia de la notación: [Ligaduras mensurales blancas], página 474, [Ligaduras], página 466.

## Contextos del canto gregoriano

El contexto predefinido de voz `VaticanaVoice` y el de pauta `VaticanaStaff` se pueden utilizar para grabar una pieza de canto gregoriano en el estilo de la Editio Vaticana. Estos contextos establecen a unos valores iniciales adecuados todas las propiedades relevantes de los contextos y de los objetos gráficos, de forma que pueda inmediatamente comenzar a escribir el canto en sí, como lo demuestra el siguiente fragmento:

```
\include "gregorian.ly"
\score {
  <<
    \new VaticanaVoice = "cantus" {
      \[ c'\melisma c' \flexa a \]
      \[ a \flexa \deminutum g\melismaEnd \]
      f \divisioMinima
      \[ f\melisma \pes a c' c' \pes d'\melismaEnd \]
      c' \divisioMinima \break
      \[ c'\melisma c' \flexa a \]
      \[ a \flexa \deminutum g\melismaEnd \] f \divisioMinima
    }
    \new Lyrics \lyricsto "cantus" {
      San- ctus, San- ctus, San- ctus
    }
  >>
}
```



San-ctus, San-ctus,

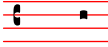
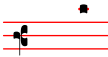

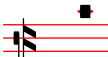
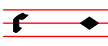
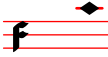



San-ctus

## Claves de canto gregoriano

La tabla siguiente muestra todas las claves gregorianas contempladas a través de la instrucción `\clef`. Algunas de las claves usan el mismo glifo, pero se diferencian solamente en la línea en que se imprimen. En estos casos, para enumerar estas claves se usa un número a continuación del nombre de la clave, contando desde la línea inferior. Aún se puede forzar manualmente la composición tipográfica de un glifo de clave sobre una línea arbitraria, como se describe en [Clave], página 18. La nota que se imprime a la derecha junto a cada una de las claves en la columna de los ejemplos, denota el Do central (c') respecto a dicha clave.

Descripción	Claves posibles	Ejemplo
-------------	-----------------	---------

Clave de Do, estilo Editio Vaticana	<code>vaticana-do1</code> , <code>vaticana-do2</code> , <code>vaticana-do3</code>	
Clave de Fa, estilo Editio Vaticana	<code>vaticana-fa1</code> , <code>vaticana-fa2</code>	
Clave de Do, estilo Editio Medicaea	<code>medicaea-do1</code> , <code>medicaea-do2</code> , <code>medicaea-do3</code>	
Clave de Fa, estilo Editio Medicaea	<code>medicaea-fa1</code> , <code>medicaea-fa2</code>	
Clave de Do, estilo hufnagel	<code>hufnagel-do1</code> , <code>hufnagel-do2</code> , <code>hufnagel-do3</code>	
Clave de Fa, estilo hufnagel	<code>hufnagel-fa1</code> , <code>hufnagel-fa2</code>	
Clave combinada de Do y Fa, estilo hufnagel	<code>hufnagel-do-fa</code>	

## Véase también

Glosario musical: Sección “clef” en *Glosario Musical*.

Referencia de la notación: [Clave], página 18.

## Alteraciones y armaduras de canto gregoriano

Están disponibles las alteraciones de los tres estilos gregorianos:

**vaticana medicaea hufnagel**

♭ ♯      ♮      ♭

Como se ve en el ejemplo, cada estilo no contempla todas las alteraciones. Al intentar acceder a una alteración que no está contemplada, LilyPond cambia a un estilo diferente.

El estilo de las alteraciones y las armaduras de tonalidad se controla a través de la propiedad `alteration-glyph-name-alist` de los grobs (objetos gráficos) Sección “Accidental” en *Referencia de Funcionamiento Interno* y Sección “KeySignature” en *Referencia de Funcionamiento Interno*, respectivamente; p.ej.:

```
\override Staff.Accidental.alteration-glyph-name-alist =
  #alteration-mensural-glyph-name-alist
```

## Véase también

Glosario musical: Sección “accidental” en *Glosario Musical*, Sección “key signature” en *Glosario Musical*.

Referencia de la notación: Sección 1.1 [Alturas], página 1, [Alteraciones accidentales], página 6, [Alteraciones accidentales automáticas], página 30, [Armadura de la tonalidad], página 23.

Referencia de funcionamiento interno: Sección “KeySignature” en *Referencia de Funcionamiento Interno*.

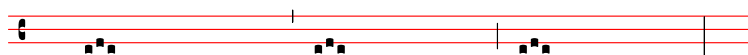
## Divisiones

En la notación del canto gregoriano no existen silencios, sino [Divisiones], página 478.

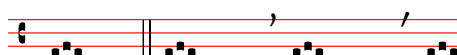
Una *divisio* (en plural: *divisiones*, que en latín significa ‘división’) es un símbolo del contexto del pentagrama que se usa para estructurar la música gregoriana en frases y secciones. El significado musical de *divisio minima*, *divisio maior* y *divisio maxima* se podrían caracterizar como una pausa corta, media y larga (respectivamente), algo así como las marcas de respiración de [Marcas de respiración], página 144. El signo de *finalis* no sólo marca el final de un canto, sino que se usa también con frecuencia dentro de un único canto antifonal o responsorial para marcar el final de cada una de las secciones.

Para usar divisiones, incluya el archivo `gregorian.ly`. Contiene definiciones que podrá aplicar con tan sólo escribir `\divisioMinima`, `\divisioMaior`, `\divisioMaxima` y `\finalis` en los lugares adecuados de la entrada. Algunas ediciones usan *virgula* o *caesura* en vez de *divisio minima*. Por ello, `gregorian.ly` define también las instrucciones `\virgula` y `\caesura`.

`divisio minima`   `divisio maior`   `divisio maxima`



`finalis`   `virgula`   `caesura`



## Instrucciones predefinidas

`\virgula`, `\caesura`, `\divisioMinima`, `\divisioMaior`, `\divisioMaxima`, `\finalis`.

## Véase también

Glosario musical: Sección “caesura” en *Glosario Musical*, Sección “divisio” en *Glosario Musical*.

Referencia de la notación: [Marcas de respiración], página 144.

Archivos de inicio: `ly/gregorian.ly`.

## Articulaciones del canto gregoriano

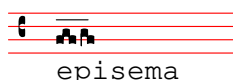
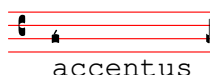
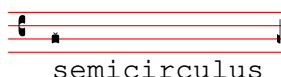
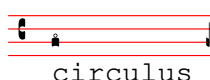
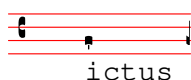
Además de los signos de articulación estándar que se hallan descritos en la sección [Articulaciones y ornamentos], página 128, se proveen signos de articulación diseñados específicamente para utilizarlos con la notación en el estilo de la *Editio Vaticana*.

```
\include "gregorian.ly"
\score {
  \new VaticanaVoice {
    \override TextScript.font-family = #'typewriter
```

```

\override TextScript.font-shape = #'upright
\override Script.padding = #-0.1
a\ictus_"ictus " \bar "" \break
a\circulus_"circulus " \bar "" \break
a\semicirculus_"semicirculus " \bar "" \break
a\accentus_"accentus " \bar "" \break
\[ a_"episema" \epistemInitium \pes b \flexa a b \epistemFinis \flexa a \]
}
}

```



## Véase también

Referencia de la notación: [Articulaciones y ornamentos], página 128.

Fragmentos de código: Sección “Ancient notation” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Episema” en *Referencia de Funcionamiento Interno*, Sección “EpisemaEvent” en *Referencia de Funcionamiento Interno*, Sección “Episema\_engraver” en *Referencia de Funcionamiento Interno*, Sección “Script” en *Referencia de Funcionamiento Interno*, Sección “ScriptEvent” en *Referencia de Funcionamiento Interno*, Sección “Script\_engraver” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Ciertas articulaciones se colocan demasiado cerca verticalmente de las cabezas de nota correspondientes.

## Puntos de aumentación (*morae*)

Los puntos de aumentación, llamados también *morae*, se escriben con la función musical `\augmentum`. Observe que `\augmentum` está programado como una función musical unaria y no como un prefijo de cabeza. Se aplica solamente a la expresión musical que sigue inmediatamente. Esto es, `\augmentum \virga c` no tendrá ningún efecto visible. En su lugar, escriba `\virga \augmentum c` o `\augmentum {\virga c}`. Tenga en cuenta también que puede escribir `\augmentum {a g}` como abreviatura de `\augmentum a \augmentum g`.

```

\include "gregorian.ly"
\score {
  \new VaticanaVoice {
    \[ \augmentum a \flexa \augmentum g \]
  }
}

```

```

    \augmentum g
  }
}

```



## Véase también

Referencia de la notación: [Marcas de respiración], página 144.

Referencia de funcionamiento interno: Sección “BreathingSign” en *Referencia de Funcionamiento Interno*.

Fragmentos de código: Sección “Ancient notation” en *Fragmentos de código*.

## Ligaduras de neumas cuadrados gregorianos

La notación de neumas cuadrados gregorianos está contemplada de forma limitada (siguiendo el estilo de la Editio Vaticana). El núcleo principal del conjunto de las ligaduras ya se pueden tipografiar, pero aún faltan aspectos esenciales para una composición tipográfica seria, como (entre otros) la alineación horizontal de varias ligaduras, la alineación de la letra y un correcto manejo de las alteraciones accidentales.

Se habilita el soporte de los neumas gregorianos por medio de la inclusión con `\include` del archivo `gregorian.ly` al principio del archivo. Esto hace que estén disponibles una cierta cantidad de instrucciones adicionales para producir los símbolos de los neumas que se usan en la notación de canto llano.

Las cabezas de las notas se pueden *modificar* y/o *unir*.

- La forma de la cabeza se puede modificar *precediendo* el nombre de la nota con una cualquiera de las instrucciones siguientes: `\virga`, `\strophæ`, `\inclinatum`, `\auctum`, `\descendens`, `\ascendens`, `\oriscus`, `\quilisma`, `\deminutum`, `\cavum`, `\linea`.
- Las ligaduras hablando en propiedad, (es decir, notas unidas), se producen escribiendo una de las instrucciones de unión `\pes` o `\flexa`, para el movimiento ascendente y descendente, respectivamente, *intercaladas* entre las notas que se han de unir.

Una nota sin modificaciones produce un *punctum*. Todos los demás neumas, incluso los neumas de una nota con forma distinta como la *virga*, se consideran en principio como ligaduras y por ello se deben escribir dentro de `\[...]`.

Neumas de una nota:

- El *punctum* es la forma básica de nota (en el estilo *Vaticana*: un cuadrado con una ligera curvatura a modo de excelencia tipográfica). Además del *punctum* normal, están el *punctum inclinatum*, oblicuo, producido con el prefijo `\inclinatum`. El *punctum* normal se puede modificar con `\cavum`, que produce una nota vacía, y `\linea`, que traza líneas verticales a ambos lados de la nota.
- La *virga* tiene una plica descendente en el lado derecho. Se produce mediante el modificador `\virga`.

## Ligaduras

A diferencia de casi todos los otros sistemas de notación de neumas, el aspecto tipográfico de las ligaduras no viene dictado directamente por las instrucciones de la entrada, sino que sigue unas convenciones que dependen del significado musical. Por ejemplo, una ligadura de tres notas con la forma musical bajo-alto-bajo, como `\[ a \pes b \flexa g ]`, produce un Torculus que consiste en tres cabezas de Punctum, mientras que la forma alto-bajo-alto, como `\[ a \flexa g \pes b ]`, produce un Porrectus con una forma de flexa curvada y una sola cabeza de Punctum. No existe ninguna instrucción para tipografiar explícitamente la forma de flexa

curvada; la decisión de cuándo tipografiar una forma de flexa curvada está basada en la entrada musical. La idea de este enfoque es separar los aspectos musicales de la entrada, del estilo de notación de la salida. De esta forma, la misma entrada se puede reutilizar para tipografiar la misma música en un estilo diferente de notación de canto gregoriano.

### Neumas licuescentes

Otra categoría fundamental de notas en el canto gregoriano es la de los llamados neumas licuescentes. Se utilizan bajo ciertas circunstancias al final de una sílaba que acaba en una letra ‘licuescente’, es decir, consonantes sonoras que pueden tener una altura tonal, (nasales, l, r, v, j, y sus diptongos equivalentes). Así, los neumas licuescentes nunca se utilizan aisladamente (aunque se pueden producir algunos de ellos), y siempre están al final de una ligadura.

Los neumas licuescentes se representan gráficamente de dos formas distintas más o menos intercambiables: con una nota pequeña o ‘girando’ la nota principal hacia arriba o hacia abajo. La primera forma se produce haciendo un **pes** o una **flexa** normales y modificando la forma de la segunda nota: `\[ a \pes \deminutum b \]`, mientras que la segunda se hace modificando la forma de un neuma de una nota con `\auctum` y uno de los generadores de dirección `\descendens` o `\ascendens`, p.ej. `\[ \auctum \descendens a \]`.

### Símbolos especiales

Hay una tercera categoría de símbolos que se hace a partir de un pequeño número de símbolos que tienen un significado especial (que, por cierto, casi siempre se conocen sólo vagamente): el *quilisma*, el *oriscus* y el *strophicus*. Todos ellos se producen anteponiendo al nombre de la nota el modificador correspondiente, `\quilisma`, `\oriscus` o `\strophica`.

Dentro de los delimitadores de ligadura `\[` y `\]`, se pueden acumular prácticamente cualquier cantidad de cabezas de nota para formar una sola ligadura, y los prefijos de cabeza como `\pes`, `\flexa`, `\virga`, `\inclinatum`, etc. se pueden mezclar libremente. El uso del conjunto de reglas que subyace a la construcción de ligaduras en la tabla anterior está consecuentemente extrapolada. De esta manera se pueden crear un número infinito de ligaduras distintas.

Observe que el uso de esos símbolos en la propia música sigue ciertas reglas que LilyPond no comprueba. P.ej., el *quilisma* siempre es la nota intermedia de una ligadura ascendente, y suele caer sobre un intervalo de semitono, pero es perfectamente posible, si bien incorrecto, hacer un quilisma de una nota.

Además de los símbolos de nota, el archivo `gregorian.ly` define también las instrucciones `\versus`, `\responsum`, `\ij`, `\iij`, `\IJ` y `\IIJ`, que producen los caracteres correspondientes, p.ej. para utilizarlos en la letra, como marcas de sección, etc. Estas instrucciones utilizan caracteres de Unicode especiales y sólo funcionan si se usa una fuente tipográfica que los contemple.

La tabla siguiente muestra un conjunto limitado, pero representativo, de ligaduras gregorianas, junto a los fragmentos de código que las producen. La tabla está basada en la tabla de neumas extendidos del segundo volumen del Antiphonale Romanum (*Liber Hymnarius*), publicado en 1983 por los monjes de Solesmes. La primera columna da el nombre de la ligadura, con la forma principal en tipo negrita y las formas licuescentes en cursiva. La tercera forma muestra el fragmento de código que produce dicha ligadura, utilizando Sol, La y Si como alturas de ejemplo.

### Neumas de una nota

Formas **Básica** y *Licuescente*

Salida

Código de LilyPond

**Punctum**

\[ b \]



\[ \cavum b \]



\[ \linea b \]

*Punctum Auctum Ascendens*

\[ \auctum \ascendens b \]

*Punctum Auctum Descendens*

\[ \auctum \descendens b \]

**Punctum inclinatum**

\[ \inclinatum b \]

*Punctum Inclinatorum Auctum*

\[ \inclinatum \auctum b \]

*Punctum Inclinatorum Parvum*

\[ \inclinatum \deminutum b \]

**Virga****Ligaduras de dos notas****Clivis vel Flexa**

\[ b \flexa g \]



*Clivis Aucta Descendens*\[ b \flexa \auctum \descendens  
g \]*Clivis Aucta Ascendens*\[ b \flexa \auctum \ascendens  
g \]*Cephalicus*

\[ b \flexa \deminutum g \]

**Podatus o Pes**

\[ g \pes b \]

*Pes Auctus Descendens*\[ g \pes \auctum \descendens b  
\]*Pes Auctus Ascendens*\[ g \pes \auctum \ascendens b  
\]*Epiphonus*

\[ g \pes \deminutum b \]

*Pes Initio Debilis*

\[ \deminutum g \pes b \]

*Pes Auctus Descendens Initio Debilis*\[ \deminutum g \pes \auctum  
\descendens b \]**Ligaduras de varias notas****Torculus**

\[ a \pes b \flexa g \]



*Torculus Auctus Descendens*

$$\backslash[ a \backslash\text{pes } b \backslash\text{flexa } \backslash\text{auctum} \\ \backslash\text{descendens } g \backslash]$$
*Torculus Deminutus*

$$\backslash[ a \backslash\text{pes } b \backslash\text{flexa } \backslash\text{deminutum } g \\ \backslash]$$
*Torculus Initio Debilis*

$$\backslash[ \backslash\text{deminutum } a \backslash\text{pes } b \backslash\text{flexa } g \\ \backslash]$$
*Torculus Auctus Descendens Initio Debilis*

$$\backslash[ \backslash\text{deminutum } a \backslash\text{pes } b \backslash\text{flexa} \\ \backslash\text{auctum } \backslash\text{descendens } g \backslash]$$
*Torculus Deminutus Initio Debilis*

$$\backslash[ \backslash\text{deminutum } a \backslash\text{pes } b \backslash\text{flexa} \\ \backslash\text{deminutum } g \backslash]$$
**Porrectus**

$$\backslash[ a \backslash\text{flexa } g \backslash\text{pes } b \backslash]$$
*Porrectus Auctus Descendens*











$$\backslash[ a \backslash\text{flexa } g \backslash\text{pes } \backslash\text{auctum} \\ \backslash\text{descendens } b \backslash]$$
*Porrectus Deminutus*






$$\backslash[ a \backslash\text{flexa } g \backslash\text{pes } \backslash\text{deminutum } b \\ \backslash]$$
**Climacus**

$$\backslash[ \backslash\text{virga } b \backslash\text{inclinatum } a \\ \backslash\text{inclinatum } g \backslash]$$
*Climacus Auctus*

$$\backslash[ \backslash\text{virga } b \backslash\text{inclinatum } a \\ \backslash\text{inclinatum } \backslash\text{auctum } g \backslash]$$
*Climacus Deminutus*

$$\backslash[ \backslash\text{virga } b \backslash\text{inclinatum } a \\ \backslash\text{inclinatum } \backslash\text{deminutum } g \backslash]$$

<b>Scandicus</b>		<code>\[ g \pes a \virga b \]</code>
<i>Scandicus Auctus Descendens</i>		<code>\[ g \pes a \pes \auctum \descendens b \]</code>
<i>Scandicus Deminutus</i>		<code>\[ g \pes a \pes \deminutum b \]</code>
<b>Símbolos especiales</b>		
<b>Quilisma</b>		<code>\[ g \pes \quilisma a \pes b \]</code>
<i>Quilisma Pes Auctus Descendens</i>		<code>\[ \quilisma g \pes \auctum \descendens b \]</code>
<b>Oriscus</b>		<code>\[ \oriscus b \]</code>
<i>Pes Quassus</i>		<code>\[ \oriscus g \pes \virga b \]</code>
<i>Pes Quassus Auctus Descendens</i>		<code>\[ \oriscus g \pes \auctum \descendens b \]</code>
<b>Salicus</b>		<code>\[ g \oriscus a \pes \virga b \]</code>
<i>Salicus Auctus Descendens</i>		<code>\[ g \oriscus a \pes \auctum \descendens b \]</code>

<b>(Apo)stropha</b>	$\backslash[ \backslashstropha b \backslash]$
	
<i>Stropha Aucta</i>	$\backslash[ \backslashstropha \backslashauctum b \backslash]$
	
<b>Bistropha</b>	$\backslash[ \backslashstropha b \backslashstropha b \backslash]$
	
<b>Tristropha</b>	$\backslash[ \backslashstropha b \backslashstropha b \backslashstropha b \backslash]$
	
<i>Trigonus</i>	$\backslash[ \backslashstropha b \backslashstropha b \backslashstropha a \backslash]$
	

## Instrucciones predefinidas

Están contemplados los siguientes prefijos de cabeza: `\virga`, `\stropha`, `\inclinatum`, `\auctum`, `\descendens`, `\ascendens`, `\oriscus`, `\quilisma`, `\deminutum`, `\cavum`, `\linea`.

Se pueden acumular los prefijos de cabeza, aunque con ciertas restricciones. Por ejemplo, se pueden aplicar `\descendens` o `\ascendens` a una cabeza, pero no los dos a la misma cabeza.

Se pueden ligar dos cabezas adyacentes con las instrucciones infijas `\pes` y `\flexa` para una línea melódica ascendente y descendente, respectivamente.

Utilice la función musical unaria `\augmentum` para añadir puntos de augmentum.

## Véase también

Glosario musical: Sección “ligature” en *Glosario Musical*.

Referencia de la notación: [Ligaduras de neumas cuadrados gregorianos], página 480, [Ligaduras mensurales blancas], página 474, [Ligaduras], página 466.

## Advertencias y problemas conocidos

Cuando aparece un punto de `\augmentum` al final de la última pauta dentro de una ligadura, a veces se posiciona defectuosamente en el sentido vertical. Para rodear el problema, escriba una nota adicional de salto (p.ej. `s8`) como última nota de la pauta.

`\augmentum` debería estar implementado como un prefijo de cabeza en vez de una función musical unaria, de forma que `\augmentum` se pudiera entremezclar con los prefijos de cabeza en un orden arbitrario.

### 2.9.5 Tipografiado del canto kievano en notación cuadrada

#### Contextos del canto kievano

De la misma manera que con las notaciones mensural y gregoriana, pueden usarse los contextos predefinidos `KievanVoice` y `KievanStaff` para tipografiar una pieza en notación cuadrada.

Estos contextos inician todas las propiedades de contexto necesarias a sus valores adecuados, de manera que pueda ponerse manos a la obra inmediatamente con la introducción del propio canto:

```
% Font settings for Cyrillic
\paper {
  #(define fonts
    (set-global-fonts
      #:roman "Linux Libertine O,serif"
    ))
}

\score {
  <<
    \new KievanVoice = "melody" \relative c' {
      \cadenzaOn
      c4 c c c c2 b\longa
      \bar "k"
    }
    \new Lyrics \lyricsto "melody" {
      Го -- спо -- ди по -- ми -- луй.
    }
  >>
}
```



## Véase también

Glosario musical: Sección “kievan notation” en *Glosario Musical*.

## Advertencias y problemas conocidos

LilyPond contempla la notación del canto kievano en el estilo sinodal, tal y como se usa en los cantorales impresos por el Sacro Sínodo Ruso en la década de 1910 y recientemente reimpresso por la Casa de Publicaciones del Patriarcado de Moscú. LilyPond no contempla las formas anteriores (menos corrientes) de notación kievana que se usaban en Galitzia para la notación del canto llano ruteno.

## Claves del canto kievano

En la notación kievana solamente se utiliza una clave (la clave Tse-fa-ut). Se utiliza para indicar la posición del Do:

```
\clef "kievan-do"
\kievanOn
c'
```



## Véase también

Glosario musical: Sección “kievan notation” en *Glosario Musical*, Sección “clef” en *Glosario Musical*.

Referencia de la notación: [Clave], página 18.

## Notas del canto kievano

Para la notación cuadrada del canto kievano, debe escogerse el estilo apropiado para la forma de la cabeza de las notas y se deben suprimir los corchetes y las plicas. Esto se consigue mediante una llamada a la función `\kievanOn`, que establece las propiedades adecuadas para la cabeza, plica y corchete de las notas. Una vez que las notas de estilo kievano ya no se necesiten, pueden revertirse estas propiedades llamando a la función `\kievanOff`.

La nota final del canto kievano, que suele ir al final de una pieza musical, puede seleccionarse estableciendo la duración al valor `\longa`. La marca kievana de recitativo, utilizada para indicar la entonación de varias sílabas sobre una sola nota, se puede seleccionar estableciendo la duración al valor `\breve`. El siguiente ejemplo muestra las diversas formas de la cabeza de las notas del canto kievano:

```
\autoBeamOff
\cadenzaOn
\kievanOn
b'1 b'2 b'4 b'8 b'\breve b'\longa
\kievanOff
b'2
```



## Véase también

Glosario musical: Sección “kievan notation” en *Glosario Musical*, Sección “note head” en *Glosario Musical*.

Referencia de la notación: Sección A.9 [Estilos de cabezas de nota], página 734.

## Advertencias y problemas conocidos

LilyPond determina de forma automática si debe trazarse la forma de una nota con la plica hacia arriba o hacia abajo. Sin embargo, cuando se está tipografiando un canto en notación cuadrada, es preceptivo que las plicas estén orientadas en la misma dirección dentro de un melisma. Esto puede hacerse manualmente fijando la propiedad `direction` del objeto `Stem`.

## Alteraciones accidentales del canto kievano

El estilo `kievan` para las alteraciones accidentales se selecciona con la propiedad `alteration-glyph-name-alist` del grob `Accidental`. El estilo `kievan` provee unos signos de sostenido y de bemol diferentes del estilo predeterminado. No existe el becuadro en la notación kievana. El símbolo del sostenido no se utiliza en la música sinodal pero puede aparecer en manuscritos más antiguos. Se ha incluido principalmente a efectos de compatibilidad.

```
\clef "kievan-do"
\override Accidental.alteration-glyph-name-alist =
  #alteration-kievan-glyph-name-alist
bes' dis'
```



## Véase también

Glosario musical: Sección “kievan notation” en *Glosario Musical*, Sección “accidental” en *Glosario Musical*.

Referencia de la notación: [Alteraciones accidentales], página 6, [Alteraciones accidentales automáticas], página 30, Sección A.8 [La tipografía Emmentaler], página 713.

## Líneas divisorias del canto kievano

Normalmente se coloca una figura decorativa al final de una pieza de notación kievana, que puede denominarse como «doble barra final del canto kievano». Se puede invocar como `\bar "k"`.

```
\kievanOn
\clef "kievan-do"
c' \bar "k"
```



## Véase también

[Barras de compás], página 104, Sección A.8 [La tipografía Emmentaler], página 713.

## Melismas del canto kievano

Las notas que están dentro de un melisma del canto kievano se suelen colocar a poca distancia entre sí y cada melisma está separado de los demás por un espacio vacío. Esto se hace para permitir al cantor la identificación rápida de las estructuras melódicas del canto Znamenny. En LilyPond, los melismas se tratan como ligaduras y el espaciado está implementado por el grabador `Kievan_ligature_engraver`.

Cuando se usan los contextos `KievanVoice` y `KievanStaff`, se habilita el grabador `Kievan_ligature_engraver` de forma predeterminada. En otros contextos, se puede invocar sustituyendo el grabador `Ligature_bracket_engraver` por `Kievan_ligature_engraver` en el bloque layout:

```
\layout {
  \context {
    \Voice
    \remove "Ligature_bracket_engraver"
    \consists "Kievan_ligature_engraver"
  }
}
```

El espaciado entre las notas que están dentro de una misma ligadura kievana puede controlarse estableciendo la propiedad `padding` de `KievanLigature`.

El ejemplo siguiente muestra el uso de las ligaduras del canto kievano:

```
% Font settings for Cyrillic
\paper {
  #(define fonts
    (set-global-fonts
      #:roman "Linux Libertine O,serif"
    ))
}

\score {
```

```

<<
  \new KievanVoice = "melody" \relative c' {
    \cadenzaOn
    e2 \[ e4( d4 ) \] \[ c4( d e d ) \] e1 \bar "k"
  }
  \new Lyrics \lyricsto "melody" {
    Га -- вpi -- и -- лу
  }
  >>
}

```



## Véase también

Glosario musical: Sección “ligature” en *Glosario Musical*.

Referencia de la notación: [Ligaduras mensurales blancas], página 474, [Ligaduras de neumas cuadrados gregorianos], página 480, [Ligaduras], página 466.

## Advertencias y problemas conocidos

El espaciado horizontal de las ligaduras es mediocre.

### 2.9.6 Trabajar con música antigua - escenarios y soluciones

El trabajo con música antigua implica con frecuencia tareas adicionales que difieren considerablemente de la notación moderna para la que se diseñó LilyPond. En el resto de esta sección se perfilan unos cuantos escenarios típicos, con sugerencias de soluciones. Entre ellos están los siguientes:

- cómo hacer incipits (o sea, material a modo de prefacio que indica cómo era el original) a las transcripciones modernas de música mensural;
- cómo conseguir el diseño *Mensurstriche* que se usa con frecuencia para las transcripciones modernas de música polifónica;
- cómo transcribir canto gregoriano en notación moderna;
- cómo generar notación antigua y moderna a partir de la misma fuente.

## Incipits

Al transcribir música mensural a la notación moderna, es costumbre poner una indicación de la forma en que aparecían en la versión original los silencios y la nota o notas iniciales (incluso con las claves originales). Esto recibe el nombre de *incipit*. La instrucción `\incipit` usa el valor de `indent` del pentagrama principal para fijar el espacio horizontal que ocupa el incipit, y `incipit-width` para establecer la anchura del pentagrama del propio incipit.

```

\score {
  \new Staff <<
    \new Voice = Tenor {
      \set Staff.instrumentName = "Tenor"
      \override Staff.InstrumentName.self-alignment-X = #RIGHT
      \incipit { \clef "mensural-c4" \key f \major r\breve r1 c'1 }
      \clef "treble_8"
      \key f \major
    }
  }
}

```

```

      R1 r2 c'2 |
      a4. c'8
    }
    \new Lyrics \lyricsto Tenor { Cyn -- thia your }
  >>
  \layout
  {
    indent = 5\cm
    incipit-width = 3\cm
  }
}

```



## Advertencias y problemas conocidos

Observe que se debe configurar el contenido de `instrumentName` dentro de la música, para que aparezca el incipit. Si no quiere que aparezca ningún nombre para el instrumento, utilice `\set Staff.instrumentName = ""`.

## Disposición tipo «Mensurstriche»

*Mensurstriche* ('líneas de mensuración') es el término aceptado para las barras de compás que se trazan entre los pentagramas de un sistema pero no dentro de los propios pentagramas. Es una forma común de preservar el aspecto visual de las duraciones del original, es decir, sin tener que dividir las notas sincopadas en las barras de compás, y al mismo tiempo ofrecer la ayuda orientativa que proporcionan las barras de compás.

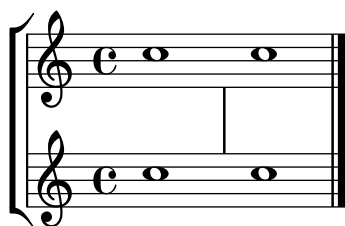
La disposición «mensurstriche» en que las líneas divisorias no están dibujadas sobre los pentagramas, sino entre ellos, se puede conseguir con un `StaffGroup` en vez de un `ChoirStaff`. La línea divisoria sobre los pentagramas se borra usando `\hide`.

```

global = {
  \hide Staff.BarLine
  s1 s
  % the final bar line is not interrupted
  \undo \hide Staff.BarLine
  \bar "|."
}

\new StaffGroup \relative c'' {
  <<
    \new Staff { << \global { c1 c } >> }
    \new Staff { << \global { c c } >> }
  >>
}

```





## Transcripción de canto gregoriano

El canto gregoriano se puede transcribir a notación moderna con ciertos trucos sencillos.

**Plicas.** Todas las plicas se pueden eliminar de una vez retirando el grabador `Stem_engraver` del contexto de voz:

```
\layout {
  ...
  \context {
    \Voice
    \remove "Stem_engraver"
  }
}
```

**Compás.** Para el canto no medido, existen diversas alternativas.

Se puede eliminar el grabador `Time_signature_engraver` del contexto de `Staff` sin ningún efecto secundario negativo. La alternativa, que es hacer transparente la indicación de compás, deja un espacio en blanco en la partitura, pues la indicación invisible sigue ocupando un espacio.

En muchos casos da un buen resultado `\set Score.timing = ##f`. Otra alternativa es utilizar `\cadenzaOn` y `\cadenzaOff`.

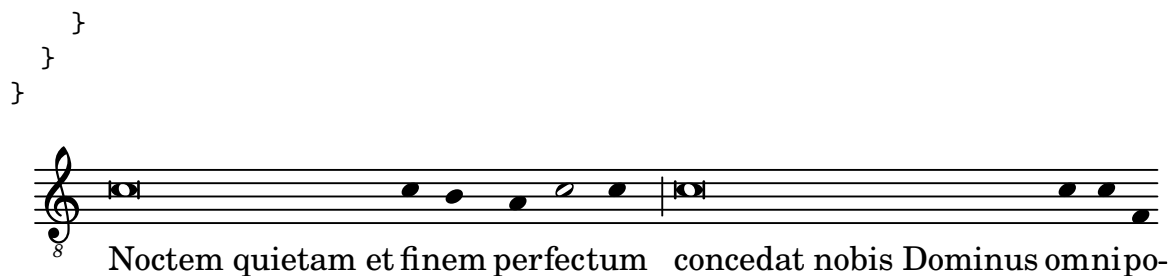
Para quitar las barras de compás, el enfoque radical consiste en retirar (mediante la instrucción `\remove`) el grabador `Bar_engraver` del contexto de pentagrama `Staff`. Una vez más, en vez de ello podemos usar `\hide BarLine` si se necesita ocasionalmente una barra de compás.

Un tipo de transcripción muy común es el canto de recitativo, en el que la nota repetida se indica con una breve. Se puede tratar el texto del tono de recitativo de dos maneras. Podemos ponerlo como una sola sílaba alineada por la izquierda:

```
\include "gregorian.ly"
chant = \relative {
  \clef "G_8"
  c'\breve c4 b4 a c2 c4 \divisioMaior
  c\breve c4 c f, f \finalis
}

verba = \lyricmode {
  \once \override LyricText.self-alignment-X = #-1
  "Noctem quietam et" fi -- nem per -- fec -- tum
  \once \override LyricText.self-alignment-X = #-1
  "concedat nobis Dominus" om -- ni -- po -- tens.
}

\score {
  \new Staff <<
  \new Voice = "melody" \chant
  \new Lyrics = "one" \lyricsto melody \verba
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \remove "Bar_engraver"
    }
    \context {
      \Voice
      \remove "Stem_engraver"
    }
  }
}
```



tens.

Esto funciona bien en la medida en que el texto no comprenda un salto de línea. Si es el caso, una alternativa sería añadir notas ocultas a la partitura, en combinación con cambios en la visibilidad de las plicas:

En ciertos estilos de transcripción, se usan plicas de forma ocasional, por ejemplo para indicar la transición a partir de un recitativo monotónico a un gesto melódico fijo. En estos casos podemos usar bien `\hide Stem` o bien `\override Stem.length = #0` en lugar de eliminar el grabador `Stem_engraver` con `\remove`, y restaurar la plica cuando sea necesario, con el correspondiente `\undo \hide Stem`.

```
\include "gregorian.ly"
chant = \relative {
  \clef "G_8"
  \set Score.timing = ##f
  \hide Stem
  c'\breve \hide NoteHead c c c c c
  \undo \hide NoteHead
  \undo \hide Stem \stemUp c4 b4 a
  \hide Stem c2 c4 \divisioMaior
  c\breve \hide NoteHead c c c c c c c
  \undo \hide NoteHead c4 c f, f \finalis
}

verba = \lyricmode {
  No -- ctem qui -- e -- tam et fi -- nem per -- fec -- tum
  con -- ce -- dat no -- bis Do -- mi -- nus om -- ni -- po -- tens.
}

\score {
  \new Staff <<
    \new Voice = "melody" \chant
    \new Lyrics \lyricsto "melody" \verba
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \hide BarLine
    }
  }
}
```

}



Otra situación común es la transcripción de cantos neumáticos o melismáticos, es decir, cantos con un número variable de notas en cada sílaba. En este caso, podemos establecer los grupos de sílabas claramente separados, normalmente también las subdivisiones de un melisma mayor. Una forma de conseguirlo es usar un `\time` fijo, p.ej. `1/4`, y dejar que cada sílaba o grupo de notas llene uno de esos compases, con la ayuda de grupos de valoración especial o duraciones más breves. Si las líneas divisorias y todas las demás indicaciones rítmicas se hacen transparentes, y el espacio que rodea las líneas divisorias se aumenta, ello proporciona una aceptable representación del original, en notación moderna.

Para evitar que las sílabas de distinta anchura (como “-ri” y “-rum”) dispongan los grupos de notas de las sílabas de forma irregular, se puede ajustar la propiedad `'X-extent` del objeto `LyricText` a un valor fijo. Otra forma más engorrosa sería añadir las sílabas como elementos de `\markup`. Si se necesitan más ajustes posteriormente, esto se puede hacer fácilmente con ‘notas’ s.

```
spiritus = \relative {
  \time 1/4
  \override Lyrics.LyricText.X-extent = #'(0 . 3)
  d'4 \tuplet 3/2 { f8 a g } g a a4 g f8 e
  d4 f8 g g8 d f g a g f4 g8 a a4 s
  \tuplet 3/2 { g8 f d } e f g a g4
}

spirLyr = \lyricmode {
  Spi -- ri -- _ _ tus _ Do -- mi -- ni _ re -- ple -- _ vit _
  or -- _ bem _ ter -- ra -- _ rum, al -- _ _ le -- _ lu
  -- _ ia.
}

\score {
  \new Staff <<
    \new Voice = "chant" \spiritus
    \new Lyrics = "one" \lyricsto "chant" \spirLyr
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \override BarLine.X-extent = #'(-1 . 1)
      \hide Stem
      \hide Beam
      \hide BarLine
      \hide TupletNumber
    }
  }
}
```

}



## Antigua y moderna desde la misma fuente

### *Using tags to produce mensural and modern music from the same source*

By using tags, it's possible to use the same music to produce both mensural and modern music. In this snippet, a function `menrest` is introduced, allowing mensural rests to be pitched as in the original, but with modern rests in the standard staff position. Tags are used to produce different types of bar line at the end of the music, but tags can also be used where other differences are needed: for example using “whole measure rests” (`R1`, `R\breve` etc.) in modern music, but normal rests (`r1`, `r\breve`, etc.) in the mensural version. Note that converting mensural music to its modern equivalent is usually referred to as **transcription**.

```
menrest = #(define-music-function (note)
  (ly:music?)
  #{
    \tag #'mens $(make-music 'RestEvent note)
    \tag #'mod $(make-music 'RestEvent note 'pitch '())
  })

MensStyle = {
  \autoBeamOff
  \override NoteHead.style = #'petrucci
  \override Score.BarNumber.transparent = ##t
  \override Stem.neutral-direction = #up
}

finalis = {
  \once \override BreathingSign.stencil = #ly:breathing-sign::finalis
  \once \override BreathingSign.Y-offset = #0
  \once \override BreathingSign.minimum-X-extent = #'(-1.0 . 0.0)
  \once \override BreathingSign.minimum-Y-extent = #'(-2.5 . 2.5)

  \breathe
}

Music = \relative c'' {
  \set Score.tempoHideNote = ##t
  \key f \major
  \time 4/4
  g1 d'2 \menrest bes4 bes2 a2 r4 g4 fis2.
```

```

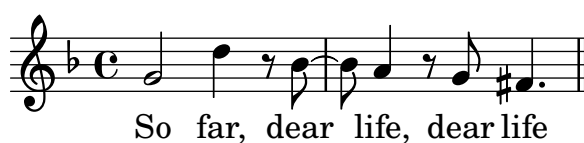
\tag #'mens { \finalis }
\tag #'mod { \bar "||" }
}

MenLyr = \lyricmode { So farre, deere life, deare life }
ModLyr = \lyricmode { So far, dear life, dear life }

\score {
  \keepWithTag #'mens {
    <<
      \new MensuralStaff
      {
        \new MensuralVoice = Cantus
        \clef "mensural-c1" \MensStyle \Music
      }
      \new Lyrics \lyricsto Cantus \MenLyr
    >>
  }
}

\score {
  \keepWithTag #'mod {
    \new ChoirStaff <<
      \new Staff
      {
        \new Voice = Sop \with {
          \remove "Note_heads_engraver"
          \consists "Completion_heads_engraver"
          \remove "Rest_engraver"
          \consists "Completion_rest_engraver" }
        {
          \shiftDurations #1 #0 { \autoBeamOff \Music }
        }
      }
      \new Lyrics \lyricsto Sop \ModLyr
    >>
  }
}

```



## Marcado editorial

Continuará...

## 2.10 Músicas del mundo

El propósito de esta sección es remarcar los problemas de notación que son de relevancia a las tradiciones musicales distintas a la occidental.

### 2.10.1 Notación común para músicas no occidentales

Esta sección estudia la forma de introducir e imprimir partituras musicales que no pertenecen a la tradición clásica occidental, también conocida como *Período de la práctica común*.

#### Extensión de la notación y los sistemas de afinación

La notación clásica estándar (también conocida como la notación del *Período de la práctica común*) se usa normalmente en todo tipo de música, sin limitación a la música occidental ‘clásica’. Esta notación se estudia en Sección 1.1.1 [Escritura de notas], página 1, y los distintos nombres de las notas que se pueden utilizar están explicados en [Nombres de las notas en otros idiomas], página 8.

Ciertos tipos de música no occidental, folk y tradicional con frecuencia emplean sistemas de afinación alternativos o extendidos que no encajan fácilmente dentro de la notación clásica estándar.

Muchas veces se usa la notación estándar, pero con algunas diferencias de afinación implícitas. Por ejemplo, la notación de la *música árabe* se realiza con alteraciones accidentales de semitono y de cuarto de tono, estando determinadas por el contexto las exactas variaciones en la entonación. En el caso de la *música árabe*, el archivo de inicio `arabic.ly` proporciona un conjunto adecuado de macros y definiciones que extienden la notación estándar que emplea los nombres de nota italianos. Para ver más detalles, consulte Sección 2.10.2 [Música árabe], página 497.

Otros tipos de música requieren notaciones extendidas o singulares; por ejemplo, la *música clásica turca* o música clásica otomana, por ejemplo, emplea formas melódicas conocidas como *makamlar* cuyos intervalos están basados en divisiones de  $1/9$  de tono. Se siguen usando las notas del pentagrama occidental, pero con alteraciones accidentales que están definidas de forma inequívoca en el archivo `turkish-makam.ly`. Para ver más información sobre la música clásica turca y los *makamlar*, consulte Sección 2.10.3 [Música clásica de Turquía], página 503.

También están disponibles otros archivos de inicio relacionados, `hel-arabic.ly` and `makam.ly`.

Para localizar estos archivos de inicio en su sistema, consulte Sección “Otras fuentes de información” en *Manual de Aprendizaje*.

#### Véase también

Glosario musical: Sección “Common Practice Period” en *Glosario Musical*, Sección “makamlar” en *Glosario Musical*.

Manual de aprendizaje: Sección “Otras fuentes de información” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 1.1.1 [Escritura de notas], página 1, [Nombres de las notas en otros idiomas], página 8, Sección 2.10.2 [Música árabe], página 497, Sección 2.10.3 [Música clásica de Turquía], página 503.

### 2.10.2 Música árabe

Esta sección destaca asuntos relativos a la música árabe.

#### Referencias para música árabe

La música árabe ha sido principalmente una tradición oral hasta el momento. Cuando se transcribe la música, se suele hacer en la forma de un esbozo sobre el que los intérpretes deben

improvisar ampliamente. En una medida creciente, se está adoptando la notación occidental con ciertas variantes para preservar y transmitir la música árabe.

Algunos elementos de la notación musical occidental como la transcripción de acordes o las particellas independientes, no se necesitan para tipografiar las piezas árabes, más tradicionales. Sin embargo hay ciertos problemas, como la necesidad de indicar intervalos que tienen una medida intermedia entre el semitono y el tono, además de los intervalos mayores y menores de la música occidental. También existe la necesidad de agrupar e indicar los numerosos maqams (modos) que forman parte de la música árabe.

Por lo general la notación musical árabe no trata de indicar con precisión los elementos microtonales que se encuentran presentes en la práctica musical.

Son de relevancia para la música árabe ciertos asuntos que se encuentran cubiertos en otras partes del manual:

- Los nombres de las notas y sus alteraciones (entre ellos, cuartos de tono) se pueden realizar como se estudia en Sección 2.10.1 [Notación común para músicas no occidentales], página 497.
- Las armaduras adicionales también pueden realizarse como se describe en [Armadura de la tonalidad], página 23.
- Los compases complejos pueden requerir agrupaciones manuales de las notas tal y como se describe en [Barras manuales], página 100.
- Los *takasim* o improvisaciones libres desde el punto de vista rítmico se pueden escribir omitiendo las barras de compás como aparece en [Música sin compasear], página 79.

## Véase también

Referencia de la notación: Sección 2.10.1 [Notación común para músicas no occidentales], página 497, [Armadura de la tonalidad], página 23, [Barras manuales], página 100.

Fragmentos de código: Sección “World music” en *Fragmentos de código*.

## Nombres de nota árabes

Los nombres tradicionales de las notas en árabe pueden ser bastante largos y por ello no siempre son adecuados para el propósito de escribir música.

El archivo `hel-arabic.ly` permite usar los nombres de nota ingleses. He aquí una escala `rast` que usa `hel-arabic.ly`;

```
\include "hel-arabic.ly"
\relative {
\key c \rast
c' d edb f | g a bdb c | c bb a g | f d c
}
```



En lugar de ello se pueden usar los nombres de las notas en italiano (o Solfege) gracias al archivo `arabic.ly`. He aquí una escala `rast` que usa `arabic.ly`;

```
\include "arabic.ly"
\relative {
do' re misb fa | sol la sisb do | sisb la sol fa | misb re do
```

}



“Rast” es una escala heptatónica que usa intervalos en cuartos de tono y se considera la escala central y más importante del “Maqamat árabe”. Para ver la lista completa de las escalas árabes contempladas, consulte los archivos `hel-arabic.ly` o `arabic.ly` que están incluidos en la distribución de LilyPond.

El uso de notación occidental estándar para música no occidental se discute en Sección 2.10.1 [Notación común para músicas no occidentales], página 497. Véase también [Nombres de las notas en otros idiomas], página 8.

El símbolo del semi-bemol no coincide con el símbolo que se utiliza en la notación árabe. Se puede usar el símbolo `\dwn` definido en `arabic.ly` precediendo a un bemol como vía alternativa si es importante usar el símbolo árabe específico del semi-bemol. El aspecto del símbolo del semi-bemol dentro de la armadura no se puede alterar mediante este método.

```
\include "arabic.ly"
\relative {
  \set Staff.extraNatural = ##f
  dod' dob dosd \dwn dob dobsd dodsd do do
}
```



## Véase también

Referencia de la notación: [Nombres de las notas en otros idiomas], página 8, Sección 2.10.1 [Notación común para músicas no occidentales], página 497, Sección 3.3.1 [Inclusión de archivos de LilyPond], página 538.

Archivos instalados: `ly/arabic.ly` `ly/hel-arabic.ly`

Fragmentos de código: Sección “World music” en *Fragmentos de código*.

## Armaduras de tonalidad árabes

Además de las armaduras de tonalidad mayores y menores, las armaduras de tonalidad de la música árabe vienen definidas en los archivos `hel-arabic.ly` o `arabic.ly` y definen muchos grupos de maqam distintos.

En general, un maqam usa la armadura de su grupo o la de un grupo vecino, y las alteraciones que difieren se van marcando a lo largo de la música. Los maqams árabes admiten sólo modulaciones limitadas, a causa de la naturaleza de los instrumentos musicales árabes.

He aquí un ejemplo de armadura de tonalidad para una pieza del tipo “maqam muhayer”:

```
\key re \bayati
```

Aquí, *re* es la altura predeterminada del maqam muhayer, y *bayati* es el nombre del maqam básico dentro del grupo.

Al tiempo que la armadura indica el grupo, el título suele indicar el maqam específico, y así en este ejemplo el nombre del “maqam muhayer” debe aparecer también en el título.

Se pueden indicar de la misma forma otros maqams dentro del mismo grupo *bayati*, como aparece en la tabla a continuación, (p. ej. *bayati*, *hussaini*, *saba* y *ushaq*). Todos ellos son



variantes del maqam básico y más común dentro del grupo, que es *bayati*. Normalmente se diferencian del maqam básico en su tetracordo superior, o en ciertos detalles de discurso que no cambian su naturaleza fundamental como parientes.

El otro maqam del grupo (*nawa*) se relaciona con el *bayati* a través de una modulación y se indica en la tabla entre paréntesis para aquellos que son modulaciones de su maqam básico. El *Nawa*, por ejemplo, se puede indicar de la siguiente forma:

```
\key sol \bayati
```

En la música árabe, el mismo término (p.ej. *bayati*) que se utiliza para indicar un grupo de maqam, es también un maqam que normalmente es el más importante del grupo, y podemos verlo como el *maqam básico*.

A continuación presentamos una posibilidad de agrupación que asigna los maqams más comunes a armaduras de tonalidad:

grupo maqam	de	tono	final	otros maqams del grupo (final)
ajam		major	sib	jaharka (fa)
bayati		bayati	re	hussaini, muhayer, saba, ushaq, nawa (sol)
hijaz		kurd	re	shahnaz, shad arban (sol), hijazkar (do)
iraq		iraq	sisb	-
kurd		kurd	re	hijazkar kurd (do)
nahawand		minor	do	busalik (re), farah faza (sol)
nakriz		minor	do	nawa athar, hisar (re)
rast		rast	do	mahur, yakah (sol)
sikah		sikah	misb	huzam

## Fragmentos de código seleccionados

### *Armaduras de tonalidad no tradicionales*

La muy utilizada instrucción `\key` establece la propiedad `keyAlterations` dentro del contexto `Staff`. Para crear armaduras de tonalidad no estándar, ajuste esta propiedad directamente.

El formato de esta instrucción es una lista:

```
\set Staff.keyAlterations = #`(((octava . paso) . alteración) ((octava . paso) .
alteración) ...)
```

donde, para cada elemento dentro de la lista, *octava* especifica la octava (siendo cero la octava desde el Do central hasta el Si por encima), *paso* especifica la nota dentro de la octava (cero significa Do y 6 significa Si), y *alteración* es `,SHARP`, `,FLAT`, `,DOUBLE-SHARP` etc.

De forma alternativa, el uso del formato más conciso (*paso . alteración*) para cada elemento de la lista especifica que la misma alteración debe estar en todas las octavas. Para escalas microtonales en las que un “sostenido” no son 100 cents, *alteración* se refiere a la proporción de un tono entero de 200 cents.

```
\include "arabic.ly"
\relative do' {
  \set Staff.keyAlterations = #`((0 . ,SEMI-FLAT)
                                (1 . ,SEMI-FLAT)
                                (2 . ,FLAT)
                                (5 . ,FLAT)
                                (6 . ,SEMI-FLAT))

  %\set Staff.extraNatural = ##f
  re reb \down reb resd
```

```

dod dob dosd \dwn dob |
dobsb dodsd do do |
}

```



## Véase también

Glosario musical: Sección “maqam” en *Glosario Musical*, Sección “bayati” en *Glosario Musical*, Sección “rast” en *Glosario Musical*, Sección “sikah” en *Glosario Musical*, Sección “iraq” en *Glosario Musical*, Sección “kurd” en *Glosario Musical*.

Manual de aprendizaje: Sección “Alturas y armaduras” en *Manual de Aprendizaje*.

Referencia de la notación: [Armadura de la tonalidad], página 23.

Archivos instalados: `ly/arabic.ly` `ly/hel-arabic.ly`

Fragmentos de código: Sección “World music” en *Fragmentos de código*, Sección “Pitches” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “KeySignature” en *Referencia de Funcionamiento Interno*.

## Indicaciones de compás árabes

Ciertas formas clásicas de música árabe y turca como el *Semai* utilizan compases poco comunes como el 10/8. Ello puede derivar en una agrupación automática de las notas muy distinta a la de las partituras impresas existentes, donde las notas pueden no estar agrupadas sobre el pulso, sino de una forma difícil de emular ajustando el barrado automático. La alternativa sería desactivar el barrado automático y realizar un barrado manual de las notas. Incluso si no es necesaria una correspondencia con música escrita ya existente, puede seguir siendo deseable ajustar el comportamiento del barrado automático y/o utilizar indicaciones de compás compuestas.

## Fragmentos de código seleccionados

### *Improvisación de música árabe*

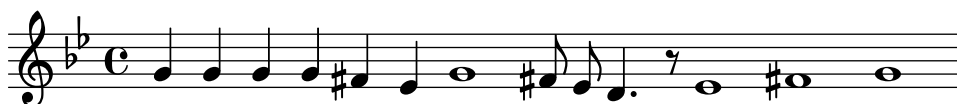
Para las improvisaciones o *tagasim* que son libres durante unos momentos, se puede omitir la indicación de compás y se puede usar `\cadenzaOn`. Podría ser necesario ajustar el estilo de alteraciones accidentales, porque la ausencia de líneas divisorias hará que la alteración aparezca una sola vez. He aquí un ejemplo de cómo podría ser el comienzo de una improvisación *hijaz*:

```

\include "arabic.ly"

\relative sol' {
  \key re \kurd
  \accidentalStyle forget
  \cadenzaOn
  sol4 sol sol sol fad mib sol1 fad8 mib re4. r8 mib1 fad sol
}

```



## Véase también

Glosario musical: Sección “semai” en *Glosario Musical*, Sección “taqasim” en *Glosario Musical*.

Referencia de la notación: [Barras manuales], página 100, [Barras automáticas], página 88, [Música sin compasear], página 79, [Alteraciones accidentales automáticas], página 30, [Establecer el comportamiento de las barras automáticas], página 91, [Indicación de compás], página 70.

Archivos instalados: `ly/arabic.ly` `ly/hel-arabic.ly`

Fragmentos de código: Sección “World music” en *Fragmentos de código*.

## Ejemplo de música árabe

A continuación aparece una plantilla que emplea también el comienzo de un *Semai* turco muy común en la educación musical árabe con el objeto de ilustrar algunas de las peculiaridades de la notación árabe, como los intervalos intermedios y los modos poco usuales que tratamos en esta sección.

```
\include "arabic.ly"
\score {
  \header {
    title = "Semai Muhayer"
    composer = "Jamil Bek"
  }
  \relative {
    \set Staff.extraNatural = ##f
    \set Staff.autoBeaming = ##f
    \key re \bayati
    \time 10/8

    re'4 re'8 re16 [misb re do] sisb [la sisb do] re4 r8
    re16 [misb do re] sisb [do] la [sisb sol8] la [sisb] do [re] misb
    fa4 fa16 [misb] misb8. [re16] re8 [misb] re [do] sisb
    do4 sisb8 misb16 [re do sisb] la [do sisb la] la4 r8
  }
}
```



## Véase también

Archivos instalados: `ly/arabic.ly` `ly/hel-arabic.ly`

## Véase también

Fragmentos de código: Sección “World music” en *Fragmentos de código*.

## Lecturas adicionales sobre música árabe

Existen ciertas variantes en los detalles sobre la forma en que se agrupan los maqams, a pesar del relativo acuerdo existente acerca de los criterios de agrupación de maqams en función de los tetracordos inferiores comunes, o por medio de la modulación. No es completamente consistente, incluso a veces dentro del mismo texto, la forma en que se deben especificar las armaduras para los maqams concretos. Sin embargo, es usual utilizar una armadura por ‘grupo’ de maqams, más que una armadura distinta para cada maqam concreto.

- *The music of the Arabs*, de Habib Hassan Touma [Amadeus Press, 1996], contiene un tratado de los maqams y sus formas de agrupación.

También existen algunos sitios web que explican los maqams e incluso ofrecen ejemplos de audio:

- <https://www.maqamworld.com/>
- <https://www.turath.org/>
- Varios métodos de los siguientes autores para el *Ud*, el laúd árabe, contienen ejemplos de composiciones turcas y árabes, principalmente.
  - Charbel Rouhana
  - George Farah
  - Ibrahim Ali Darwish Al-masri

### 2.10.3 Música clásica de Turquía

Esta sección destaca los problemas de relevancia para la notación de la música clásica de Turquía.

#### Referencias para música clásica de Turquía

La música clásica de Turquía se desarrolló en el imperio Otomano durante un período aproximadamente contemporáneo con la música clásica de Europa, y se ha prolongado hasta los ss.XX y XXI como una viva y característica tradición con sus propias formas composicionales, teorías y estilos de interpretación. Entre sus características más sobresalientes está el uso de intervalos microtonales basados en ‘comas’ de  $1/9$  de tono, a partir de los que se construyen las formas melódicas conocidas como *makam* (en plural *makamlar*).

Ciertos asuntos de relevancia respecto a la música clásica de Turquía se tratan en otras partes. Los nombres especiales de las notas y las alteraciones se dan en Sección 2.10.1 [Notación común para músicas no occidentales], página 497.

#### Nombres de las notas en turco

Las notas en la música clásica de Turquía tienen tradicionalmente nombres únicos, y la base de las alturas sobre divisiones de  $1/9$  de tono significan que los makamlar utilizan un conjunto de intervalos completamente diferente de las escalas y modos occidentales.

Desde el punto de vista de la notación moderna, es conveniente utilizar las notas occidentales (do, re, mi, . . . ó c, d, e, . . .) con alteraciones especiales que elevan o bajan las notas en intervalos de  $1/9$ ,  $4/9$ ,  $5/9$  y  $8/9$  de tono.

Estas alteraciones personalizadas están definidas en el archivo `makam.ly`.

Para ver una explicación más general sobre la notación musical no occidental, consulte Sección 2.10.1 [Notación común para músicas no occidentales], página 497.

#### Véase también

Glosario musical: Sección “makam” en *Glosario Musical*, Sección “makamlar” en *Glosario Musical*.

Referencia de la notación: Sección 2.10.1 [Notación común para músicas no occidentales], página 497.

## Armaduras de tonalidad de la música turca

Lilypond da soporte a más de 200 definiciones de armaduras de makam (mucho más de las que se utilizan en la música clásica turca), teniendo cada makam su propia tónica o finalis (conocida en turco como ‘karar’).

Es importante tener en cuenta la nota finalis de cada makam. He aquí un ejemplo en el que *g* es la tónica predeterminada y *rast* es el nombre del makam.

```
\key g \rast
```

Las alteraciones correctas, bemol de coma (*b1*) y sostenido de coma (*f4*), (ambos en relación a la tónica *g*), se imprimen automáticamente.

## Fragmentos de código seleccionados

### *Turkish Makam example*

This template uses the start of a well-known Turkish Saz Semai that is familiar in the repertoire in order to illustrate some of the elements of Turkish music notation.

```
% Initialize makam settings
\include "turkish-makam.ly"

\header {
  title = "Hüseyni Saz Semaisi"
  composer = "Lavtac1 Andon"
}

\relative {
  \set Staff.extraNatural = ##f
  \set Staff.autoBeaming = ##f

  \key a \huseyni
  \time 10/8

  a'4 g'16 [fb] e8. [d16] d [c d e] c [d c8] bfc |
  a16 [bfc a8] bfc c16 [d c8] d16 [e d8] e4 fb8 |
  d4 a'8 a16 [g fb e] fb8 [g] a8. [b16] a16 [g] |
  g4 g16 [fb] fb8. [e16] e [g fb e] e4 r8 |
}
```

## Hüseyni Saz Semaisi

Lavtac1 Andon



**Lecturas adicionales sobre música turca**

- *Türk Musikisi Nazariyatı ve Usulleri: Kudüm Velveleleri* de İsmail Hakkı Özkan [(Kültür serisi, 41) (Turkish) Paperback – 1986]  
contiene información acerca de la teoría de los maqams y los usul.
- *Music of the Ottoman Court* de Walter Feldman [VWB Hardback – 1996]  
contiene información sobre la historia de la música cortesana otomana.
- *Turkish Music Makam Guide* de Murat Aydemir [Pan Paperback – 2010]  
contiene información en inglés concerniente al makam turco, incluyendo dos CDs.

## 3 Entrada y salida generales

Esta sección trata de asuntos relacionados con la entrada y salida generales de LilyPond, más que con la notación específica.

### 3.1 Estructura del código de entrada

El formato principal de entrada para LilyPond son archivos de texto. Por convenio, el nombre de estos archivos termina en `.ly`.

#### 3.1.1 Estructura de una partitura

Un bloque `\score` debe contener una expresión musical única delimitada por un par de llaves:

```
\score {  
  ...  
}
```

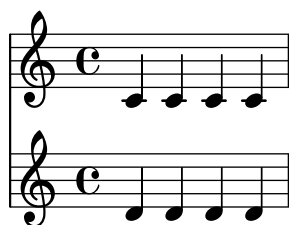
**Nota:** Sólo puede haber **una** expresión musical dentro del bloque `\score`, y **debe** estar encerrada entre llaves.

Esta expresión musical única puede tener cualquier tamaño, y puede contener otras expresiones musicales hasta cualquier grado de complejidad. Todos los siguientes ejemplos son expresiones musicales:

```
{ c'4 c' c' c' }  
{  
  { c'4 c' c' c' }  
  { d'4 d' d' d' }  
}
```



```
<<  
  \new Staff { c'4 c' c' c' }  
  \new Staff { d'4 d' d' d' }  
>>
```



```
{  
  \new GrandStaff <<  
    \new StaffGroup <<  
      \new Staff { \flute }  
      \new Staff { \oboe }  
    >>  
    \new StaffGroup <<  
      \new Staff { \violinI }  
    >>  
}
```

```

        \new Staff { \violinII }
    >>
>>
}

```

Los comentarios son una excepción a esta regla general (para ver otras excepciones, consulte Sección 3.1.5 [Estructura del archivo], página 510). Tanto los comentarios de una línea como los que están delimitados por `%{ ... %}` se pueden escribir en cualquier lugar dentro del archivo de entrada. Se pueden escribir dentro o fuera del bloque `\score`, y dentro o fuera de la expresión musical única dentro del bloque `\score`.

Recuerde que incluso dentro de un archivo que contenga solamente un bloque `\score`, éste se encuentra incluido implícitamente dentro de un bloque `\book`. Un bloque `\book` dentro de un archivo fuente produce al menos un archivo de salida, y de forma predeterminada el nombre del archivo de salida que se produce está derivado del nombre del archivo de entrada, de forma que `fandangofoforelephants.ly` da como resultado `fandangofoforelephants.pdf`.

(Para ver más detalles acerca de los bloques `\book`, consulte Sección 3.1.2 [Varias partituras en un libro], página 507, Sección 3.1.3 [Varios archivos de salida a partir de uno de entrada], página 508, y Sección 3.1.5 [Estructura del archivo], página 510.)

## Véase también

Manual de aprendizaje: Sección “Trabajar sobre los archivos de entrada” en *Manual de Aprendizaje*, Sección “Explicación de las expresiones musicales” en *Manual de Aprendizaje*, Sección “La partitura es una (única) expresión musical compuesta” en *Manual de Aprendizaje*.

### 3.1.2 Varias partituras en un libro

Un documento puede contener varias piezas de música y texto. Son ejemplos un libro de estudios o una parte orquestal con varios movimientos. Cada movimiento se introduce con un bloque `\score`:

```

\score {
  ...música...
}

```

y los textos se introducen con un bloque `\markup`:

```

\markup {
  ...texto...
}

```

Todos los movimientos y textos que aparecen en el mismo archivo `.ly` se tipografían normalmente en la forma de un solo archivo de salida.

```

\score {
  ...
}
\markup {
  ...
}
\score {
  ...
}

```

Una excepción importante se da dentro de los documentos de `lilypond-book`, en los que tenemos que escribir explícitamente un bloque `\book`, pues en caso contrario sólo aparecerá en la salida el primer `\score` o `\markup`.

El encabezamiento de cada pieza se puede poner dentro del bloque `\score`. Antes de cada movimiento, se imprime el nombre de la pieza (`piece`) extraído del encabezamiento. El título



del libro entero se puede poner dentro del bloque `\book`, pero si no está presente se inserta el encabezamiento `\header` que está al principio del archivo.

```
\header {
  title = "Ocho miniaturas"
  composer = "Igor Stravinsky"
}
\score {
  \header { piece = "Romanze" }
  ...
}
\markup {
  ...texto de la segunda estrofa...
}
\markup {
  ...texto de la tercera estrofa...
}
\score {
  \header { piece = "Menuetto" }
  ...
}
```

Se pueden agrupar distintas piezas de música en partes de un libro utilizando bloques `\bookpart`. Las partes de libro se separan por medio de saltos de página, y pueden dar comienzo con un título, como el libro propiamente dicho, mediante la especificación del bloque `\header`.

```
\bookpart {
  \header {
    title = "Título del libro"
    subtitle = "Primera parte"
  }
  \score { ... }
  ...
}
\bookpart {
  \header {
    subtitle = "Segunda parte"
  }
  \score { ... }
  ...
}
```

### 3.1.3 Varios archivos de salida a partir de uno de entrada

Si queremos varios archivos de salida a partir del mismo archivo `.ly`, podemos escribir varios bloques `\book`, donde cada uno de estos bloques `\book` dará como resultado un archivo de salida distinto. Si no especificamos ningún bloque `\book` en el archivo, LilyPond trata implícitamente todo el archivo como un solo bloque `\book`, véase Sección 3.1.5 [Estructura del archivo], página 510.

Al producir varios archivos a partir de un solo archivo fuente, LilyPond se asegura de que ninguno de los archivos de salida de ningún bloque `\book` sobrescribe el archivo de salida producido por un `\book` anterior del mismo archivo de entrada.

Lo hace añadiendo un sufijo al nombre de la salida para cada libro `\book` que utiliza el nombre del archivo de salida predeterminado, derivado del archivo fuente de entrada.

El comportamiento predeterminado es añadir un sufijo con el número de versión para cada nombre que pueda coincidir; así:

```
\book {
  \score { ... }
  \paper { ... }
}
\book {
  \score { ... }
  \paper { ... }
}
\book {
  \score { ... }
  \paper { ... }
}
```

en el archivo fuente `eightminiatures.ly`, produce

- `eightminiatures.pdf`,
- `eightminiatures-1.pdf` y
- `eightminiatures-2.pdf`.

### 3.1.4 Nombres de los archivos de salida

LilyPond le ofrece unos mecanismos que le permiten controlar los nombres de archivo que se utilizan por parte de los diversos motores finales al producir los archivos de salida.

En la sección anterior vimos cómo LilyPond evita las coincidencias de nombres al producir varias salidas a partir de un único archivo fuente de entrada. También podemos especificar nuestros propios sufijos para cada bloque `\book`, de manera que podemos producir, por ejemplo, archivos con los nombres `eightminiatures-Romanze.pdf`, `eightminiatures-Menuetto.pdf` y `eightminiatures-Nocturne.pdf` añadiendo una declaración `\bookOutputSuffix` dentro de cada bloque `\book`.

```
\book {
  \bookOutputSuffix "Romanze"
  \score { ... }
  \paper { ... }
}
\book {
  \bookOutputSuffix "Menuetto"
  \score { ... }
  \paper { ... }
}
\book {
  \bookOutputSuffix "Nocturne"
  \score { ... }
  \paper { ... }
}
```

También podemos especificar un nombre de archivo de salida diferente para cada bloque `book`, mediante la utilización de declaraciones `\bookOutputName`:

```
\book {
  \bookOutputName "Romanze"
  \score { ... }
  \paper { ... }
}
```

```

\book {
  \bookOutputName "Menuetto"
  \score { ... }
  \paper { ... }
}
\book {
  \bookOutputName "Nocturne"
  \score { ... }
  \paper { ... }
}

```

El código anterior produce los siguientes archivos de salida:

- Romanze.pdf,
- Menuetto.pdf y
- Nocturne.pdf.

### 3.1.5 Estructura del archivo

Un archivo `.ly` puede contener cualquier número de expresiones del nivel superior, donde una expresión del nivel superior es una de las siguientes posibilidades:

- Una definición de salida, como `\paper`, `\midi` y `\layout`. Tal definición en el nivel más alto cambia los ajustes predeterminados para todo el libro. Si se introduce más de una de estas definiciones del mismo tipo en el nivel más alto, las definiciones se combinan, pero en situaciones de conflicto las definiciones más tardías tienen preferencia. Para ver los detalles sobre cómo afecta al bloque `\layout`, consulte Sección 4.2.1 [El bloque `\layout`], página 579.
- Una expresión directa de Scheme, como `#{set-default-paper-size "a7" 'landscape}` o `#{ly:set-option 'point-and-click #f}`.
- Un bloque `\header`. Esto establece el bloque de encabezamiento global (es decir, al principio del archivo). Es el bloque que contiene los ajustes predeterminados de los campos de encabezamientos como el compositor, título, etc. para todos los libros que están dentro del archivo (véase [Explicación de los títulos], página 512).
- Un bloque `\score`. Esta partitura se reúne con otras partituras del nivel superior, y se combina con ellas formando un `\book` (libro) único. Este comportamiento se puede cambiar estableciendo el valor de la variable `toplevel-score-handler` en el nivel superior (el *handler* predeterminado se define en el archivo `../scm/lily-library.scm` y su valor se establece en el archivo `../ly/declarations-init.ly`).
- Un bloque `\book` combina lógicamente varios movimientos (es decir, varios bloques `\score`) en un solo documento. Si hay varias partituras `\score`, se crea un solo archivo de salida para cada bloque `\book`, en el que se concatenan todos los movimientos correspondientes. La única razón para especificar explícitamente bloques `\book` en un archivo de entrada `.ly` es si queremos crear varios archivos de salida a partir de un único archivo de entrada. Una excepción se da dentro de los documentos de lilypond-book, en los que tenemos que escribir explícitamente un bloque `\book` si queremos tener más de una única partitura `\score` o marcado `\markup` en el mismo ejemplo. Se puede cambiar este comportamiento estableciendo el valor de la variable `toplevel-book-handler` en el nivel superior. El *handler* predeterminado se define en el archivo de inicio `../scm/lily.scm`.
- Un bloque `\bookpart`. Un libro puede dividirse en varias partes, utilizando bloques `\bookpart`, con objeto de facilitar la inserción de saltos de página o usar distintos ajustes de `\paper` en las distintas partes.
- Una expresión musical compuesta, como
 

```
{ c'4 d' e'2 }
```

Esto añade la pieza al `\score` y la conforma dentro de un libro único junto a todas las demás partituras `\score` y expresiones musicales del nivel superior. En otras palabras, un archivo que contiene sólo la expresión musical anterior se traduce a

```
\book {
  \score {
    \new Staff {
      \new Voice {
        { c'4 d' e'2 }
      }
    }
    \layout { }
  }
  \paper { }
  \header { }
}
```

Se puede cambiar este comportamiento estableciendo el valor de la variable `toplevel-music-handler` en el nivel más alto. El *handler* predeterminado se define en el archivo de inicio `../scm/lily.scm`.

- Un texto de marcado, por ejemplo una estrofa

```
\markup {
  2. Estrofa dos, línea uno
}
```

Los textos de marcado se colocan encima, en medio o debajo de las partituras o expresiones musicales, según donde aparezcan escritos.

- Una variable, como

```
fulano = { c4 d e d }
```

Esto se puede utilizar más tarde en el archivo escribiendo `\fulano`. El nombre de las variables ha de ser exclusivamente alfabético; sin números, guiones ni barras bajas.

El ejemplo siguiente muestra tres cosas que se pueden introducir en el nivel superior:

```
\layout {
  % No justificar la salida
  ragged-right = ##t
}

\header {
  title = "Do-re-mi"
}

{ c'4 d' e2 }
```

En cualquier punto del archivo se pueden introducir cualquiera de las instrucciones léxicas siguientes:

- `\version`
- `\include`
- `\sourcefilename`
- `\sourcefileline`
- Un comentario de una línea, determinado por un símbolo `%` al principio.
- Un comentario de varias líneas, delimitado por `%{ ... %}`.

Por lo general se ignoran los espacios entre elementos del flujo de entrada, y se pueden omitir o aumentar tranquilamente para mejorar la legibilidad. Sin embargo, los espacios se deben utilizar siempre para evitar errores, en las siguientes situaciones:

- Alrededor de todas y cada una de las llaves curvas de apertura y cierre.
- A continuación de las instrucciones y variables, es decir, todos los elementos que comienzan con un signo `\` de barra invertida.
- A continuación de todo elemento que se deba interpretar como expresión de Scheme, es decir, todo elemento que comience por un signo `#`.
- Para separar los elementos de una expresión de Scheme.
- Dentro de `lyricmode` antes y después de las instrucciones `\set` y `\override`.

## Véase también

Manual de aprendizaje: Sección “Cómo funcionan los archivos de entrada de LilyPond” en *Manual de Aprendizaje*.

Referencia de la notación: [Explicación de los títulos], página 512, Sección 4.2.1 [El bloque `\layout`], página 579.

## 3.2 Títulos y encabezamientos

Casi toda la música impresa tiene un título y el nombre del compositor; ciertas piezas tienen mucha más información.

### 3.2.1 Crear títulos, encabezamientos y pies de página

#### Explicación de los títulos

Cada bloque `\book` de un solo archivo de entrada produce un archivo de salida distinto, véase Sección 3.1.5 [Estructura del archivo], página 510. Dentro de cada archivo de salida están disponibles tres tipos de zonas para títulos: *Títulos de libro* al principio de cada libro, *Títulos de parte de libro* al principio de cada parte de un libro y *Títulos de partitura* al comienzo de una partitura.

Los valores de los campos para encabezamientos tales como `title` (para el título) y `composer` (para el autor) se establecen en bloques `\header` (para ver la sintaxis de los bloques `\header` y una lista completa de los campos disponibles de forma predeterminada, consulte [Presentación predeterminada de los títulos de partes de libro y partitura], página 515). Tanto los títulos de libro como los títulos de parte de libro y los títulos de partitura pueden contener los mismos campos, aunque de forma predeterminada los campos de los títulos de partitura se limitan a `piece` (pieza) y `opus`.

Los bloques `\header` se pueden situar en cuatro lugares distintos para formar una jerarquía descendente de bloques `\header`:

- Al principio del archivo de entrada, antes de cualquier otro bloque `\book`, `\bookpart` o `\score`.
- Dentro de un bloque `\book` pero fuera de los bloques `\bookpart` y `\score` que están dentro de ese libro.
- Dentro de un bloque `\bookpart` pero fuera de cualquier bloque `\score` dentro de dicha parte de libro.
- Dentro de un bloque `\score`.

Los valores de los campos se filtran a través de esta jerarquía, persistiendo aquellos que se establecen más arriba a no ser que se sobrescriban por un valor establecido más abajo en la jerarquía, y así:

- El título de un libro deriva de los campos establecidos al principio del archivo de entrada, modificado por campos establecidos en el bloque `\book`. Los campos resultantes se usan para imprimir el título de libro para dicho libro, siempre y cuando haya otro material que genere una página al comienzo del libro, antes de la primera parte de libro. Es suficiente una sola instrucción `\pageBreak`.
- Un título de parte de libro se deriva de los campos fijados al principio del archivo de entrada, modificado por los campos establecidos dentro del bloque `\book`, y posteriormente modificado por los campos que se fijan dentro del bloque `\bookpart`. Los valores resultantes se usan para imprimir el título de parte de libro para esa parte.
- Un título de partitura se deriva de los campos que se fijan al principio del archivo de entrada, modificados por los campos fijados dentro del bloque `\book`, posteriormente modificados por los campos que se fijan dentro del bloque `\bookpart` y finalmente modificados por los campos establecidos dentro del bloque `\score`. Los valores resultantes se utilizan para imprimir el título de partitura. Observe, sin embargo, que de forma predeterminada solamente se imprimen los campos `piece` y `opus` en los títulos de partitura a no ser que la variable de `\paper`, `print-all-headers`, tenga el valor `#t`.

No es necesario aportar bloques `\header` en los cuatro lugares: cualquiera de ellos o incluso todos pueden omitirse. De forma similar, los archivos de entrada sencillos pueden omitir los bloques `\book` y `\bookpart`, dejando que se creen de forma implícita.

Si el libro solamente tiene una partitura, el bloque `\header` debiera situarse normalmente al principio del archivo de forma que solamente se produce una zona de título de parte de libro, haciendo que estén disponibles para su uso todos los títulos de encabezamientos.

Si el libro tiene varias partituras, son posibles distintas combinaciones de bloques `\header`, según los distintos tipos de publicaciones musicales. Por ejemplo, si la publicación contiene varias piezas del mismo autor, un bloque `\header` colocado al principio del archivo que especifique el título del libro y el autor, con bloques `\header` en cada bloque `\score` que especifique la `piece` y/o el `opus` sería lo más adecuado, como aquí:

```
\header {
  title = "SUITE I."
  composer = "J. S. Bach."
}

\score {
  \header {
    piece = "Prélude."
  }
  \new Staff \relative {
    \clef bass
    \key g \major
    \repeat unfold 2 { g,16( d' b') a b d, b' d, } |
    \repeat unfold 2 { g,16( e' c') b c e, c' e, } |
  }
}

\score {
  \header {
    piece = "Allemande."
  }
  \new Staff \relative {
    \clef bass
```

```

\key g \major
\partial 16 b16 |
<g, d' b'~>4 b'16 a( g fis) g( d e fis) g( a b c) |
d16( b g fis) g( e d c) b(c d e) fis( g a b) |
}
}

```

## SUITE I.

J. S. Bach.

Prélude.



Allemande.



Son posibles disposiciones más complejas. Por ejemplo, los campos de texto extraídos del bloque `\header` de un libro se pueden imprimir en todos los títulos de partitura, con ciertos campos sobrescritos y otros suprimidos manualmente:

```

\book {
  \paper {
    print-all-headers = ##t
  }
  \header {
    title = "DAS WOHLTEMPERIRTE CLAVIER"
    subtitle = "TEIL I"
    % Do not display the default LilyPond footer for this book
    tagline = ##f
  }
  \markup { \vspace #1 }
  \score {
    \header {
      title = "PRAELUDIUM I"
      opus = "BWV 846"
      % Do not display the subtitle for this score
      subtitle = ##f
    }
    \new PianoStaff <<
      \new Staff { s1 }
      \new Staff { \clef "bass" s1 }
    >>
  }
  \score {

```

```

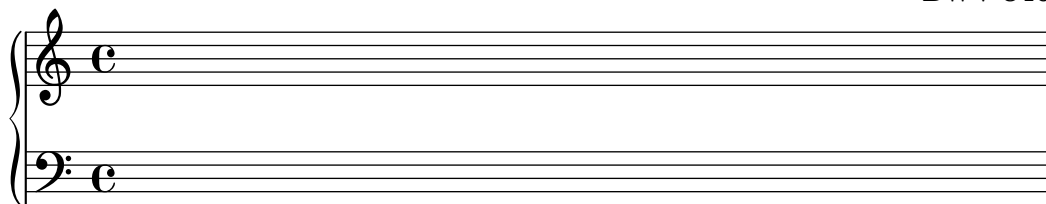
\header {
  title = "FUGA I"
  subsubtitle = "A 4 VOCI"
  opus = "BWV 846"
  % Do not display the subtitle for this score
  subtitle = ##f
}
\new PianoStaff <<
  \new Staff { s1 }
  \new Staff { \clef "bass" s1 }
>>
}

```

## DAS WOHLTEMPERIRTE CLAVIER TEIL I

### PRAELUDIUM I

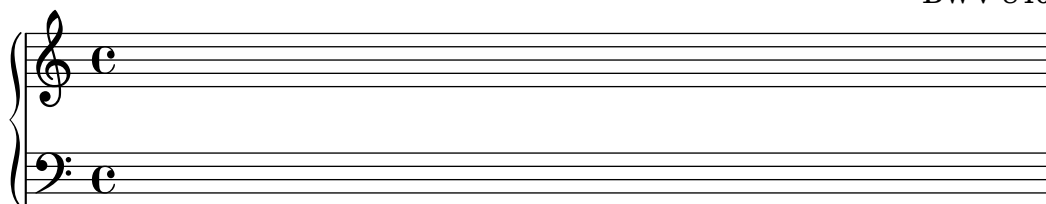
BWV 846



### FUGA I

A 4 VOCI

BWV 846



#### Véase también

Referencia de la notación: Sección 3.1.5 [Estructura del archivo], página 510, [Presentación predeterminada de los títulos de partes de libro y partitura], página 515, [Personalización de los títulos], página 520.

#### Presentación predeterminada de los títulos de partes de libro y partitura

Este ejemplo muestra todas las variables impresas de `\header`:

```

\book {
  \header {

```



```

    % Los siguientes campos están centrados
    dedication = "Dedication (dedicatoria)"
    title = "Title (título)"
    subtitle = "Subtitle (subtítulo)"
    subsubtitle = "Subsubtitle (sub-subtítulo)"
    % Los siguientes campos se distribuyen regularmente sobre una línea
    % el campo "instrument" también aparece en las siguientes páginas
    instrument = \markup \with-color #green "Instrument (instrumento)"
    poet = "Poet (poeta)"
    composer = "Composer (autor)"
    % Los siguientes campos se sitúan en los extremos opuestos de la misma línea
    meter = "Meter (metro)"
    arranger = "Arranger (arreglista)"
    % Los siguientes campos están centrados en la parte inferior
    tagline = "La tagline o línea de etiqueta va al final de la última página"
    copyright = "El copyright va al final de la primera página"
}
\score {
  \header {
    % Los siguientes campos se colocan en los extremos de una línea
    piece = "Piece (pieza) 1"
    opus = "Opus 1"
  }
  { s1 }
}
\score {
  \header {
    % Los siguientes campos se sitúan en los extremos de una línea
    piece = "Piece (pieza) 2 en la misma página"
    opus = "Opus 2"
  }
  { s1 }
}
\pageBreak
\score {
  \header {
    % Los siguientes campos se sitúan en los extremos de una línea
    piece = "Piece (pieza) 3 en una página nueva"
    opus = "Opus 3"
  }
  { s1 }
}

```

}

Dedication (dedicatoria)

**Title (título)****Subtitle (subtítulo)****Subsubtitle (sub-subtítulo)**

Poet (poeta)

**Instrument (instrumento)**

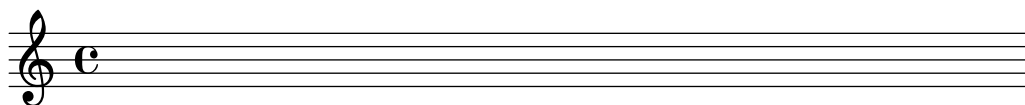
Composer (autor)

Meter (metro)

Arranger (arreglista)

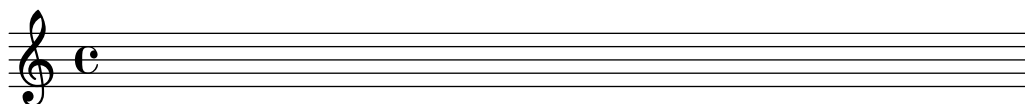
Piece (pieza) 1

Opus 1



Piece (pieza) 2 en la misma página

Opus 2



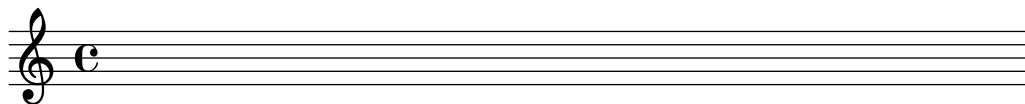
El copyright va al final de la primera página

2

**Instrument (instrumento)**

Piece (pieza) 3 en una página nueva

Opus 3



La tagline o línea de etiqueta va al final de la última página

Observe que

- El nombre de instrumento se repite en todas las páginas.
- Solamente **piece** (la pieza) y **opus** se imprimen en una `\score` cuando la variable de papel `print-all-headers` se establece a `##f` (el valor predeterminado).

- Los campos de texto que se dejan sin establecer dentro de un bloque `\header` se sustituyen por elementos de marcado nulos `\null` de manera que no ocupen espacio innecesariamente.
- Los ajustes predeterminados para `scoreTitleMarkup` sitúan los campos de texto `piece` y `opus` en extremos opuestos de la misma línea.

Para cambiar la disposición predeterminada, véase [Personalización de los títulos], página 520.

Si un bloque `\book` comienza inmediatamente con un bloque `\bookpart`, no se imprime ningún título de libro, pues no existe ninguna página en que imprimirlo. Si es necesario un título de libro, inicie el bloque `\book` con algún material de marcado o con una instrucción `\pageBreak`.

Utilice la variable `breakbefore` dentro de un bloque `\header` que está propiamente dentro de un bloque `\score` para hacer que los títulos del bloque `\header` del nivel superior aparezcan ocupando toda la primera página, empezando la música (definida en el bloque `\score`) en la página siguiente.

```
\book {
  \header {
    title = "Este es mi Title (título)"
    subtitle = "Este es mi Subtitle (subtítulo)"
    copyright = "Este es el final de la primera página"
  }
  \score {
    \header {
      piece = "Esto es la música"
      breakbefore = ##t
    }
    \repeat unfold 4 { e'' e'' e'' e'' }
  }
}
```

**Este es mi Title (título)**  
**Este es mi Subtitle (subtítulo)**

Este es el final de la primera página

2

Esto es la música



Music engraving by LilyPond 2.23.3—[www.lilypond.org](http://www.lilypond.org)

## Véase también

Manual de aprendizaje: Sección “Cómo funcionan los archivos de entrada de LilyPond” en *Manual de Aprendizaje*

Referencia de la notación: [Personalización de los títulos], página 520, Sección 3.1.5 [Estructura del archivo], página 510.

Archivos de inicio: `ly/titling-init.ly`.

## Disposición predeterminada de las cabeceras y pies de página

Las *cabeceras* y los *pies* de página son líneas de texto que aparecen en la parte más alta y en la más baja de las páginas separadas del texto principal de un libro. Se controlan mediante las siguientes variables de `\paper`:

- `oddHeaderMarkup` (marcado de cabecera impar)
- `evenHeaderMarkup` (marcado de cabecera par)
- `oddFooterMarkup` (marcado de pie impar)
- `evenFooterMarkup` (marcado de pie par)

Estas variables de marcado sólo pueden acceder a los campos de texto extraídos de bloques `\header` del nivel superior (que se aplican a todas las partituras del libro) y se definen en el archivo `ly/titling-init.ly`. De forma predeterminada:

- los números de página se sitúan automáticamente en el extremo superior izquierdo (si es par) o superior derecho (si es impar), a partir de la segunda página.
- el campo de texto `instrument` se sitúa en el centro en cada página, a partir de la segunda página.
- el texto de `copyright` se centra en la parte baja de la primera página.
- la línea `tagline` se centra al final de la última página, y debajo del texto de `copyright` si sólo hay una página.

La línea predeterminada de pie de página de LilyPond se puede cambiar añadiendo un campo `tagline` en el bloque `\header` del nivel superior.

```
\book {
  \header {
    tagline = "... music notation for Everyone"
  }
  \score {
    \relative {
```

```

      c'4 d e f
    }
  }
}

```



... music notation for Everyone

Para eliminar la línea de pie de página predeterminada de LilyPond, establezca el valor de `tagline` a `##f`.

### 3.2.2 Títulos, encabezamientos y pies de página personalizados

#### Formateo personalizado del texto de los bloques de título

Se pueden usar instrucciones `\markup` estándar para personalizar el texto de cualquier cabecera, pie o título dentro del bloque `\header`.

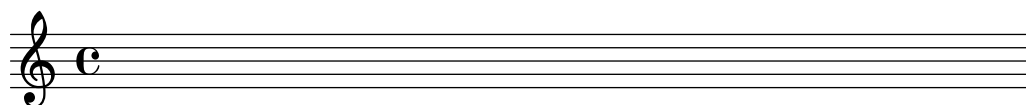
```

\score {
  \header {
    piece = \markup { \fontsize #4 \bold "PRAELUDIUM I" }
    opus  = \markup { \italic "BWV 846" }
  }
  { s1 }
}

```

**PRAELUDIUM I**

*BWV 846*



#### Véase también

Referencia de la notación: Sección 1.8.2 [Formatear el texto], página 264.

#### Personalización de los títulos

Las instrucciones `\markup` dentro del bloque `\header` son útiles para dar un formato simple al texto, pero no permiten un control preciso sobre la colocación de los títulos. Para personalizar la colocación de los campos de texto, cambie una o las dos variables de `\paper` siguientes:

- `bookTitleMarkup` (marcado de título del libro)
- `scoreTitleMarkup` (marcado de título de la partitura)

La colocación de los títulos cuando se utilizan los valores predeterminados de estas variables de `\markup` se muestra en los ejemplos de [Presentación predeterminada de los títulos de partes de libro y partitura], página 515.

Los ajustes predeterminados para `scoreTitleMarkup` tal y como están definidos en el archivo `ly/titling-init.ly` son:

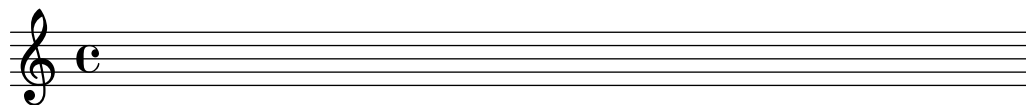
```
scoreTitleMarkup = \markup { \column {
  \on-the-fly \print-all-headers { \bookTitleMarkup \hspace #1 }
  \fill-line {
    \fromproperty #'header:piece
    \fromproperty #'header:opus
  }
}
}
```

Esto sitúa los campos de texto `piece` y `opus` en extremos opuestos de la misma línea:

```
\score {
  \header {
    piece = "PRAELUDIUM I"
    opus = "BWV 846"
  }
  { s1 }
}
```

PRAELUDIUM I

BWV 846

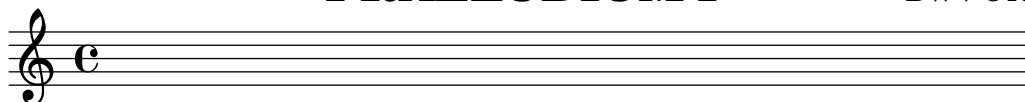


Este ejemplo redefine `scoreTitleMarkup` de manera que el campo de texto `piece` aparece centrado y en un tipo de letra grande y en negrita.

```
\book {
  \paper {
    indent = 0\mm
    scoreTitleMarkup = \markup {
      \fill-line {
        \null
        \fontsize #4 \bold \fromproperty #'header:piece
        \fromproperty #'header:opus
      }
    }
  }
  \header { tagline = ##f }
  \score {
    \header {
      piece = "PRAELUDIUM I"
      opus = "BWV 846"
    }
    { s1 }
  }
}
```

**PRAELUDIUM I**

BWV 846



Los campos de texto que normalmente no son efectivos dentro de los bloques `\header` de una partitura se pueden imprimir en la zona del Título de la partitura si se coloca `print-all-headers` dentro del bloque `\paper`. Una desventaja de la utilización de este método es que los campos de texto que están orientados específicamente para la zona del título de parte de libro han de suprimirse manualmente en cada uno de los bloques `\score`. Véase [Explicación de los títulos], página 512.

Para evitarlo, añada el campo de texto deseado a la definición de `scoreTitleMarkup`. En el ejemplo siguiente, el campo de texto `composer` (asociado normalmente con `bookTitleMarkup`) se añade a `scoreTitleMarkup`, permitiendo que cada partitura muestre un compositor diferente:

```
\book {
  \paper {
    indent = 0\mm
    scoreTitleMarkup = \markup {
      \fill-line {
        \null
        \fontsize #4 \bold \fromproperty #'header:piece
        \fromproperty #'header:composer
      }
    }
  }
  \header { tagline = ##f }
  \score {
    \header {
      piece = "MENUET"
      composer = "Christian Petzold"
    }
    { s1 }
  }
  \score {
    \header {
      piece = "RONDEAU"
      composer = "François Couperin"
    }
    { s1 }
  }
}
```



También podemos crear nuestros propios campos de texto personalizados, y referirnos a ellos en la definición del elemento de marcado.

```
\book {
```

```

\paper {
  indent = 0\mm
  scoreTitleMarkup = \markup {
    \fill-line {
      \null
      \override #`(direction . ,UP)
      \dir-column {
        \center-align \fontsize #-1 \bold
        \fromproperty #'header:mycustomtext %% User-defined field
        \center-align \fontsize #4 \bold
        \fromproperty #'header:piece
      }
      \fromproperty #'header:opus
    }
  }
}
\header { tagline = ##f }
\score {
  \header {
    piece = "FUGA I"
    mycustomtext = "A 4 VOICI" %% User-defined field
    opus = "BWV 846"
  }
  { s1 }
}

```



## Véase también

Referencia de la notación: [Explicación de los títulos], página 512.

## Disposición personalizada de cabeceras y pies de página

Las instrucciones `\markup` dentro del bloque `\header` son de utilidad para dar formato al texto de una manera sencilla, pero no permiten un control preciso sobre la colocación de las cabeceras y los pies de página. Para personalizar la colocación de los campos de texto, use una o más de las siguientes variables de `\paper`:

- `oddHeaderMarkup` (marcado de encabezamiento impar)
- `evenHeaderMarkup` (marcado de encabezamiento par)
- `oddFooterMarkup` (marcado de pie de página impar)
- `evenFooterMarkup` (marcado de pie de página par)

La instrucción de marcado `\on-the-fly` se puede utilizar para añadir elementos de forma condicional al texto de encabezamiento y pie de página definido dentro del bloque `\paper`, usando la sintaxis siguiente:

```

variable = \markup {
  ...

```



```

\on-the-fly \procedimiento marcado
...
}

```

El *procedimiento* se llama cada vez que se evalúa la instrucción `\markup` en que ésta aparece. El *procedimiento* debería comprobar si se cumple una condición determinada e interpretar (es decir: imprimir) el argumento *marcado* si, y sólo si, la condición es verdadera.

Se proveen un cierto número de procedimientos ya hechos para la comprobación de diversas condiciones:

Nombre del procedimiento	Condición que se comprueba
<code>print-page-number-check-first</code>	¿debería imprimirse este número de página?
<code>create-page-number-stencil</code>	¿es <code>print-page-numbers</code> verdadero?
<code>print-all-headers</code>	¿es <code>print-all-headers</code> verdadero?
<code>first-page</code>	¿es la primera página del libro?
<code>not-first-page</code>	¿no es la primera página del libro?
<code>(on-page nmbr)</code>	¿es el número de página = <code>nmbr</code> ?
<code>last-page</code>	¿es la última página del libro?
<code>part-first-page</code>	¿es la primera página de la parte de libro?
<code>not-part-first-page</code>	¿no es la primera página de la parte de libro?
<code>part-last-page</code>	¿es la última página de la parte de libro?
<code>not-single-page</code>	¿es el núm. de páginas en la parte de libro > 1?

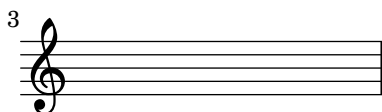
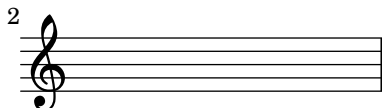
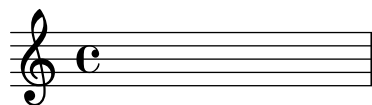
El ejemplo siguiente centra los números de página en la parte baja de las páginas. En primer lugar, los ajustes predeterminados para `oddHeaderMarkup` y `evenHeaderMarkup` se eliminan definiéndolos como un marcado *nulo*. Después de esto, se redefine `oddFooterMarkup` con el número de página centrado. Finalmente, `evenFooterMarkup` recibe la misma disposición definiéndola como `\oddFooterMarkup`:

```

\book {
  \paper {
    print-page-number = ##t
    print-first-page-number = ##t
    oddHeaderMarkup = \markup \null
    evenHeaderMarkup = \markup \null
    oddFooterMarkup = \markup {
      \fill-line {
        \on-the-fly \print-page-number-check-first
        \fromproperty #'page:page-number-string
      }
    }
    evenFooterMarkup = \oddFooterMarkup
  }
  \score {
    \new Staff { s1 \break s1 \break s1 }
  }
}

```

}



1

Se pueden combinar varias condiciones de `\on-the-fly` con un operador ‘and’, por ejemplo:

```
\on-the-fly \first-page
\on-the-fly \last-page
{ \markup ... \fromproperty #'header: ... }
```

determina si la salida es una sola página.

## Véase también

Referencia de la notación: [Explicación de los títulos], página 512, [Presentación predeterminada de los títulos de partes de libro y partitura], página 515.

Archivos instalados: `../ly/titling-init.ly`.

### 3.2.3 Crear metadatos en los archivos de salida

Además de mostrarse en la salida impresa, las variables de encabezamiento de `\header` se usan para establecer los metadatos para los archivos de salida. Por ejemplo, con los archivos PDF, estos metadatos pueden mostrarse por parte de los lectores de documentos como **propiedades** del PDF. Para cada tipo de archivo de salida, solamente se consultan las definiciones de `\header` de los bloques que definen archivos separados de ese tipo, y los bloques más altos en la jerarquía de bloques. Por tanto, para los archivos PDF, solo afectan a los metadatos de PDF de cada documento las definiciones de `\header` de los niveles de `\book` y de nivel superior, mientras que para archivos MIDI, se usan todas las cabeceras que están por encima o al nivel de `\score`.

Por ejemplo, si se establece la propiedad `title` del bloque `header` como ‘Sinfonía núm. 1’, el documento PDF llevará también este título, y lo usará como el nombre de la secuencia del archivo MIDI.

```
\header {
  title = "Sinfonía núm. 1"
}
```

Si quiere fijar el título de la salida impresa con un texto pero quiere que la propiedad de título del PDF tenga un texto diferente, puede usar `pdftitle` como se muestra a continuación.

```
\header {
  title = "Sinfonía núm. 1"
  pdftitle = "Sinfonía núm. 1 de Beethoven"
}
```

Todas las variables `title`, `subject`, `keywords`, `subtitle`, `composer`, `arranger`, `poet`, `author` y `copyright` establecen propiedades del PDF y pueden hacerse preceder de ‘pdf’ para fijar una propiedad del archivo PDF a un valor distinto del de la salida impresa.

La propiedad de PDF Creator se fija automáticamente a ‘LilyPond’ más la versión actual de LilyPond, y las variables `CreationDate` y `ModDate` se fijan a la fecha y hora actuales. Se puede sobrescribir `ModDate` mediante el establecimiento de un valor para la variable de cabecera `moddate` (o `pdfmoddate`) a una cadena de fecha de PDF válida.

La variable `title` establece también el nombre de la secuencia para el archivo MIDI. Se puede usar la variable `midititle` para fijar el nombre de la secuencia independientemente del valor usado para la salida impresa.

### 3.2.4 Crear notas al pie

Se pueden crear dos tipos de notas al pie: automáticas y manuales.

#### Notas al pie dentro de expresiones musicales

##### *Panorámica de las notas al pie*

Las notas al pie dentro de expresiones musicales caen dentro de dos categorías:

##### *Notas al pie basadas en eventos*

Se adjuntan a un evento en particular. Son ejemplos de tales eventos las notas sueltas, las articulaciones (como indicaciones de digitación, acentos, indicaciones dinámicas) y los post-eventos (como las ligaduras de expresión y las barras de corchea manuales). La forma general de las notas al pie basadas en eventos es como sigue:

`[dirección] \footnote [marca] desplazamiento texto-de-la-nota música`

##### *Notas al pie basadas en tiempo*

Se ligan a un punto temporal determinado, dentro de un contexto musical. Algunas instrucciones como `\time` y `\clef` en realidad no usan eventos para crear objetos como la indicación de compás y la clave. Un acorde tampoco crea un evento por sí mismo: su plica o corchete se crea al final de un paso de tiempo (nominalmente, a través de uno de los eventos de nota que contiene). Una nota al pie basada en tiempo permite anotar tales objetos de presentación sin referirse a ningún evento.

Una nota al pie basada en tiempo permite que tales objetos de presentación se puedan anotar sin hacer referencia a un evento. La forma general para las notas al pie basadas en tiempo es:

`\footnote [marca] desplazamiento texto-de-la-nota [Contexto].Nombre-del-Grob`

Los elementos para las dos formas son como sigue:

- |                       |  |
|-----------------------|--|
| <i>dirección</i>      | Si (y sólo si) el <code>\footnote</code> se está aplicando a un post-evento o articulación, debe ir precedida de un indicador de dirección ( <code>-</code> , <code>_</code> , <code>^</code> ) con el objeto de adjuntar la <i>música</i> (con una marca de nota al pie) a la nota o silencio precedente.   |
| <i>marca</i>          | es un elemento de marcado o cadena de caracteres que especifica la marca de nota al pie que se usa para marcar tanto el punto de referencia como la propia nota al pie en la parte inferior de la página. Se puede omitir (o, de forma equivalente, sustituirse por <code>\default</code> ) en cuyo caso se genera un número secuencialmente de forma automática. Tales secuencias numéricas se reinician en cada página que contiene una nota al pie. |
| <i>desplazamiento</i> | es una pareja de números tal como ‘ <code>#(2 . 1)</code> ’ que especifica los desplazamientos en X y en Y en unidades de espacios de pentagrama a partir del límite del objeto en   |

que se desea situar la marca. Los valores de desplazamiento positivos se toman a partir del borde superior derecho, los valores negativos a partir del borde inferior izquierdo y cero implica que la marca se centra sobre el borde.

*Contexto* es el contexto en que se crea el grob que recibe la nota al pie. Se puede omitir si el grob está en un contexto del nivel inferior, p. ej. un contexto *Voice*.

*Nombre-del-Grob*

especifica un tipo de grob que marcar (como 'Flag' para el corchete). Si se da, la nota al pie no se adjunta a una expresión musical en particular, sino a todos los grobs del tipo especificado que ocurren en dicho momento de tiempo musical.

*texto-de-la-nota*

es el elemento de marcado o cadena de caracteres que especifica el texto de la nota al pie que utilizar en la parte baja de la página.

*música* es el evento musical, post-evento o articulación que se está anotando.

### *Notas al pie basadas en eventos*

Una nota al pie se adjunta a un objeto de presentación causado directamente por el evento que corresponde a *música*, con la sintaxis siguiente:

```
\footnote [marca] desplazamiento texto-de-la-nota música
\book {
  \header { tagline = ##f }
  \relative c'' {
    \footnote #'(-1 . 3) "Una nota" a4
    a4
    \footnote #'(2 . 2) "Un silencio" r4
    a4
  }
}
```




---

<sup>1</sup>Una nota  
<sup>2</sup>Un silencio

No es posible marcar un acorde *completo* con una nota al pie basada en evento: un acorde, incluso aquel que contiene una sola nota, no produce un verdadero evento por sí mismo. Sin embargo, las notas individuales que están *dentro* del acorde sí se pueden marcar:

```
\book {
  \header { tagline = ##f }
  \relative c'' {
    \footnote #'(2 . 3) "No funciona" <a-3>2
    <\footnote #'(-2 . -3) "Funciona" a-3>4
    <a-3 \footnote #'(3 . 1/2) "También funciona" c-5>4
  }
}
```

}



<sup>1</sup>No funciona  
<sup>2</sup>Funciona  
<sup>3</sup>También funciona

Si se desea que la nota al pie se adjunte a un post-evento o articulación, la instrucción `\footnote` debe ir precedida por un indicador de dirección, `-`, `_`, `^`, y seguida por el post-evento o articulación que se desea anotar como argumento *música*. En esta forma, la `\footnote` se puede considerar como una simple copia de su último argumento con una marca de nota al pie adjunta. La sintaxis es:

```

dirección \footnote [marca] desplazamiento texto-de-la-nota música
\book {
  \header { tagline = ##f }
  \relative {
    a'4_\footnote #'(0 . -1) "Ligadura forzada hacia abajo" (
    b8^\footnote #'(1 . 0.5) "Barra manual forzada hacia arriba" [
    b8 ]
    c4 )
    c-\footnote #'(1 . 1) "Tenuto" --
  }
}

```



<sup>1</sup>Ligadura forzada hacia abajo  
<sup>2</sup>Barra manual forzada hacia arriba  
<sup>3</sup>Tenuto

### *notas al pie basadas en tiempo*

Si el objeto de presentación que recibe la nota al pie está causado *indirectamente* por un evento (como un objeto `Accidental`, alteración, o `Stem`, plica, causados por un evento `NoteHead`, cabeza de nota), el nombre del grob `GrobName` del objeto de presentación es necesario después del texto de la nota al pie, en lugar de *música*:

```

\book {
  \header { tagline = ##f }
  \relative c'' {
    \footnote #'(-1 . -3) "Un bemol" Accidental
  }
}

```

```

aes4 c
\footnote #'(-1 . 0.5) "Otro bemol" Accidental
ees
\footnote #'(1 . -2) "Una plica" Stem
aes
}
}

```



<sup>1</sup>Un bemol  
<sup>2</sup>Otro bemol  
<sup>3</sup>Una plica

Sin embargo, observe que cuando se especifica el nombre de un grob, se adjunta una nota al pie a todos los grobs de ese tipo dentro del instante de tiempo actual:

```

\book {
  \header { tagline = ##f }
  \relative c' {
    \footnote #'(-1 . 3) "Un bemol" Accidental
    <ees ges bes>4
    \footnote #'(2 . 0.5) "Articulación" Script
    c'->-.
  }
}

```



<sup>1</sup>Un bemol  
<sup>2</sup>Un bemol  
<sup>3</sup>Un bemol  
<sup>4</sup>Articulación  
<sup>5</sup>Articulación

Una nota dentro de un acorde puede recibir una nota al pie individual, basada en evento. El único grob causado directamente por una nota de un acorde es ‘**NoteHead**’, por lo que la instrucción de nota al pie basada en evento *solamente* es apta para añadir una nota al pie a la ‘**NoteHead**’ dentro de un acorde. Todos los demás grobs de notas de acorde están causados indirectamente. La propia instrucción `\footnote` no ofrece ninguna sintaxis para especificar al mismo tiempo un tipo concreto de grob *y también* un evento concreto al que adjuntar la nota. Sin embargo, se puede usar una instrucción `\footnote` basada en tiempo para especificar el tipo de grob, y luego añadir la instrucción `\single` como prefijo, con el objeto de que se aplique solamente al evento siguiente:

```

\book {

```

```

\header { tagline = ##f }
\relative c'' {
  < \footnote #'(1 . -2) "Un La" a
    \single \footnote #'(-1 . -1) "Un sostenido" Accidental
    cis
    \single \footnote #'(0.5 . 0.5) "Un bemol" Accidental
    ees fis
  >2
}
}

```



<sup>1</sup>Un bemol  
<sup>2</sup>Un sostenido  
<sup>3</sup>Un La

**Nota:** Cuando las notas al pie se adjuntan a varios elementos musicales que están dentro del mismo momento musical, como ocurre en el ejemplo anterior, las notas al pie se numeran desde los elementos más altos a los más bajos tal y como aparecen en la salida impresa, no en el orden en que se escriben en el flujo de entrada.

Los objetos de presentación como las claves y las armaduras de cambio de tonalidad están, más bien, causadas por propiedades que cambian, y no por eventos. Otros, como las líneas divisorias y los números de compás, son una consecuencia directa del paso del tiempo. Por esta razón, las notas al pie sobre tales objetos tienen que basarse en su tiempo musical. Las notas al pie basadas en tiempo son preferibles también cuando se marcan cosas como las plicas y las barras de corchea en *acordes*: dado que estas funcionalidades de cada acorde individual están asignadas normalmente a *un solo* evento dentro del acorde, sería imprudente descansar en una elección particular.

En las notas al pie basadas en tiempo, siempre se debe especificar explícitamente el objeto de presentación en cuestión, y si el grob se crea dentro de un contexto distinto al de nivel más bajo, debe especificarse el contexto apropiado.

```

\book {
  \header { tagline = ##f }
  \relative c'' {
    r1 |
    \footnote #'(-0.5 . -1) "Cambio de compás" Staff.TimeSignature
    \time 3/4
    \footnote #'(1 . -1) "Plica de acorde" Stem
    <c e g>4 q q
    \footnote #'(-0.5 . 1) "Línea divisoria" Staff.BarLine
    q q
    \footnote #'(0.5 . -1) "Cambio de tonalidad" Staff.KeySignature
  }
}

```

```

\key c\minor
q
}
}

```



- 
- <sup>1</sup>Cambio de compás  
<sup>2</sup>Plica de acorde  
<sup>3</sup>Línea divisoria  
<sup>4</sup>Cambio de tonalidad

Se pueden usar marcas personalizadas como alternativas a las marcas numéricas, y se puede suprimir la línea de anotación que une el objeto marcado con su marca:

```

\book {
  \header { tagline = ##f }
  \relative c' {
    \footnote "*" #'(0.5 . -2) \markup { \italic "*" La primera nota" } a'4
    b8
    \footnote \markup { \super "$" } #'(0.5 . 1)
    \markup { \super "$" \italic " La segunda nota" } e
    c4
    \once \override Score.FootnoteItem.annotation-line = ##f
    b-\footnote \markup \tiny "+" #'(0.1 . 0.1)
    \markup { \super "+" \italic " Editorial" } \p
  }
}

```



- 
- \* *La primera nota*  
 \$ *La segunda nota*  
 + *Editorial*

Se muestran más ejemplos de marcas personalizadas en [Notas al pie en texto independiente], página 531.

## Notas al pie en texto independiente

Se utilizan dentro de un elemento de marcado fuera de cualquier expresión musical. No se traza una línea hasta el punto de referencia: la marca simplemente sigue al elemento de marcado que se referencia. Se pueden insertar las marcas automáticamente, en cuyo caso son numéricas. De forma alternativa, se pueden aportar manualmente marcas personalizadas.



Las notas al pie para texto independiente con marcas automáticas y personalizadas se crean de distintas formas.

### *Notas al pie en texto independiente con marcas automáticas*

La sintaxis de una nota al pie sobre texto independiente con marcas automáticas es

```
\markup { ... \footnote texto nota ... }
```

Los elementos son:

*texto* es el elemento de marcado o cadena de caracteres que se desea anotar.

*nota* es el elemento de marcado o cadena de caracteres que especifica el texto de la nota al pie que utilizar en la parte baja de la página.

Por ejemplo:

```
\book {
  \header { tagline = ##f }
  \markup {
    "A simple"
    \footnote "tune" \italic " By me"
    "is shown below. It is a"
    \footnote "recent" \italic " Aug 2012"
    "composition."
  }
  \relative {
    a'4 b8 e c4 d
  }
}
```

A simple tune is shown below. It is a recent composition.



*By me*  
*Aug 2012*

### *Notas al pie sobre texto independiente con marcas personalizadas*

La sintaxis de una nota al pie sobre texto independiente con marcas personalizadas es

```
\markup { ... \footnote marca nota ... }
```

Los elementos son:

*marca* es un elemento de marcado o cadena de caracteres que especifica la marca que se usa para señalar el punto de referencia. Observe que esta marca *no* se inserta automáticamente antes de la propia nota al pie.

*nota* es el elemento de marcado o cadena de caracteres que especifica el texto de la nota que aparece en la parte baja de la página, precedida por la *marca* especificada.

Se puede usar como marca un carácter fácil de teclear, como \* ó +, según se muestra en [Notas al pie dentro de expresiones musicales], página 526. De forma alternativa, se pueden usar alias en ASCII (véase [Alias de ASCII], página 547):

```
\book {
  \paper { #(include-special-characters) }
  \header { tagline = ##f }
  \markup {
    "A simple tune"
    \footnote "*" \italic "* By me"
    "is shown below. It is a recent"
    \footnote \super &dagger; \concat {
      \super &dagger; \italic " Aug 2012"
    }
    "composition."
  }
  \relative {
    a'4 b8 e c4 d
  }
}
```

A simple tune \* is shown below. It is a recent<sup>†</sup> composition.




---

\* *By me*

<sup>†</sup> *Aug 2012*

También pueden utilizarse códigos de caracteres Unicode para especificar las marcas (véase [Unicode], página 546):

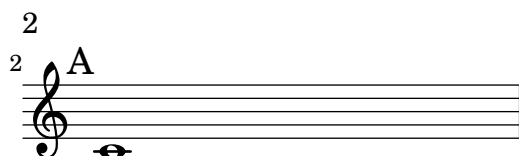
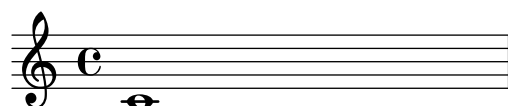
```
\book {
  \header { tagline = ##f }
  \markup {
    "A simple tune"
    \footnote \super \char##x00a7 \concat {
      \super \char##x00a7 \italic " By me"
    }
    "is shown below. It is a recent"
    \footnote \super \char##x00b6 \concat {
      \super \char##x00b6 \italic " Aug 2012"
    }
  }
}
```



```

{
  c'1
  \pageBreak \mark A \label #'markA
  c'1
}
}
\markup { La primera partitura comienza en la página \page-ref #'firstScore "0" "?" }
\markup { La marca A está en la página \page-ref #'markA "0" "?" }
}

```



La primera partitura comienza en la página 1

La marca A está en la página 2

La instrucción de marcado `\page-ref` toma tres argumentos:

1. la etiqueta, un símbolo de Scheme, p.ej.  `#'firstScore`;
2. un elemento de marcado que se usará como medidor para estimar las dimensiones del marcado;
3. un elemento de marcado que se utilizará en sustitución del número de página si la etiqueta es desconocida.

El motivo de que se necesite un medidor es que en el momento en que se están interpretando los marcados, los saltos de página aún no se han producido y por tanto los números de página no se conocen todavía. Para sortear este inconveniente, la interpretación real del marcado se retrasa hasta un momento posterior; sin embargo, las dimensiones del marcado se tienen que conocer de antemano, así que se usa el medidor para decidir estas dimensiones. Si el libro tiene entre 10 y 99 páginas, el medidor puede ser `"00"`, es decir, un número de dos dígitos.

## Instrucciones predefinidas

`\label`, `\page-ref`.

### 3.2.6 Índice general

Se puede insertar un índice general o tabla de contenidos utilizando la instrucción `\markuplist` `\table-of-contents`. Los elementos que deben aparecer en la tabla de contenidos se introducen con la instrucción `\tocItem`, que se puede usar en el nivel más alto de la jerarquía del código, o dentro de una expresión musical.

```

\markuplist \table-of-contents
\pageBreak

```

```

\tocItem \markup "Primera partitura"

```

```

\score {
  {
    c'4 % ...
    \tocItem \markup "Un punto concreto dentro de la primera partitura"
    d'4 % ...
  }
}

\tocItem \markup "Segunda partitura"
\score {
  {
    e'4 % ...
    \tocItem actI \markup "Act I"
    f'4 % ...
    \tocItem actI.sceneI \markup "Scene 1"
    g'4 % ...
    \tocItem actI.sceneI.recitativo \markup "Recit."
    a'4 % ...
  }
}

```

De manera opcional se puede asociar una etiqueta con un elemento en particular o con una lista jerárquica de etiquetas existentes, terminando con la etiqueta de ese elemento. Esta última posibilidad nos permite marcar el elemento como un ‘hijo’ de los elementos etiquetados precedentes, haciendo así aparente la estructura de la partitura en el índice general.

Los elementos de marcado que se usan para dar formato al índice general o tabla de contenidos se encuentran definidos dentro del bloque `\paper`. Hay tres elementos de marcado ‘predefinidos’ disponibles:

- `tocTitleMarkup`

Utilizado para dar formato al título del índice general.

```

tocTitleMarkup = \markup \huge \column {
  \fill-line { \null "Índice general" \null }
  \null
}

```

- `tocItemMarkup`

Usado para dar formato a los elementos del índice general.

```

tocItemMarkup = \markup \fill-line {
  \fromproperty #'toc:text \fromproperty #'toc:page
}

```

- `tocFormatMarkup`

Cómo se formatean las entradas del nivel superior de la tabla (si existen varios niveles jerárquicos). Esto es un procedimiento en realidad, como se explica en Sección “Construcción de elementos de marcado en Scheme” en *Extender*.

```

tocFormatMarkup = #make-bold-markup

```

- `tocIndentMarkup`

Usado para definir cómo la jerarquía del esquema general se hace aparente. Este elemento de marcado se imprime ninguna, una o varias veces dependiendo del nivel de cada entrada.

```

tocIndentMarkup = \markup \hspace #4

```

Cualquiera de estas variables puede modificarse.

He aquí un ejemplo que traduce el título del índice general al francés:

```
\paper {
  tocTitleMarkup = \markup \huge \column {
    \fill-line { \null "Table des matières" \null }
    \hspace #1
  }
}
```

A continuación aparece un ejemplo que modifica el tamaño de fuente tipográfica de los elementos del índice general:

```
tocItemMarkup = \markup \large \fill-line {
  \fromproperty #'toc:text \fromproperty #'toc:page
}
```

Observe la forma en que nos referimos al texto y al número de página del elemento de índice, dentro de la definición `tocItemMarkup`.

La instrucción `\tocItemWithDotsMarkup` se puede incluir dentro de `tocItemMarkup` para llenar la línea con puntos entre un elemento del índice general y su número de página correspondiente:

```
\header { tagline = ##f }
\paper {
  tocItemMarkup = \tocItemWithDotsMarkup
}

\book {
  \markuplist \table-of-contents
  \tocItem \markup { Allegro }
  \tocItem \markup { Largo }
  \markup \null
}
```

## Table of Contents

<b>Allegro</b> . . . . .	<b>1</b>
<b>Largo</b> . . . . .	<b>1</b>

Además del mecanismo incorporado de construcción del índice general, las instrucciones predefinidas pueden definirse también para que construyan un índice general más personalizado con diferentes elementos de marcado. En el ejemplo siguiente se define un estilo nuevo para introducir los actos y escenas de una ópera en el índice general:

Una nueva variable de marcado (llamada `tocActMarkup`) se define en el bloque `\paper`:

```
\paper {
  tocActMarkup = \markup \large \column {
    \hspace #1
    \fill-line { \null \italic \fromproperty #'toc:text \null }
    \hspace #1
  }
}
```

A continuación se crea una función musical personalizada (`tocAct`), que usa la nueva definición de marcado `tocActMarkup`, y permite especificar una etiqueta para cada acto.

```
tocAct =
  #(define-music-function (label text) (symbol? markup?)
```

```
(add-toc-item! 'tocActMarkup label text))
```

Usando estas definiciones personalizadas y modificando algunas de las definiciones existentes, el archivo fuente se podría escribir entonces de la manera siguiente:

## Table of Contents

### *Atto Primo*

Coro. Viva il nostro Alcide . . . . .	1
Cesare. Presti omai l'Egizia terra . . . . .	1
<i>Recit.</i> Curio, Cesare venne, e vide, e vinse. .	1

### *Atto Secondo*

Sinfonia . . . . .	1
Cleopatra. V'adoro, pupille, saette d'Amore . .	1

El ejemplo anterior muestra también cómo usar la instrucción de marcado `\fill-with-pattern` dentro del contexto de un índice general o tabla de contenidos.

## Véase también

Archivos de inicio: `ly/toc-init.ly`.

## Instrucciones predefinidas

`\table-of-contents`, `\tocItem`, `tocItemMarkup`, `tocTitleMarkup`, `tocFormatMarkup`, `tocIndentMarkup`.

## 3.3 Trabajar sobre los archivos de entrada

### 3.3.1 Inclusión de archivos de LilyPond

Un proyecto grande se puede dividir en varios archivos. Para referirse a otro archivo, utilice

```
\include "otroarchivo.ly"
```

La línea `\include "otroarchivo.ly"` equivale a pegar todo el contenido de `otroarchivo.ly` en el archivo actual en el lugar en que aparece el `\include`. Por ejemplo, en un proyecto grande podríamos querer archivos distintos para cada parte instrumental y crear un archivo de “partitura completa” que reúne los archivos individuales de los instrumentos. Normalmente el archivo incluido define un cierto número de variables que a partir de entonces quedan disponibles para poderlas utilizar en el archivo de la partitura completa. En los archivos incluidos se pueden marcar las secciones etiquetadas para ayudar a hacerlas utilizables en distintos lugares de la partitura, véase Sección 3.3.2 [Distintas ediciones a partir de una misma fuente], página 540.

Los archivos que están en el directorio de trabajo actual se pueden referenciar simplemente especificando el nombre después de la instrucción `\include`. Los archivos en otros lugares se pueden incluir proporcionando una referencia de ruta completa o una ruta relativa (pero utilice la barra inclinada normal del UNIX, `/`, no la barra invertida de DOS/Windows, `\`, como separador de directorio). Por ejemplo, si `material.ly` está situado un directorio por encima del directorio de trabajo actual, utilice

```
\include "../material.ly"
```

o si los archivos de las partes orquestales incluidas están todos situados en un subdirectorio llamado **partes** dentro del directorio actual, use

```
\include "partes/VI.ly"
\include "partes/VII.ly"
... etc.
```

Los archivos de inclusión también pueden contener enunciados `\include` a su vez. Estos enunciados `\include` de segundo nivel se interpretan de manera relativa a la ruta del archivo que contiene dicha instrucción, lo que es conveniente para varios archivos ubicados dentro del mismo subdirectorio. Por ejemplo, una biblioteca general, **bibA**, puede a su vez usar sub-archivos que son incluidos por el archivo principal de dicha biblioteca, así:

```
bibA/
  bibA.ly
  A1.ly
  A2.ly
  ...
```

entonces el archivo de entrada, **bibA.ly**, contiene

```
\include "A1.ly"
\include "A2.ly"
...
```

Cualquier archivo **.ly** puede entonces incluir la biblioteca completa, simplemente con

```
\include "~/bibA/bibA.ly"
```

Sin embargo, este comportamiento se puede modificar de forma global pasando la opción `-drelative-includes=#f` en la línea de órdenes (o añadiendo `#{ly:set-option 'relative-includes #f}` al principio del archivo principal de entrada). En tal caso, cada archivo se incluirá relativamente a la ubicación del archivo principal, sin importar dónde está situado su instrucción `\include`. Estructuras de archivo complejas que requieren realizar la inclusión tanto de archivos relativos al directorio principal como de archivos relativos a algún otro directorio, pueden incluso configurarse estableciendo **relative-includes** a los valores **#f** o **#t** en los lugares adecuados dentro de dichos archivos.

También se pueden incluir archivos de un directorio que está en una ruta de búsqueda especificada como opción al invocar a LilyPond desde la línea de órdenes. Los archivos incluidos se especifican usando solamente su nombre de archivo. Por ejemplo, para compilar mediante este método un archivo **principal.ly** que incluye archivos situados en un subdirectorio llamado **partes**, cambie al directorio que contiene **principal.ly** e introduzca

```
lilypond --include=partes principal.ly
y en principal.ly escriba
\include "VI.ly"
\include "VII.ly"
... etc.
```

Los archivos que se incluyen en muchas partituras se pueden poner en el directorio de LilyPond **../ly** (la localización de este directorio depende de la instalación: véase Sección “Otras fuentes de información” en *Manual de Aprendizaje*). Estos archivos se pueden incluir sencillamente por su nombre en un enunciado `\include`. De esta forma se incluyen los archivos dependientes del idioma como **espanol.ly**.

LilyPond incluye de forma predeterminada ciertos archivos cuando se inicia el programa. Estas inclusiones no son visibles para el usuario, pero los archivos se pueden identificar ejecutando `lilypond --verbose` desde la línea de órdenes. Esto presentará una lista de rutas y archivos que utiliza LilyPond, junto a muchas otras indicaciones. De forma alternativa, pueden verse explicaciones sobre los archivos más importantes de este conjunto en Sección “Otras fuentes



de información” en *Manual de Aprendizaje*. Estos archivos se pueden editar, pero los cambios realizados sobre ellos se perderán al instalar una nueva versión de LilyPond.

Pueden verse ejemplos sencillos sobre el uso de `\include` en Sección “Partituras y particellas” en *Manual de Aprendizaje*.

## Véase también

Manual de aprendizaje: Sección “Otras fuentes de información” en *Manual de Aprendizaje*, Sección “Partituras y particellas” en *Manual de Aprendizaje*.

## Advertencias y problemas conocidos

Si un archivo incluido recibe un nombre igual al de uno de los archivos de instalación de LilyPond, el archivo que tiene prioridad es el de los archivos de instalación de LilyPond.

### 3.3.2 Distintas ediciones a partir de una misma fuente

Se pueden usar varios métodos para generar la producción de distintas versiones de una partitura a partir de la misma fuente musical. Las variables son quizá el más útil para combinar secciones extensas de música y/o anotaciones. Las etiquetas son más útiles para seleccionar una sección de entre varias secciones de música alternativas, más cortas, y se pueden usar también para dividir piezas de música en distintos trozos.

Cualquiera que sea el método utilizado, la separación de la notación de la estructura de la partitura hará que sea más fácil cambiar la estructura dejando intacta la notación.

## Uso de las variables

Si las secciones de la música se definen dentro de variables, se pueden reutilizar en distintas partes de la partitura, véase Sección “Organizar las piezas mediante variables” en *Manual de Aprendizaje*. Por ejemplo, una partitura vocal *a cappella* con frecuencia incluye una reducción de piano de las partes para ensayar que es idéntica a la música vocal, por lo que ésta sólo se tiene que escribir una vez. Se puede combinar sobre un solo pentagrama la música de dos variables, véase [Combinación automática de las partes], página 192. He aquí un ejemplo:

```
sopranoMusic = \relative { a'4 b c b8( a) }
altoMusic = \relative { e'4 e e f }
tenorMusic = \relative { c'4 b e d8( c) }
bassMusic = \relative { a4 gis a d, }
allLyrics = \lyricmode { King of glo -- ry }
<<
  \new Staff = "Soprano" \sopranoMusic
  \new Lyrics \allLyrics
  \new Staff = "Alto" \altoMusic
  \new Lyrics \allLyrics
  \new Staff = "Tenor" {
    \clef "treble_8"
    \tenorMusic
  }
  \new Lyrics \allLyrics
  \new Staff = "Bass" {
    \clef "bass"
    \bassMusic
  }
  \new Lyrics \allLyrics
  \new PianoStaff <<
    \new Staff = "RH" {
```

```

    \partCombine \sopranoMusic \altoMusic
  }
  \new Staff = "LH" {
    \clef "bass"
    \partCombine \tenorMusic \bassMusic
  }
>>
>>

```



Se pueden producir partituras distintas que presenten sólo las partes vocales o sólo la parte de piano, cambiando solamente los enunciados estructurales, sin modificar la notación musical.

Para partituras extensas, las definiciones de variable se pueden poner en archivos separados que se incluyen más tarde, véase Sección 3.3.1 [Inclusión de archivos de LilyPond], página 538.

## Uso de etiquetas

La instrucción `\tag #'parteA` marca una expresión musical con el nombre *parteA*. Las expresiones etiquetadas de esta manera se pueden seleccionar o filtrar más tarde por su nombre, usando bien `\keepWithTag #'nombre` o bien `\removeWithTag #'nombre`. El resultado de la aplicación de estos filtros a la música etiquetada es como sigue:

### Filtro

Música etiquetada precedida de `\keepWithTag #'nombre` o de `\keepWithTag #'(nombre1 nombre2...)`

Música etiquetada precedida de `\removeWithTag #'nombre` o `\removeWithTag #'(nombre1 nombre2...)`

### Resultado

Se incluye la música no etiquetada y la música etiquetada con el nombre o nombres de etiqueta dados; se excluye la música etiquetada con cualquier otro nombre de etiqueta.

Se incluye la música no etiquetada y la música que no está etiquetada con ninguno de los nombres de etiqueta dados; se excluye la música etiquetada con los nombres de etiqueta dados.

Música etiquetada no precedida de Se incluye toda la música etiquetada y no `\keepWithTag` ni de `\removeWithTag` etiquetada.

Los argumentos de las instrucciones `\tag`, `\keepWithTag` y `\removeWithTag` deben ser un símbolo o lista de símbolos (como `#'score` o `#'(violinI violinII)`), seguida de una expresión musical. Si *y solo si* los símbolos son identificadores de LilyPond válidos (caracteres alfabéticos solamente, sin números, guiones bajos ni guiones normales) que no puedan confundirse con notas, la porción `#'` se puede omitir y, como abreviatura, una lista de símbolos puede usar el separador de punto, es decir: `\tag #'(violinI violinII)` se puede escribir como `\tag violinI.violinII`. Lo mismo vale para `\keepWithTag` y `\removeWithTag`.

En el ejemplo siguiente vemos dos versiones de una pieza musical, una que muestra trinos con la notación usual y otra con los trinos expandidos explícitamente:

```
music = \relative {
  g'8. c32 d
  \tag #'trills { d8.\trill }
  \tag #'expand { \repeat unfold 3 { e32 d } }
  c32 d
}

\score {
  \keepWithTag #'trills \music
}
\score {
  \keepWithTag #'expand \music
}
```

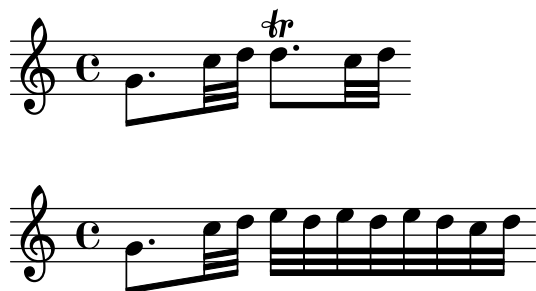


De forma alternativa, a veces es más fácil excluir secciones de música:

```
music = \relative {
  g'8. c32 d
  \tag #'trills { d8.\trill }
  \tag #'expand { \repeat unfold 3 { e32 d } }
  c32 d
}

\score {
  \removeWithTag #'expand
  \music
}
\score {
  \removeWithTag #'trills
  \music
}
```

}



El filtrado con etiquetas se puede aplicar a articulaciones, textos, etc. anteponiendo

`-\tag #'mi-etiqueta`

a una articulación. Por ejemplo, esto define una nota con una indicación opcional de digitación y una nota con una anotación condicional:

```
c1-\tag #'finger ^4
c1-\tag #'warn ^"¡Ojo!"
```

Se pueden poner varias etiquetas sobre expresiones con varias entradas de `\tag`, o combinando varias etiquetas en una lista de símbolos:

```
music = \relative c'' {
  \tag #'a \tag #'both { a4 a a a }
  \tag #'(b both) { b4 b b b }
}
<<
\keepWithTag #'a \music
\keepWithTag #'b \music
\keepWithTag #'both \music
>>
```



Se pueden aplicar varios filtros `\removeWithTag` a una sola expresión musical para quitar varias secciones etiquetadas con nombres distintos. De forma alternativa, podemos usar una sola instrucción `\removeWithTag` con una lista de etiquetas.

```
music = \relative c'' {
  \tag #'A { a4 a a a }
  \tag #'B { b4 b b b }
  \tag #'C { c4 c c c }
  \tag #'D { d4 d d d }
}
\new Voice {
  \removeWithTag #'B
```

```

\removeWithTag #'C
\music
\removeWithTag #'(B C)
\music
}

```



El uso de dos o más filtros `\keepWithTag` sobre una sola expresión musical produce la eliminación de *todas* las secciones etiquetadas. El primer filtro quita todas excepto la que se nombra, y cualquier filtro posterior elimina el resto. Usando una sola instrucción `\keepWithTag` con una lista de varias etiquetas se suprimirán solamente las secciones etiquetadas que no aparezcan especificadas en la lista.

```

music = \relative c'' {
  \tag #'violinI { a4 a a a }
  \tag #'violinII { b4 b b b }
  \tag #'viola { c4 c c c }
  \tag #'cello { d4 d d d }
}

\new Staff {
  \keepWithTag #'(violinI violinII)
  \music
}

```



will print `\tags violinI` and `violinII` but not `viola` or `cello`.

Aunque `\keepWithTag` es útil para el manejo de *un solo* conjunto de alternativas, la eliminación de música etiquetada con otras etiquetas *no relacionadas* es problemática si se utilizan para más de un propósito. En ese caso pueden declararse ‘grupos de etiquetas’:

```
\tagGroup #'(violinI violinII viola cello)
```

Ahora todas las etiquetas pertenecen a un solo ‘grupo de etiquetas’. Observe que las etiquetas individuales no pueden ser miembros de más de un *grupo de etiquetas*.

```
\keepWithTag #'violinI ...
```

mostrará solamente la música etiquetada a partir del grupo de etiquetas de `violinI` y cualquier fragmento musical etiquetado con una de las *otras* etiquetas resultará suprimido.

```

music = \relative {
  \tagGroup #'(violinI violinII viola cello)
  \tag #'violinI { c''4~"violinI" c c c }
  \tag #'violinII { a2 a }
  \tag #'viola { e8 e e2. }
  \tag #'cello { d'2 d4 d }
  R1~"untagged"
}

\new Voice {

```

```
\keepWithTag #'violinI
\music
}
```



Al usar la instrucción `\keepWithTag` solo son visibles las etiquetas de los grupos de aquellas etiquetas se se incluyen en la instrucción.

A veces podemos desear dividir la música en un determinado lugar dentro de una expresión musical existente. Podemos usar `\pushToTag` y `\appendToTag` para añadir material delante o al final de los `elements` de una construcción musical existente. No toda construcción musical tiene la parte `elements`, pero las músicas secuenciales y simultáneas son dos apuestas seguras:

```
music = { \tag #'here { \tag #'here <<c''>> } }
```

```
{
  \pushToTag #'here c'
  \pushToTag #'here e'
  \pushToTag #'here g' \music
  \appendToTag #'here c'
  \appendToTag #'here e'
  \appendToTag #'here g' \music
}
```



Las dos instrucciones admiten una etiqueta, el material que dividir en cada ocurrencia de la etiqueta, y la expresión etiquetada.

## Véase también

Manual de aprendizaje: Sección “Organizar las piezas mediante variables” en *Manual de Aprendizaje*.

Referencia de la notación: [Combinación automática de las partes], página 192, Sección 3.3.1 [Inclusión de archivos de LilyPond], página 538.

## Advertencias y problemas conocidos

Al llamar a `\relative` sobre una expresión musical obtenida mediante el filtrado de música con las instrucciones `\keepWithTag` o `\removeWithTag`, podría ocurrir que cambiaran las relaciones de octava, dado que solo se tendrán en cuenta las alturas de las notas que efectivamente permanecen en la expresión filtrada. La aplicación de `\relative` en primer lugar, antes de `\keepWithTag` o de `\removeWithTag`, evita este peligro porque en este caso `\relative` actúa sobre todas las notas tal y como se introducen.

## Uso de ajustes globales

Se pueden incluir ajustes globales a partir de un archivo distinto:

```
lilypond -dinclude-settings=MIS_AJUSTES.ly MI_PARTITURA.ly
```

En archivos diferentes se pueden almacenar grupos de ajustes como el tamaño de la página o las fuentes tipográficas. Ello permite hacer diferentes ediciones de la misma partitura así como

aplicar ajustes estándar a muchas partituras, simplemente por medio de la especificación del archivo de ajustes adecuado.

Esta técnica también funciona bien con el uso de hojas de estilo, como se estudia en Sección “Hojas de estilo” en *Manual de Aprendizaje*.

## Véase también

Manual de aprendizaje: Sección “Organizar las piezas mediante variables” en *Manual de Aprendizaje*, Sección “Hojas de estilo” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 3.3.1 [Inclusión de archivos de LilyPond], página 538.

### 3.3.3 Caracteres especiales

#### Codificación del texto

LilyPond usa el conjunto de caracteres definido por el consorcio Unicode y la norma ISO/IEC 10646. Define un nombre único y un código para los conjuntos de caracteres que se utilizan en prácticamente todos los idiomas modernos y también en muchos otros. Unicode se puede implementar utilizando varios esquemas de codificación distintos. LilyPond usa la codificación UTF-8 (UTF son las siglas de *Unicode Transformation Format*, o formato de transformación de Unicode) que representa todos los caracteres comunes de la codificación Latin en un solo byte, y representa otros caracteres usando un formato de longitud variable de hasta cuatro bytes.

El aspecto visual real de los caracteres viene determinado por los glifos que se definen en las fuentes tipográficas concretas que se tengan disponibles: una fuente tipográfica define la asignación de un subconjunto de los códigos de Unicode a glifos. LilyPond usa la biblioteca Pango para representar y disponer tipográficamente textos multilingües.

LilyPond no realiza ninguna conversión en la codificación de la entrada. Esto significa que cualquier text, ya sea el título, la letra de la canción o una instrucción musical que contenga caracteres distintos a los del conjunto ASCII, se deben codificar en UTF-8. La forma más fácil de escribir dicho texto es utilizar un editor preparado para Unicode y guardar el archivo con la codificación UTF-8. Casi todos los editores modernos populares contemplan el UTF-8, por ejemplo lo hacen vim, Emacs, jEdit y Gedit. Todos los sistemas MS Windows posteriores a NT usan Unicode como codificación de caracteres nativa, de manera que incluso el accesorio Bloc de Notas (Notepad) puede editar y guardar un archivo en el formato UTF-8. Una alternativa más funcional para Windows es BabelPad.

Si un archivo de entrada que contiene un carácter que no es ASCII, no se guarda en el formato UTF-8, se genera el mensaje de error

```
FT_Get_Glyph_Name () error: argumento inválido
```

He aquí un ejemplo que muestra texto cirílico, hebreo y portugués:



## Unicode

Para introducir un carácter aislado para el que se conoce el punto de código Unicode pero no está disponible en el editor que se está usando, use `\char ##xhhh` o bien `\char #dddd` dentro de un bloque `\markup`, donde `hhh` es el código hexadecimal del carácter en cuestión y `dddd`

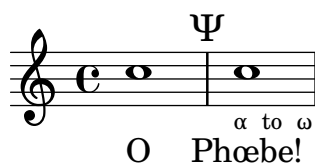
es su valor decimal correspondiente. Pueden omitirse los ceros iniciales, pero es costumbre indicar los cuatro caracteres en la representación hexadecimal (observe que *no* debe utilizarse la codificación UTF-8 del punto de código Unicode después de `\char`, pues la codificación UTF-8 contiene bits adicionales que indican el número de octetos). Hay tablas de códigos Unicode y un índice de nombres de caracteres que da el punto de código en hexadecimal para cualquier carácter en el portal del Consorcio Unicode, <http://www.unicode.org/>.

Por ejemplo, tanto `\char ##x03BE` como `\char #958` insertan el carácter Unicode U+03BE, que tiene el nombre Unicode “Letra griega Xi pequeña”.

Se puede escribir de esta forma cualquier punto de código Unicode, y si todos los caracteres especiales se escriben en este formato no es necesario guardar el archivo de entrada en formato UTF-8. Por supuesto, debe estar instalada y estar disponible para LilyPond una fuente tipográfica que contenga codificados todos estos caracteres.

El ejemplo siguiente muestra valores hexadecimales Unicode que se usan en cuatro lugares: en una llamada de ensayo, como texto de articulación, en la letra y como texto independiente bajo la partitura:

```
\score {
  \relative {
    c'1 \markup { \markup { \char ##x03A8 }
    c1_\markup { \tiny { \char ##x03B1 " to " \char ##x03C9 } }
  }
  \addlyrics { 0 \markup { \concat { Ph \char ##x0153 be! } } }
}
\markup { "Copyright 2008--2021" \char ##x00A9 }
```



Copyright 2008--2021 ©

Para escribir el símbolo de copyright en la nota de créditos, utilice:

```
\header {
  copyright = \markup { \char ##x00A9 "2008" }
}
```

## Alias de ASCII

Se puede incluir una lista de alias de ASCII para caracteres especiales:

```
\paper {
  #(include-special-characters)
}

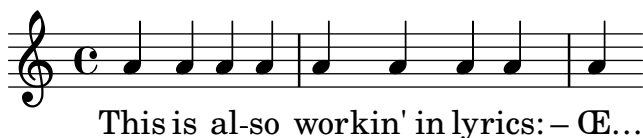
\markup "&flqq; &ndash; &OE;uvre incomplète&hellip; &frqq;"

\score {
  \new Staff { \repeat unfold 9 a'4 }
  \addlyrics {
    This is al -- so wor -- kin'~in ly -- rics: &ndash;_&OE;&hellip;
  }
}
```



```
\markup \column {
  "The replacement can be disabled:"
  "&ndash; &OE; &hellip;"
  \override #'(replacement-alist . ()) "&ndash; &OE; &hellip;"
}
```

« – Œuvre incomplète... »



The replacement can be disabled:

– Œ ...

&ndash; &OE; &hellip;

También podemos hacer nuestros propios alias, ya sea de forma global,

```
\paper {
  #(add-text-replacements!
    '(("100" . "hundred")
      ("dpi" . "dots per inch")))
}
\markup "A 100 dpi."
```

A hundred dots per inch.

o local:

```
\markup \replace #'(("100" . "hundred")
  ("dpi" . "dots per inch")) "A 100 dpi."
```

A hundred dots per inch.

## Véase también

Referencia de la notación: Sección A.13 [Lista de caracteres especiales], página 798.

Archivos instalados: `ly/text-replacements.ly`.

## 3.4 Controlar la salida

### 3.4.1 Extracción de fragmentos de música

Es posible producir en la salida uno o más fragmentos de una partitura mediante la definición de la situación explícita de la música que se quiere extraer dentro del bloque `\layout` del archivo de entrada, utilizando la función `clip-regions`, y a continuación ejecutando LilyPond con la opción `-dclip-systems`.

```
\layout {
  clip-regions
  = #(list
    (cons
      (make-rhythmic-location 5 1 2)
```

```
(make-rhythmic-location 7 3 4)))
}
```

Este ejemplo extrae un solo fragmento del archivo de entrada *comenzando* después de la duración de una blanca en el quinto compás (5 1 2) y *terminando* después de la tercera negra del compás séptimo (7 3 4).

Se pueden extraer fragmentos adicionales añadiendo más pares de entradas de `make-rhythmic-location` a la lista `clip-regions` dentro del bloque `\layout`.

De forma predeterminada, cada fragmento de música se obtendrá en la salida como un archivo EPS distinto, pero también se pueden crear otros formatos como PDF o PNG si es necesario. La música extraída se obtiene en la salida como si hubiera sido literalmente ‘cortada’ de la partitura original impresa, de forma que si un fragmento se extiende sobre una o más líneas, se genera un archivo de salida distinto por cada línea generada.

## Véase también

Referencia de la notación: Sección 4.2.1 [El bloque `\layout`], página 579.

Manual de utilización: Sección “Utilización desde la línea de órdenes” en *Utilización del Programa*.

### 3.4.2 Saltar la música corregida

Al escribir o copiar música, normalmente sólo es interesante de ver y corregir la música cercana al final (donde estamos añadiendo las notas). Para acelerar este proceso de corrección, es posible saltar la composición tipográfica de todos excepto unos pocos de los últimos compases. Esto se consigue definiendo una variable especial al principio del archivo fuente, como sigue:

```
showLastLength = R1*5
\score { ... }
```

En este ejemplo no se dibujará nada excepto los últimos cinco compases (suponiendo un compás de 4/4) de cada una de las partituras `\score` del archivo de entrada. Para piezas largas, el tipografiado de únicamente una parte pequeña es con frecuencia un orden de magnitud más rápido que el de la obra completa. Si estamos trabajando sobre el principio de una partitura que ya hemos tipografiado (p.ej., para añadir una parte instrumental nueva), también puede ser útil la propiedad `showFirstLength`.

Este mecanismo de pasar por alto partes de una partitura se puede controlar con un grano más fino mediante la propiedad `Score.skipTypesetting`. Cuando su valor está establecido, no se lleva a cabo ningún tipografiado en absoluto. Como propiedad del contexto `Score`, afecta a todas las voces y pentagramas; véase [Score. El contexto maestro], página 623.

Esta propiedad se usa también para controlar la salida hacia el archivo MIDI. Si algún evento dentro de la sección que se salta altera alguna de sus propiedades de contexto, por ejemplo una indicación de tempo o un cambio de instrumento, entonces el ajuste tendrá efecto solamente en el punto temporal en que `skipTypesetting` se desactiva de nuevo:

```
\relative c' {
  c4 c c c
  \set Score.skipTypesetting = ##t
  d4 d d d
  \tempo 4 = 80
  e4 e e e
  \set Score.skipTypesetting = ##f
  f4 f f f
```

}



## Instrucciones predefinidas

`showLastLength`, `showFirstLength`.

## Véase también

Referencia de la notación: Sección 5.1 [Contextos de interpretación], página 622, [Score. El contexto maestro], página 623, Sección A.17 [Todas las propiedades de contexto], página 808.

### 3.4.3 Formatos de salida alternativos

Los formatos de salida predeterminados para la partitura impresa son el formato de documento portátil (PDF) y PostScript (PS). Los formatos de salida Gráficos de red portátiles (PNG), Gráficos de vector escalables (SVG) y PostScript encapsulado (EPS) también están disponibles a través de opciones de la línea de órdenes, véase Sección “Opciones básicas de la línea de órdenes para LilyPond” en *Utilización del Programa*.

## Salida de SVG

La salida de SVG puede contener, opcionalmente, metadatos para objetos gráficos (grobs) como cabezas de nota, silencios, etc. Estos metadatos pueden ser atributos SVG estándar como `id` y `class`, o atributos personalizados no estándar. Especifique los atributos y sus valores mediante la sobrescritura de la propiedad de grob `output-attributes` con una lista de asociación de Scheme (alist). Los valores pueden ser números, cadenas de caracteres o símbolos. Por ejemplo:

```
{
  \once \override NoteHead.output-attributes =
    #'((id . 123)
      (class . "esto aquello")
      (data-lo-que-sea . algo))
  c
}
```

El código de entrada anterior produce la siguiente etiqueta `<g>` (grupo) dentro del archivo SVG:

```
<g id="123" class="esto aquello" data-lo-que-sea="algo">
  ...elementos del SVG de objetos gráficos como cabezas de nota...
</g>
```

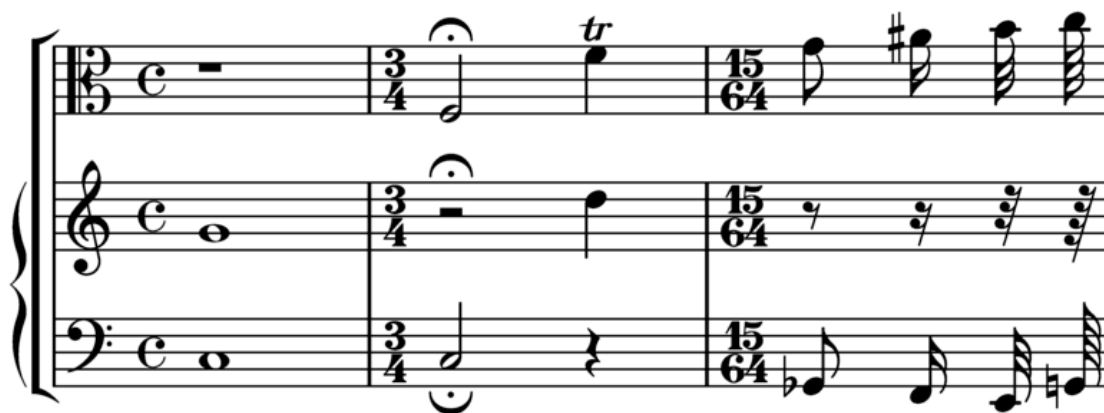
La etiqueta `<g>` contiene todos los elementos de SVG para un grob dado (ciertos grobs generan más de un elemento del SVG). En la sintaxis del SVG el prefijo `data-` se usa para atributos de medios personalizados no estándar.

### 3.4.4 Sustituir la tipografía de la notación

Gonville es un conjunto de glifos alternativo a la tipografía Feta (que es parte de la tipografía Emmentaler que se usa en LilyPond). Se puede descargar de:

<http://www.chiark.greenend.org.uk/~sgtatham/gonville/> (<http://www.chiark.greenend.org.uk/~sgtatham/gonville/> )

Aquí presentamos algunos compases de muestra tipografiados con la fuente Gonville:



Aquí hay unos compases de muestra tipografiados con los glifos Feta de LilyPond:



## Instrucciones de instalación

Descargue y extraiga los archivos de fuente tipográfica. Copie los archivos `gonville-11.otf`, `gonville-13.otf`, `gonville-14.otf`, `gonville-16.otf`, `gonville-18.otf`, `gonville-20.otf`, `gonville-23.otf`, `gonville-26.otf` y `gonville-brace.otf` en la carpeta `.../share/lilypond/current/fonts/otf` o `.../share/lilypond/X.Y.Z/fonts/otf`. Si tiene los archivos `gonville-*.svg` y `gonville-*.woff`, cópielos a `.../share/lilypond/current/fonts/svg` o a `.../share/lilypond/X.Y.Z/fonts/svg`. Para más información consulte Sección “Otras fuentes de información” en *Manual de Aprendizaje*.

Nota: los archivos `gonville-*.otf` son para los *backends* `ps` y `eps` (para las salidas en formato PDF y PostScript). Los archivos `gonville-*.svg` son para el *backend* `svg` sin la opción `svg-woff`. Los archivos `gonville-*.woff` son para el *backend* `svg` con la opción `svg-woff`. Para más información, consulte Sección “Opciones avanzadas de línea de órdenes para LilyPond” en *Utilización del Programa*.

La sintaxis siguiente cambia la fuente tipográfica de la notación (general y llaves) al tipo Gonville.

```
\paper {
  #(define fonts
    (set-global-fonts
      #:music "gonville")
```

```

    #:brace "gonville"
  ))
}

```

Nota: cada llamada a `set-global-fonts` reinicia completamente tanto la fuente principal de notación como la de texto. Si se deja sin especificar una cualquiera de las categorías, entonces se utiliza la tipografía predeterminada para esa categoría. Cada llamada a `set-global-fonts` cambia las fuentes para cada `\book` que le sigue, tanto si se ha creado explícitamente como implícitamente. Esto significa que cada `\book` puede tener su propio conjunto de fuentes principales llamando a `set-global-fonts` previamente. Para más información, véase [Fuentes tipográficas del documento completo], página 284.

## Véase también

Manual de aprendizaje: Sección “Otras fuentes de información” en *Manual de Aprendizaje*.

Referencia de la notación: Sección A.8 [La tipografía Emmentaler], página 713, `\undefined` [`\undefined`], página `\undefined`.

## Advertencias y problemas conocidos

Gonville no se puede usar para tipografiar notación de ‘Música Antigua’ y es probable que cualquier nuevo glifo en versiones posteriores de LilyPond no existan en la familia tipográfica Gonville. Diríjase a la página web del autor para obtener más información sobre éste y otros asuntos, entre ellos el régimen de licencias de Gonville.

## Otras fuentes de notación

Si tiene otras fuentes de notación como `fontname-*.otf`, `fontname-*.svg` y `fontname-*.woff`, puede usarlas de la misma manera que Gonville.

Esto es, copie los archivos `fontname-*.otf` a la carpeta `.../share/lilypond/current/fonts/otf` o a `.../share/lilypond/X.Y.Z/fonts/otf`. Si tiene los archivos `fontname-*.svg` y `fontname-*.woff`, cópielos a `.../share/lilypond/current/fonts/svg` o a `.../share/lilypond/X.Y.Z/fonts/svg`.

Nota: por el momento, LilyPond espera que los nombres de archivo tengan los siguientes sufijos, todos los cuales deben estar presentes en las carpetas de instalación arriba mencionadas para que funcionen adecuadamente: `-11`, `-13`, `-14`, `-16`, `-18`, `-20`, `-23`, `-26`, `-brace`. Por ejemplo, `emmentaler-11.otf`, `emmentaler-20.svg` y `emmentaler-brace.woff`, etc.

La sintaxis siguiente cambia la fuente tipográfica de la notación (general y de llaves) a la fuente *tipoletra*.

```

\paper {
  #(define fonts
    (set-global-fonts
      #:music "tipoletra" ; nombre del archivo de fuente sin el sufijo ni la extensión
      #:brace "tipoletra" ; nombre del archivo de fuente sin el sufijo ni la extensión
    ))
}

```

Nota: para las categorías `music` y `brace`, especifique el nombre del archivo de fuente sin el sufijo ni la extensión.

## 3.5 Creación de salida MIDI

LilyPond puede producir archivos que siguen el estándar MIDI (Musical Instrument Digital Interface; interfaz digital para instrumentos musicales) y permitir así comprobar la música por el oído (con la ayuda de alguna aplicación o dispositivo que entienda el MIDI). La escucha de

la salida MIDI también puede ser de ayuda en la localización de errores tales como notas que se han introducido incorrectamente, alteraciones omitidas y otros casos por el estilo.

Los archivos MIDI no contienen sonido (como los archivos AAC, MP3 o Vorbis) pero requieren software adicional para producir el sonido a partir de ellos.

### 3.5.1 Notación contemplada por el MIDI

Se puede usar la notación musical siguiente con las posibilidades predeterminadas de LilyPond para producir salida MIDI:

- Marcas de respiración
- Acordes introducidos como nombres de acorde
- Crescendos y decrescendos sobre más de una nota. El volumen se altera de forma lineal entre los dos extremos
- Indicadores dinámicos desde **ppppp** hasta **fffff**, incluyendo **mp**, **mf** y **sf**
- Microtonos, pero *no* acordes microtonales. Se necesita también un reproductor de MIDI capaz de ejecutar eventos de rueda de tono.
- Letra de las canciones
- Altura de las notas
- Ritmo introducido como duraciones de las notas, incluidos los grupos de valoración especial como los tresillos
- Articulaciones ‘sencillas’; picado, staccato, acento, marcato y portato
- Cambios de tempo introducidos con la función `\tempo`
- Ligaduras de unión
- Trémolos que *not* se hayan introducido como un valor del tipo ‘:[*número*]

También se pueden controlar el panorama, el balance, la expresión y los efectos de reverberación y chorus mediante el establecimiento de propiedades de contexto, véase Sección 3.5.8 [Propiedades de contexto para efectos MIDI], página 563.

Si se combina con el script `articulate`, es posible producir la salida por el MIDI de los siguientes elementos adicionales de notación musical:

- Appoggiaturas (apoyaturas). Se les da la mitad de la duración de la nota siguiente (sin tener en cuenta los puntillos). Por ejemplo:

```
\appoggiatura c8 d2.
```

El Do toma la duración de una negra.

- Notas y grupos de adorno (mordentes de una nota, trinos, grupetos circulares, etc.)
- Rallentando, acelerando, ritardando y a tempo
- Ligaduras de expresión y de fraseo
- Tenuto

Véase Sección 3.5.9 [Enriquecimiento de la salida MIDI], página 564.

### 3.5.2 Notación no contemplada en el MIDI

Los siguientes elementos de notación musical no se pueden hacer salir por el MIDI:

- Articulaciones distintas al picado, staccato, acento, marcato y portato
- Crescendos y decrescendos sobre una *sola* nota
- Calderón
- Bajo cifrado
- Glissandos

- Caídas y elevaciones
- Acordes microtonales
- Duraciones introducidas como anotaciones, p.ej. el swing
- Cambios de tempo sin `\tempo` (p.ej., introducidas como anotaciones)
- Trémolos que *se introducen* con la notación de la forma ‘:[*número*]’

### 3.5.3 El bloque MIDI

Para crear un archivo de salida MIDI a partir de un archivo de entrada de LilyPond, introduzca un bloque `\midi`, que puede estar vacío, dentro del bloque `\score`:

```
\score {
  ... música ...
  \layout { }
  \midi { }
}
```

**Nota:** Un bloque `\score` que, además de la música, contiene solamente un bloque `\midi` (o sea, *sin* el bloque `\layout`), solo produce archivos de salida MIDI; no se imprime ninguna notación musical.

La extensión predeterminada para los archivos de salida (`.midi`) se puede cambiar usando la opción `-dmidi-extension` con la instrucción `lilypond`:

```
lilypond -dmidi-extension=mid Archivo.ly
```

Como alternativa, puede añadirse la siguiente expresión de Scheme antes del comienzo de uno de los bloques `\book`, `\bookpart` o `\score`. Véase Sección 3.1.5 [Estructura del archivo], página 510.

```
 #(ly:set-option 'midi-extension "mid")
```

### Véase también

Referencia de la notación: Sección 3.1.5 [Estructura del archivo], página 510, Sección 3.2.3 [Crear metadatos en los archivos de salida], página 525.

Archivos instalados: `scm/midi.scm`.

### Advertencias y problemas conocidos

Están disponibles 15 canales MIDI y un canal adicional (el número 10) para la percusión. Los pentagramas se asignan a los canales por orden, de forma que una partitura que contenga más de 15 pentagramas da como resultado que los pentagramas adicionales comparten (pero no sobrescriben) el mismo canal MIDI. Esto puede ser problemático si los pentagramas que comparten canal tienen establecidas propiedades MIDI, basadas en canal, en mutuo conflicto (como distintos instrumentos MIDI).

Using a `mid` block with polymetric notation may cause unexpected barcheck warnings. In this case move the `Timing_translator` from the `Score` context to the `Staff` context within the `mid` block.

```
\midi {
  \context {
    \Score
    \remove "Timing_translator"
  }
  \context {
    \Staff
```

```

    \consists "Timing_translator"
  }
}

```

### 3.5.4 Control de las dinámicas del MIDI

Es posible controlar el volumen MIDI general, el volumen relativo de las indicaciones de matiz dinámico y el volumen relativo de los distintos instrumentos.

Las indicaciones matiz dinámico se traducen automáticamente a niveles de volumen dentro del rango de volúmenes MIDI disponible mientras que los crescendos y decrescendos varían el volumen linealmente entre sus dos extremos. Es posible controlar el volumen relativo de las indicaciones de matiz dinámico, y el volumen general de los diferentes instrumentos.

### Indicaciones de matiz dinámico en el MIDI

Solo las marcas de dinámica desde *ppppp* hasta *fffff*, incluidos *mp*, *mf* y *sf* tienen valores asignados. Este valor se aplica a continuación al valor del rango de volumen MIDI general para obtener el volumen final que se incluye en la salida MIDI para esa indicación de matiz dinámico en particular. Las fracciones predeterminadas varían desde 0.25 para el *ppppp* hasta 0.95 para el *fffff*. El conjunto completo de las indicaciones de matiz dinámico pueden verse en el archivo `scm/midi.scm`.

### Fragmentos de código seleccionados

#### *Creating custom dynamics in MIDI output*

The following example shows how to create a dynamic marking, not included in the default list, and assign it a specific value so that it can be used to affect MIDI output.

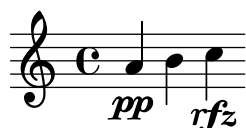
The dynamic mark `\rfz` is assigned a value of 0.9.

```

#(define (myDynamics dynamic)
  (if (equal? dynamic "rfz")
      0.9
      (default-dynamic-absolute-volume dynamic)))

\score {
  \new Staff {
    \set Staff.midiInstrument = #"cello"
    \set Score.dynamicAbsoluteVolumeFunction = #myDynamics
    \new Voice {
      \relative {
        a'4\pp b c-\rfz
      }
    }
  }
  \layout {}
  \midi {}
}

```



Archivos instalados: `ly/script-init.ly` `scm/midi.scm`.



Fragmentos de código: Sección “MIDI” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Dynamic\_performer” en *Referencia de Funcionamiento Interno*.

## Establecimiento del volumen MIDI

Los volúmenes generales mínimo y máximo de las indicaciones de matiz dinámico MIDI se controlan estableciendo las propiedades `midiMinimumVolume` y `midiMaximumVolume` en el nivel del bloque `Score`. Estas propiedades tienen efecto solamente al comienzo de una voz y sobre las indicaciones dinámicas. La fracción que corresponde a cada indicación dinámica se modifica con esta fórmula:

$$\text{midiMinimumVolume} + (\text{midiMaximumVolume} - \text{midiMinimumVolume}) * \text{fracción}$$

En el ejemplo siguiente, el rango dinámico del volumen MIDI general se limita al intervalo entre 0.2 y 0.5.

```
\score {
  <<
    \new Staff {
      \set Staff.midiInstrument = "flute"
      ... music ...
    }
    \new Staff {
      \set Staff.midiInstrument = "clarinet"
      ... music ...
    }
  >>
  \midi {
    \context {
      \Score
      midiMinimumVolume = #0.2
      midiMaximumVolume = #0.5
    }
  }
}
```

Se puede obtener una ecualización sencilla de instrumentos MIDI estableciendo las propiedades `midiMinimumVolume` y `midiMaximumVolume` dentro del contexto `Staff`.

```
\score {
  \new Staff {
    \set Staff.midiInstrument = "flute"
    \set Staff.midiMinimumVolume = #0.7
    \set Staff.midiMaximumVolume = #0.9
    ... music ...
  }
  \midi { }
}
```

Para partituras con más de un pentagrama y más de un instrumento MIDI, se pueden fijar individualmente los volúmenes relativos de cada instrumento:

```
\score {
  <<
    \new Staff {
      \set Staff.midiInstrument = "flute"
      \set Staff.midiMinimumVolume = #0.7
```

```

    \set Staff.midiMaximumVolume = #0.9
    ... music ...
  }
  \new Staff {
    \set Staff.midiInstrument = "clarinet"
    \set Staff.midiMinimumVolume = #0.3
    \set Staff.midiMaximumVolume = #0.6
    ... music ...
  }
>>
\midi { }
}

```

En este ejemplo el volumen del clarinete se reduce de forma relativa al volumen de la flauta.

Si no se establecen estas propiedades de volumen, LilyPond aún aplica un ‘pequeño grado’ de ecualización a ciertos instrumentos. Véase `scm/midi.scm`.

Archivos instalados: `scm/midi.scm`.

## Véase también

Referencia de la notación: Sección 4.2 [Disposición de la partitura], página 579.

Referencia de funcionamiento interno: Sección “Dynamic-performer” en *Referencia de Funcionamiento Interno*.

## Fragmentos de código seleccionados

### *Replacing default MIDI instrument equalization*

The default MIDI instrument equalizer can be replaced by setting the `instrumentEqualizer` property in the `Score` context to a user-defined Scheme procedure that uses a MIDI instrument name as its argument along with a pair of fractions indicating the minimum and maximum volumes respectively to be applied to that specific instrument.

The following example sets the minimum and maximum volumes for flute and clarinet respectively.

```

#(define my-instrument-equalizer-alist '())

#(set! my-instrument-equalizer-alist
  (append
    '(
      ("flute" . (0.7 . 0.9))
      ("clarinet" . (0.3 . 0.6)))
    my-instrument-equalizer-alist))

#(define (my-instrument-equalizer s)
  (let ((entry (assoc s my-instrument-equalizer-alist)))
    (if entry
      (cdr entry))))

\score {
  <<
    \new Staff {
      \key g \major
      \time 2/2

```

```

\set Score.instrumentEqualizer = #my-instrument-equalizer
\set Staff.midiInstrument = "flute"
\new Voice \relative {
  r2 g''\mp g fis~
  4 g8 fis e2~
  4 d8 cis d2
}
}
\new Staff {
  \key g \major
  \set Staff.midiInstrument = "clarinet"
  \new Voice \relative {
    b'1\p a2. b8 a
    g2. fis8 e
    fis2 r
  }
}
>>
\layout { }
\midi { }
}

```



## Advertencias y problemas conocidos

Los cambios en el volumen del MIDI solamente tienen lugar en el comienzo de una nota, por lo que los crescendos y decrescendos no afectan al volumen de una sola nota.

## Establecimiento de las propiedades del bloque MIDI

El bloque `\midi` puede contener distintas disposiciones de los contextos, definiciones de contexto nuevas o código que fija los valores de ciertas propiedades.

```

\score {
  ... music ...
  \midi {
    \tempo 4 = 72
  }
}

```

Aquí, el tempo se establece a 72 pulsos de negra por minuto. La indicación de tempo dentro del bloque `\midi` no aparece en la partitura impresa. Aunque cualquier otra indicación de `\tempo` que se especifique dentro del bloque `\score` también se reflejará en la salida MIDI.

Dentro de un bloque `\midi`, la instrucción `\tempo` está estableciendo propiedades durante la interpretación de la música y en el contexto de las definiciones de salida; así, se interpreta *como si* fuera una modificación de contexto.

Las definiciones de contexto siguen la misma sintaxis que las del bloque `\layout`:

```

\score {

```

```

... music ...
\midi {
  \context {
    \Voice
    \remove "Dynamic_performer"
  }
}

```

Este ejemplo suprime el efecto de las indicaciones dinámicas sobre la salida MIDI. Nota: los módulos de traducción de LilyPond usados para el sonido se llaman ‘performers’.

## Véase también

Manual de aprendizaje: Sección “Otras fuentes de información” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 1.3 [Expresiones], página 127, Sección 4.2 [Disposición de la partitura], página 579.

Archivos instalados: `ly/performer-init.ly`.

Fragmentos de código: Sección “MIDI” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Dynamic\_performer” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Ciertos reproductores de MIDI no siempre maneja correctamente los cambios de tempo en la salida MIDI.

Los cambios efectuados al `midiInstrument`, así como ciertas opciones MIDI, al *comienzo* de un pentagrama pueden aparecer dos veces en la salida MIDI.

### 3.5.5 Uso de los instrumentos MIDI

Los instrumentos MIDI se establecen usando la propiedad `midiInstrument` dentro de un contexto `Staff`.

```

\score {
  \new Staff {
    \set Staff.midiInstrument = "glockenspiel"
    ... music ...
  }
  \midi { }
}

o

\score {
  \new Staff \with {midiInstrument = "cello"} {
    ... music ...
  }
  \midi { }
}

```

Si el instrumento elegido no coincide exactamente con uno de los instrumentos de la lista de la sección ‘instrumentos MIDI’, se usará el instrumento Piano de Cola (“*acoustic grand*”). Véase Sección A.6 [Instrumentos MIDI], página 709.

## Véase también

Manual de aprendizaje: Sección “Otras fuentes de información” en *Manual de Aprendizaje*.

Referencia de la notación: Sección A.6 [Instrumentos MIDI], página 709, Sección 4.2 [Disposición de la partitura], página 579.

Archivos instalados: `scm/midi.scm`.

## Advertencias y problemas conocidos

Los instrumentos de percusión cuya notación está en un contexto `DrumStaff` se encontrará a la salida, correctamente, sobre el canal MIDI número 10 pero ciertos instrumentos de percusión afinada como el xilófono, el vibráfono o los timbales, se tratan como instrumentos “normales”, de forma que la música de éstos se debe introducir en un contexto `Staff` (no `DrumStaff`) para obtener una salida MIDI correcta. Hay una lista completa de entradas de los **sets de percusión del canal 10** en el archivo `scm/midi.scm`. Véase Sección “Otras fuentes de información” en *Manual de Aprendizaje*.

### 3.5.6 Uso de las repeticiones con el MIDI

Las repeticiones se pueden representar en la salida MIDI aplicando la instrucción `\unfoldRepeats`.

```
\score {
  \unfoldRepeats {
    \repeat tremolo 8 { c'32 e' }
    \repeat percent 2 { c''8 d'' }
    \repeat volta 2 { c'4 d' e' f' }
    \alternative {
      { g' a' a' g' }
      { f' e' d' c' }
    }
  }
  \midi { }
}
```

Para restringir el efecto de `\unfoldRepeats` a la salida MIDI únicamente, y al mismo tiempo genera partituras imprimibles, es necesario hacer *dos* bloques `\score`: uno para el MIDI (con las repeticiones desplegadas) y otro para la notación (con repeticiones de primera y segunda vez, trémolos y del tipo porcentaje);

```
\score {
  ... music ...
  \layout { }
}
\score {
  \unfoldRepeats {
    ... music ...
  }
  \midi { }
}
```

Si se usan varias voces, cada una de ellas debe contener las repeticiones completamente desplegadas para poder obtener una salida MIDI correcta.

## Véase también

Referencia de la notación: Sección 1.4 [Repeticiones], página 158.

### 3.5.7 Asignación de canales MIDI

Cuando se está generando un archivo MIDI de una partitura, LilyPond asigna cada nota de la partitura automáticamente a un canal MIDI, aquel en que se tocaría cuando se envía hacia un dispositivo MIDI. Un canal MIDI tiene una cierta cantidad de controles disponibles para seleccionar, por ejemplo, el instrumento que usar para reproducir las notas en ese canal, u ordenar al dispositivo MIDI que aplique diversos efectos al sonido producido sobre el canal. En todo momento, cada control de un canal MIDI solo puede tener asignado un valor único (que sin embargo se puede modificar, por ejemplo, para cambiar a otro instrumento en mitad de una partitura).

El estándar MIDI contempla solamente 16 canales por dispositivo MIDI. Este límite en el número de canales limita también el número de instrumentos diferentes que pueden estar tocando al mismo tiempo.

LilyPond crea pistas MIDI separadas para cada pentagrama (o instrumento o voz, según el valor de `Score.midiChannelMapping`), y también para cada contexto de letra. No hay un límite para el número de pistas.

Para solventar el número limitado de canales MIDI, LilyPond contempla algunos modos diferentes para la asignación de canales MIDI, que se seleccionan usando la propiedad de contexto `Score.midiChannelMapping`. En cada caso, si se necesitan más canales MIDI que el límite, los números de canal asignados vuelven a contar desde cero, causando posiblemente una asignación incorrecta de instrumento a algunas notas. Esta propiedad de contexto se puede fijar a uno de los siguientes valores:

**'staff**

Reservar un canal MIDI distinto para cada uno de los pentagramas de la partitura (es el ajuste predeterminado). Todas las notas de todas las voces comparten el canal MIDI del pentagrama que las encierra, y todas se codifican dentro de la misma pista MIDI.

El límite de 16 canales se aplica al número total de contextos de pentagrama y de letra, incluso aunque las letras del MIDI no consumen un canal MIDI.

**'instrument**

Reservar un canal MIDI distinto a cada uno de los instrumentos diferentes especificados en la partitura. Esto significa que todas las notas reproducidas con el mismo instrumento MIDI comparten el mismo canal MIDI (y la misma pista), incluso aunque las notas provengan de diferentes voces o pentagramas.

En este caso los contextos de letra no cuentan para el límite de 16 canales MIDI (ya que no se asignan a un instrumento MIDI), de forma que este ajuste puede permitir una mejor distribución de los canales MIDI cuando el número de contextos de pentagrama y de letra en la partitura excede la cantidad de 16.

**'voice**

Reservar un canal MIDI distinto a cada voz de la partitura que tenga un nombre único entre las voces de su pentagrama circundante. Las voces de los distintos pentagramas siempre se asignan a distintos canales MIDI pero dos voces cualesquiera contenidas dentro del mismo pentagrama comparten el mismo canal MIDI sin tienen el mismo nombre. A causa de que `midiInstrument` y los diversos controles MIDI para los efectos son propiedades del contexto de pentagrama, no se pueden fijar independientemente para cada voz. La primera voz se reproduce con el instrumento y los efectos especificados para el pentagrama, y las voces con un nombre distinto de la primera recibirán el instrumento y los efectos predeterminados.

Nota: se pueden asignar distintos instrumentos y/o efectos a varias voces dentro del mismo pentagrama moviendo el `Staff_performer` del contexto `Staff` al contexto

Voice, y dejando que `midiChannelMapping` conserve su valor predeterminado de `'staff` o se fije al valor `'instrument`; véase el fragmento de código siguiente.

Por ejemplo, la asignación predeterminada de canal MIDI de una partitura puede cambiarse al ajuste `'instrument` como se muestra a continuación:

```
\score {
  ...music...
  \midi {
    \context {
      \Score
      midiChannelMapping = #'instrument
    }
  }
}
```

## Fragmentos de código seleccionados

### *Modificar la salida MIDI para que tenga un canal por cada voz*

Al producir una salida MIDI, el comportamiento predeterminado es que cada pentagrama representa un canal MIDI, con todas las voces de dicho pentagrama mezcladas. Esto reduce al mínimo el riesgo de que se agote el número de canales MIDI disponibles, pues existe un máximo de 16 canales por cada puerto MIDI, y la mayoría de los dispositivos sólo tiene un puerto.

Sin embargo, cuando se traslada el interpretador `Staff_performer` al contexto `Voice`, cada voz de un pentagrama puede tener su propio canal MIDI, como se muestra en el siguiente ejemplo: a pesar de estar sobre el mismo pentagrama, se crean dos canales MIDI, cada uno con un `midiInstrument` distinto.

```
\score {
  \new Staff <<
    \new Voice \relative c'' {
      \set midiInstrument = #"flute"
      \voiceOne
      \key g \major
      \time 2/2
      r2 g-"Flute" ~
      g fis ~
      fis4 g8 fis e2 ~
      e4 d8 cis d2
    }
    \new Voice \relative c'' {
      \set midiInstrument = #"clarinet"
      \voiceTwo
      b1-"Clarinet"
      a2. b8 a
      g2. fis8 e
      fis2 r
    }
  >>
  \layout { }
  \midi {
    \context {
      \Staff
      \remove "Staff_performer"
    }
  }
}
```

```

    }
    \context {
      \Voice
      \consists "Staff_performer"
    }
    \tempo 2 = 72
  }
}

```



### 3.5.8 Propiedades de contexto para efectos MIDI

Se pueden usar las siguientes propiedades de contexto para aplicar diversos efectos MIDI a las notas que se reproducen sobre el canal MIDI asociado con el pentagrama, instrumento MIDI o voz actual (dependiendo del valor de la propiedad de contexto `Score.midiChannelMapping` y del contexto en que está situado el `Staff_performer`; véase Sección 3.5.7 [Asignación de canales MIDI], página 561).

Los cambios en estas propiedades de contexto afectan a todas las notas que se reproducen sobre el canal después del cambio, aunque algunos de los efectos se pueden incluso aplicar también a notas que están ya sonando en ese momento (dependiendo de la implementación del dispositivo de salida MIDI).

Se contemplan las siguientes propiedades de contexto:

#### `Staff.midiPanPosition`

La posición de panorama controla cómo se distribuye el sonido de un canal MIDI entre las salidas estéreo izquierda y derecha. La propiedad de contexto acepta un número entre -1.0 (`#LEFT`) y 1.0 (`#RIGHT`); el valor -1.0 pone toda la potencia de sonido en la salida estéreo izquierda (dejando la salida derecha en completo silencio), el valor 0.0 (`#CENTER`) distribuye el sonido por igual entre las salidas estéreo izquierda y derecha, y el valor 1.0 mueve todo el sonido a la salida estéreo derecha. Los valores entre -1.0 y 1.0 se pueden usar para obtener distribuciones mezcladas entre las salidas estéreo izquierda y derecha.

#### `Staff.midiBalance`

Equilibrio estéreo de un canal MIDI. De forma similar a la posición de panorama esta propiedad de contexto acepta un número entre -1.0 (`#LEFT`) y 1.0 (`#RIGHT`). Varía el volumen relativo que se envía a los dos altavoces estéreo sin afectar a la distribución de las propias señales estéreo.

#### `Staff.midiExpression`

Nivel de expresión (como una fracción del nivel máximo disponible) que se aplica a un canal MIDI. Un dispositivo MIDI combina el nivel de expresión del canal MIDI con el nivel actual de dinámica de una voz (controlado mediante las instrucciones de dinámica como `\p` o `\ff`) para obtener el volumen total de cada nota dentro de la voz. El control de expresión se puede usar, por ejemplo, para implementar efectos de crescendo y decrescendo sobre notas mantenidas (algo que LilyPond no contempla de forma automática).

El rango de los niveles de expresión se extiende desde 0.0 (ninguna expresión, lo que significa un volumen nulo) hasta 1.0 (expresión al máximo).



**Staff.midiReverbLevel**

El nivel de reverberación (como una fracción del máximo nivel disponible) que se aplica a un canal MIDI. Esta propiedad acepta números entre 0.0 (sin reverberación) y 1.0 (efecto máximo).

**Staff.midiChorusLevel**

Nivel del efecto Chorus (como una fracción del máximo nivel disponible) que se aplica a un canal MIDI. Esta propiedad acepta números entre 0.0 (sin efecto de chorus) y 1.0 (efecto al máximo).

## Advertencias y problemas conocidos

Dado que los archivos MIDI no contienen en realidad ninguna información sonora, los cambios en estas propiedades de contexto solamente se convierten en peticiones para que cambien los controles de canal MIDI dentro de los archivos MIDI emitidos. Si un dispositivo MIDI determinado (como un reproductor MIDI por software) puede de hecho manejar cualquiera de estas solicitudes de un archivo MIDI, es algo que depende totalmente de la implementación del mismo: un dispositivo podría optar por ignorar algunas o todas estas solicitudes. Asimismo, la forma en que un dispositivo MIDI interpreta los diferentes valores de estos controles (generalmente, el estándar MIDI determina el comportamiento solamente en los puntos extremos del rango de valores disponibles para cada control), y el hecho de si un cambio en el valor de un control afecta a las notas que ya se están reproduciendo sobre ese canal MIDI o no, también son aspectos específicos de la implementación del dispositivo MIDI.

Al generar archivos MIDI, LilyPond sencillamente convierte de forma lineal los valores fraccionarios dentro de cada rango (7 bits, o 14 bits para los controles de canal MIDI que admiten resolución fina) en valores enteros dentro del rango correspondiente (de 0 a 127 o de 0 a 32767, respectivamente), redondeando los valores fraccionarios hacia el entero más próximo en dirección opuesta al cero. Los valores enteros convertidos se almacenan tal cual en el archivo MIDI generado. Consulte la documentación de su dispositivo MIDI para ver información acerca de cómo interpreta estos valores.

### 3.5.9 Enriquecimiento de la salida MIDI

La salida MIDI predeterminada es muy básica, pero se puede mejorar mediante la selección de los instrumentos MIDI las propiedades del bloque `\midi` y/o usando el script `articulate`.

#### 3.5.10 El script Articulate

Para utilizar el script Articulate, tenemos que incluirlo al principio de nuestro archivo de entrada:

```
\include "articulate.ly"
```

El script crea una salida MIDI en que las notas tienen una duración ajustada en el tiempo de forma que se correspondan con una serie de indicaciones de articulación y de tempo. Sin embargo, la salida tipografiada también se altera para reflejar literalmente el contenido de la salida MIDI.

```
\score {
  \articulate <<
    ... music ...
  >>
  \midi { }
}
```

La instrucción `\articulate` habilita el procesado de las abreviaturas como trinos y mordentes de varias notas. En el propio script puede verse una lista completa de los elementos que están previstos. Véase `ly/articulate.ly`.

## Véase también

Manual de aprendizaje: Sección “Otras fuentes de información” en *Manual de Aprendizaje*.

Referencia de la notación: `\displayLilyMusic` [`\displayLilyMusic`], página `\displayLilyMusic`.

Archivos instalados: `ly/articulate.ly`.

**Nota:** El script `articulate` puede acortar los acordes, lo que podría no ser adecuado para algunos tipos de instrumento, como el órgano. Las notas que no tienen ninguna articulación también se pueden acortar; así, para contener el alcance de este efecto limite el uso de de la función `\articulate` a segmentos de música más breves o modifique los valores de las variables definidas dentro del script `articulate` para compensar el efecto de acortamiento de las notas.

## 3.6 Extraer información musical

Además de crear un resultado visual y MIDI, LilyPond es capaz de presentar información musical en forma de texto.

### 3.6.1 Displaying LilyPond notation

La impresión textual de una expresión musical en notación de LilyPond puede hacerse con la función musical `\displayLilyMusic`. Para ver el resultado, lo más usual es llamar a LilyPond mediante la línea de órdenes. Por ejemplo,

```
{
  \displayLilyMusic \transpose c a, { c4 e g a bes }
}
```

imprime lo siguiente:

```
{ a,4 cis e fis g }
```

De forma predeterminada, LilyPond imprime estos mensajes en la consola junto al resto de los mensajes de la compilación de LilyPond. Para discernir entre estos mensajes y guardar el resultado de `\displayLilyMusic`, redireccione la salida hacia un archivo.

```
lilypond archivo.ly >resultado.txt
```

Observe que LilyPond no se limita a mostrar la expresión musical, sino que también la interpreta (porque `\displayLilyMusic` la devuelve además de mostrarla). Simplemente inserte `\displayLilyMusic` dentro de la música existente con el objeto de obtener información sobre ella.

Para interpretar y mostrar una sección de música en la consola, y al mismo tiempo suprimirla del archivo de salida, use la instrucción `\void`.

```
{
  \void \displayLilyMusic \transpose c a, { c4 e g a bes }
  c1
}
```

### 3.6.2 Impresión de las expresiones musicales de Scheme

Véase Sección “Presentación de las expresiones musicales” en *Extender*.

### 3.6.3 Guardar los eventos musicales en un archivo

Los eventos musicales se pueden guardar en un archivo pentagrama a pentagrama mediante la inclusión de un archivo en nuestra partitura principal.

```
\include "event-listener.ly"
```

De esta forma se crean uno o más archivos llamados `FILENAME-STAFFNAME.notes` ó `FILENAME-unnamed-staff.notes` para cada pentagrama. Observe que si tiene más de un pentagrama sin nombre, los eventos de todos los pentagramas se mezclarán entre sí dentro del mismo archivo. El resultado tiene el aspecto siguiente:

```
0.000  note      57      4  p-c 2 12
0.000  dynamic  f
0.250  note      62      4  p-c 7 12
0.500  note      66      8  p-c 9 12
0.625  note      69      8  p-c 14 12
0.750  rest      4
0.750  breathe
```

La sintaxis consiste en una línea delimitada por caracteres de tabulación, con dos campos fijos en cada línea seguidos de parámetros opcionales.

*tiempo tipo ...parámetros...*

Esta información se puede leer fácilmente por parte de otros programas como guiones de Python, y pueden ser muy útiles para aquellos investigadores que desean realizar un análisis musical o hacer experimentos de reproducción con LilyPond.

## Advertencias y problemas conocidos

No todos los eventos musicales de lilypond están contemplados por `event-listener.ly`. Se pretende que sea una “prueba de concepto” bien realizada. Si algunos eventos que quiere ver no aparecen incluidos, copie `event-listener.ly` en su carpeta de lilypond y modifique el archivo de forma que produzca la información que desea.

## 4 Problemas de espaciado

La disposición global sobre el papel viene determinada por tres factores: el diseño de página, los saltos de línea y el espaciado. Estos factores influyen entre sí. La elección de uno u otro espaciado determina qué densidad termina teniendo cada sistema de música. Esto influye en el lugar en que se eligen los saltos de línea y, en último término, en el número de páginas que tiene una partitura.

Considerado globalmente, este proceso se produce en cuatro fases: en primer lugar se eligen unas distancias elásticas o ‘muelles’, basados en la duración de las figuras. Se prueban todas las combinaciones de saltos de línea posibles, y para cada una de ellas se calcula una puntuación de ‘maldad’. Después se hace una estimación de la altura de cada uno de los posibles sistemas. Finalmente se selecciona una combinación de saltos de página y de línea de tal forma que ni el espaciado horizontal ni el vertical estén demasiado estrechos ni estirados.

Hay dos tipos de bloques que pueden contener ajustes de disposición: `\paper {...}` y `\layout {...}`. El bloque `\paper` contiene ajustes de disposición de la página que se espera sean los mismos para todas las partituras de un libro o parte de libro, como la altura de la hoja o si se imprimen los números de página, etc. Véase Sección 4.1 [Disposición de la página], página 567. El bloque `\layout` contiene ajustes de disposición de la partitura, como el número de sistemas que utilizar, o la separación entre grupos de pentagramas, etc. Véase Sección 4.2 [Disposición de la partitura], página 579.

### 4.1 Disposición de la página

Esta sección estudia las opciones de disposición de la página para el bloque `\paper`.

#### 4.1.1 El bloque `\paper`

Los bloques `\paper` pueden aparecer en tres lugares diferentes para formar una jerarquía descendente de bloques `\paper`:

- Al principio del archivo de entrada, antes de cualquier bloque `\book`, `\bookpart`, o `\score`.
- Dentro de un bloque `\book` pero fuera de cualquier bloque `\bookpart` o `\score` que estén dentro de ese libro.
- Dentro de un bloque `\bookpart` pero fuera de cualquier bloque `\score` que esté dentro de esa parte de libro.

No se puede poner un bloque `\paper` dentro de un bloque `\score`.

Los valores de los campos filtran esta jerarquía, persistiendo los valores establecidos más arriba en la jerarquía a no ser que se sobrescriban por algún valor establecido más abajo en la jerarquía.

Pueden aparecer varios bloques `\paper` en cada uno de los niveles, por ejemplo como parte de distintos archivos incluidos con `\include`. Si es el caso, los campos en cada nivel se combinan, tomando precedencia los campos que aparecen más tarde si se producen campos duplicados.

Entre los ajustes que pueden aparecer dentro de un bloque `\paper` están los siguientes:

- la función de Scheme `set-paper-size`,
- variables de `\paper` utilizadas para personalizar la disposición de la página, y
- definiciones de marcado utilizadas para personalizar la disposición de los encabezamientos, pies y títulos.

La función `set-paper-size` se estudia en la sección siguiente, Sección 4.1.2 [Tamaño del papel y escalado automático], página 568. Las variables de `\paper` que se ocupan de la disposición de la página se estudian en secciones posteriores. Las definiciones de elementos de marcado

que se encargan de los encabezamientos, pies y títulos se estudian en Sección 3.2.2 [Títulos, encabezamientos y pies de página personalizados], página 520.

Casi todas las variables de `\paper` funcionan solamente dentro de un bloque `\paper`. Las pocas que funcionan también en un bloque `\layout` están relacionadas en Sección 4.2.1 [El bloque `\layout`], página 579.

Excepto cuando se especifica lo contrario, todas las variables de `\paper` que corresponden a distancias sobre la página se miden en milímetros, a no ser que el usuario especifique unas unidades diferentes. Por ejemplo, la declaración siguiente establece el margen superior `top-margin` a diez milímetros:

```
\paper {
  top-margin = 10
}
```

Para fijarlo en 0.5 pulgadas, utilice el sufijo de unidades `\in`:

```
\paper {
  top-margin = 0.5\in
}
```

Los sufijos de unidades que están disponibles son `\mm`, `\cm`, `\in` y `\pt`. Estas unidades son valores simples para convertir a partir de milímetros; están definidos en `ly/paper-defaults-init.ly`. Por claridad, al usar milímetros, se suele escribir el sufijo `\mm` aunque no es necesario técnicamente.

También es posible definir valores de `\paper` utilizando Scheme. El equivalente de Scheme del ejemplo anterior es:

```
\paper {
  #(define top-margin (* 0.5 in))
}
```

## Véase también

Referencia de la notación: Sección 4.1.2 [Tamaño del papel y escalado automático], página 568, Sección 3.2.2 [Títulos, encabezamientos y pies de página personalizados], página 520, Sección 4.2.1 [El bloque `\layout`], página 579.

Archivos instalados: `ly/paper-defaults-init.ly`.

### 4.1.2 Tamaño del papel y escalado automático

#### Fijar el tamaño del papel

‘A4’ es el valor predeterminado si no se establece ningún tamaño de página explícitamente. Sin embargo, existen dos funciones que se pueden usar para modificarlo:

```
set-default-paper-size
  #(set-default-paper-size "quarto")
  que siempre se debe colocar en el ámbito superior, y

set-paper-size
  \paper {
    #(set-paper-size "tabloid")
  }
```

que se debe colocar siempre dentro de un bloque `\paper`.

Si se usa la función `set-default-paper-size` en el ámbito superior jerárquico, debe ir antes de cualquier bloque `\paper`. `set-default-paper-size` establece el tamaño de todas las páginas, mientras que `set-paper-size` establece sólo el tamaño de las páginas a las que se aplica el bloque

`\paper`. Por ejemplo, si el bloque `\paper` está al principio del archivo, aplica el tamaño de papel a todas las páginas. Si el bloque `\paper` está dentro de un bloque `\book`, entonces el tamaño del papel sólo se aplica a ese libro.

Cuando se usa la función `set-paper-size`, se debe colocar *antes* de cualquier otra función que se utilice dentro del mismo bloque `\paper`. Véase [Escalado automático al tamaño del papel], página 569.

Los tamaños de página están definidos en el archivo `scm/paper.scm`, y aunque es posible añadir tamaños personalizados, se sobrescribirán en las actualizaciones del software posteriores. La lista de todos los tamaños de papel disponibles está en Sección A.5 [Tamaños de página predefinidos], página 706.

Se puede usar la siguiente instrucción en el archivo para añadir un tamaño de página personalizado que se puede usar después con `set-default-paper-size` o con `set-paper-size` según convenga,

```
#(set! paper-alist (cons '("mi tamaño" . (cons (* 15 in) (* 3 in))) paper-alist))

\paper {
  #(set-paper-size "mi tamaño")
}
```

Se puede usar cualquiera de las unidades `in` (pulgadas), `cm` (centímetros) o `mm` (milímetros).

Si se añade el símbolo `'landscape` a la función del tamaño de la página, el papel se gira 90 grados y se establece adecuadamente la anchura de las líneas.

```
#(set-default-paper-size "a6" 'landscape)
```

Se pueden intercambiar las dimensiones del papel *sin que la impresión resulte girada* (como cuando se imprime en tamaño tarjeta postal, o al crear gráficos para su inclusión en lugar de un documento independiente) añadiendo `'landscape` después del nombre del propio tamaño del papel:

```
#(set-default-paper-size "a6landscape")
```

Cuando el tamaño del papel termina de forma explícita con `'landscape` o con `'portrait`, la presencia de un símbolo `'landscape` *solamente* afecta a la orientación de la impresión, no a las dimensiones del papel que se usan para la disposición.

## Véase también

Referencia de la notación: [Escalado automático al tamaño del papel], página 569, Sección A.5 [Tamaños de página predefinidos], página 706.

Archivos instalados: `scm/paper.scm`.

## Escalado automático al tamaño del papel

Si se cambia el tamaño del papel con una de las funciones de Scheme (`set-default-paper-size` o `set-paper-size`), los valores de algunas variables de `\paper` se escalan automáticamente al nuevo tamaño. Para sortear el escalado automático para una variable determinada, fije el valor de la variable después de establecer el tamaño del papel. Observe que el escalado automático no se activa mediante el ajuste de las variables `paper-height` o de `paper-width`, incluso aunque `paper-width` puede influir sobre otros valores (esto es distinto al escalado y se estudia más abajo). Las funciones `set-default-paper-size` y `set-paper-size` se describen en [Fijar el tamaño del papel], página 568.

Las dimensiones verticales afectadas por el escalado automático son `top-margin` y `bottom-margin` (véase Sección 4.1.3 [Variables de espaciado de `\paper` verticales fijas], página 570). Las dimensiones horizontales afectadas por el escalado automático son `left-margin`, `right-margin`, `inner-margin`, `outer-margin`, `binding-offset`, `indent`

y `short-indent` (véase Sección 4.1.5 [Variables de espaciado de `\paper` horizontales], página 573).

Los valores predeterminados para estas dimensiones están fijados en el archivo `ly/paper-defaults-init.ly`, usando variables internas llamadas `top-margin-default`, `bottom-margin-default`, etc. Estos son los valores que resultan del tamaño predeterminado del papel a4. Como referencia, con el papel a4 la altura `paper-height` es 297\mm y la anchura `paper-width` es 210\mm.

## Véase también

Referencia de la notación: Sección 4.1.3 [Variables de espaciado de `\paper` verticales fijas], página 570, Sección 4.1.5 [Variables de espaciado de `\paper` horizontales], página 573.

Archivos de inicio: `ly/paper-defaults-init.ly`, `scm/paper.scm`.

### 4.1.3 Variables de espaciado de `\paper` verticales fijas

**Nota:** Algunas dimensiones de `\paper` se escalan automáticamente al tamaño del papel, lo que puede llevar a un comportamiento no esperado. Véase [Escalado automático al tamaño del papel], página 569.

Los valores predeterminados (previos al escalado) están definidos en el archivo `ly/paper-defaults-init.ly`.

`paper-height` (altura del papel)

Altura de la página, no fijada de forma predeterminada. Observe que el escalado automático de algunas dimensiones verticales no resulta afectado por esto.

`top-margin` (margen superior)

Margen entre el extremo superior de la página y la parte superior de la zona imprimible. Si se modifica el tamaño del papel, el valor predeterminado de esta dimensión se escala de acuerdo con ello.

`bottom-margin` (margen inferior)

Margen entre la parte inferior de la zona imprimible y el extremo inferior de la página. Si se modifica el tamaño del papel, se escala de acuerdo con ello el valor predeterminado de esta dimensión.

`ragged-bottom` (sin justificar por abajo)

Si esto se fija al valor verdadero, los sistemas se disponen según su espaciado natural, ni comprimido ni ampliado verticalmente para rellenar la página.

`ragged-last-bottom` (sin justificar la última por abajo)

Si se establece al valor falso, entonces la última página (tanto del documento como de cada sección que se hubiera creado con un bloque `\bookpart`) se justifica verticalmente de la misma forma que las páginas anteriores.

## Véase también

Referencia de la notación: [Escalado automático al tamaño del papel], página 569.

Archivos de inicio: `ly/paper-defaults-init.ly`.

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

## Advertencias y problemas conocidos

Los títulos (extraídos del bloque `\header`) se tratan como sistemas, así pues `ragged-bottom` y `ragged-last-bottom` aumentan la separación entre los títulos y el primer sistema de la partitura.

Los tamaños de papel definidos explícitamente sobrescriben cualquier ajuste de los márgenes superior o inferior definidos por el usuario.

#### 4.1.4 Variables de espaciado de `\paper` verticales flexibles

En casi todos los casos es preferible que las distancias verticales entre ciertos elementos (como márgenes, títulos, sistemas y las distintas partituras) sean flexibles, de manera que se amplíen y compriman adecuadamente dependiendo de la situación. Están disponibles un cierto número de variables del bloque `\paper` (relacionadas más abajo) para realizar un ajuste fino del comportamiento de estas dimensiones frente a su ampliación o compresión.

Observe que las variables de `\paper` que se estudian en esta sección no controlan el espaciado de los pentagramas que están dentro de los sistemas individuales. El espaciado dentro de los sistemas se controla por medio de propiedades de `grob`, con ajustes que se escriben normalmente dentro de un bloque `\score` o `\layout`, y no dentro del bloque `\paper`. Véase Sección 4.4.1 [Espaciado vertical flexible dentro de los sistemas], página 590.

#### Estructura de las listas-A de espaciado vertical flexible

Cada una de las variables de espaciado de `\paper` flexibles es una lista-A (lista asociativa) que contiene cuatro *claves*:

- **basic-distance** (distancia básica): distancia vertical, medida en espacios de pentagrama, entre los *puntos de referencia* de los dos elementos, cuando no resulte ninguna colisión, y no se produzca ninguna ampliación o compresión. El punto de referencia de un elemento de marcado (de título o del nivel jerárquico superior) es su punto más alto, y el punto de referencia de un sistema es el centro vertical del `StaffSymbol` más cercano (incluso si se trata de una línea que no es un pentagrama, como un contexto `Lyrics`). Los valores de **distancia básica** menores de **padding** o de **minimum-distance** no son significativos, porque la distancia resultante nunca será menor de **padding** o de **minimum-distance**.
- **minimum-distance** (distancia mínima): la distancia vertical mínima permitida, medida en espacios de pentagrama, entre los puntos de referencia de los dos elementos, cuando se produce una compresión. Los valores de **minimum-distance** menores de **padding** no son significativos, porque la distancia resultante nunca será menor de **padding**.
- **padding** (relleno): mínimo espacio vertical vacío necesario entre los rectángulos circundantes (o las líneas de horizonte) de los dos elementos, medido en espacios de pentagrama.
- **stretchability** (ampliabilidad): medida sin unidades de la propensión relativa de esta dimensión a ampliarse. Si es cero, la distancia no se ampliará (a no ser que resultasen colisiones). Si es positiva, la significación del valor de ampliabilidad de una dimensión concreta depende solamente de su relación con los valores de **stretchability** de las otras dimensiones. Por ejemplo, si una dimensión tiene el doble de **stretchability** que otra, se ampliará con el doble de facilidad. Los valores deben ser no negativos y finitos. El valor `+inf.0` desencadena un error de programación y se ignora, pero se puede usar `1.0e7` para un muelle casi infinitamente ampliable. Si no se fija un valor, el valor predeterminado se establece a **basic-distance**. Observe que la propensión de la dimensión a *comprimirse* no se puede establecer directamente por el usuario y es igual a  $(\text{basic-distance} - \text{minimum-distance})$ .

Si una página tiene un margen inferior no justificado, la distancia resultante es la mayor de:

- **basic-distance**,
- **minimum-distance**, y
- **padding** más la menor distancia necesaria para eliminar las colisiones.

Para partituras de varias páginas con la parte inferior no justificada en la última página, ésta usará el mismo espaciado que la página anterior siempre y cuando haya espacio suficiente para ello.



Los métodos específicos para la modificación de las listas-A se estudian en Sección 5.3.7 [Modificación de las listas-A], página 658. El siguiente ejemplo muestra las dos formas en que pueden modificarse estas listas-A. La primera declaración actualiza un par clave-valor individualmente, y la segunda redefine la variable completamente:

```
\paper {
  system-system-spacing.basic-distance = #8
  score-system-spacing =
    #'((basic-distance . 12)
      (minimum-distance . 6)
      (padding . 1)
      (stretchability . 12))
}
```

### Lista de variables de espaciado de \paper verticales flexibles

Los nombres de estas variables siguen el formato *superior-inferior-spacing*, donde *superior* e *inferior* son los elementos que van a ser espaciados. Cada distancia se mide entre los puntos de referencia de los dos elementos (véase más arriba la descripción de la estructura de la lista-A). Observe que en estos nombres de variable, el término ‘markup’ se refiere tanto a *elementos de marcado de título* (bookTitleMarkup o scoreTitleMarkup) como a *elementos de marcado del nivel superior* (véase Sección 3.1.5 [Estructura del archivo], página 510). Todas las distancias se miden en espacios de pentagrama.

Los ajustes predeterminados están definidos en el archivo `ly/paper-defaults-init.ly`.

#### markup-system-spacing

distancia entre un elemento de marcado (de título o del nivel superior) y el sistema que le sigue.

#### score-markup-spacing

distancia entre el último sistema de una partitura y el elemento de marcado (de título o del nivel superior) que le sigue.

#### score-system-spacing

distancia entre el último sistema de una partitura y el primer sistema de la partitura que le sigue, cuando no existe ningún elemento de marcado (de título o del nivel superior) entre ellos.

#### system-system-spacing

distancia entre dos sistemas dentro de la misma partitura.

#### markup-markup-spacing

distancia entre dos elementos de marcado (de título o del nivel superior).

#### last-bottom-spacing

distancia desde el último sistema o elemento de marcado del nivel superior en una página, hasta la parte inferior de la zona imprimible (es decir, el extremo superior del margen inferior).

#### top-system-spacing

distancia desde la parte superior de la zona imprimible (es decir, el extremo inferior del margen superior) hasta el primer sistema de una página, cuando no hay ningún elemento de marcado (de título o del nivel superior) entre los dos.

#### top-markup-spacing

distancia desde el extremo superior de la zona imprimible (es decir, el extremo inferior del margen superior) hasta el primer elemento de marcado (de título o del nivel superior) sobre una página, cuando no hay ningún sistema entre los dos.

## Véase también

Referencia de la notación: Sección 4.4.1 [Espaciado vertical flexible dentro de los sistemas], página 590.

Archivos de inicio: `ly/paper-defaults-init.ly`.

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

### 4.1.5 Variables de espaciado de `\paper` horizontales

**Nota:** Algunas dimensiones de `\paper` se escalan automáticamente al tamaño del papel, lo que puede dar lugar a un comportamiento distinto al esperado. Véase [Escalado automático al tamaño del papel], página 569.

## Variables de `\paper` para la anchura y los márgenes

Los valores predeterminados (antes del escalado) que no están relacionados aquí se encuentran definidos en el archivo `ly/paper-defaults-init.ly`.

### `paper-width` (anchura del papel)

Anchura de la página, sin fijar de forma predeterminada. Aunque `paper-width` no tiene ningún efecto sobre el escalado automático de algunas dimensiones horizontales, sí influye sobre la variable `line-width`. Si están establecidas las dos variables `paper-width` y `line-width`, entonces también se actualizan `left-margin` y `right-margin`. Véase también `check-consistency`.

### `line-width` (longitud de la línea)

Cuando se especifica dentro de un bloque `\paper`, define las dimensiones horizontales del espacio disponible para las líneas del pentagrama en los sistemas sin sangrado por la izquierda. Si se deja sin especificar, la longitud `line-width` del papel se determina a partir de  $(\text{paper-width} - \text{left-margin} - \text{right-margin})$ . Si se especifica la longitud `line-width` del papel, y no se especifican el margen izquierdo `left-margin` ni el margen derecho `right-margin`, entonces los márgenes se actualizan de forma que los sistemas se centran automáticamente sobre la página. Véase también `check-consistency`.

Los valores de `line-width` para las partituras individuales se pueden especificar dentro de los bloques `\layout` de las partituras. Estos valores controlan la longitud de las líneas producidas partitura a partitura. Si la longitud `line-width` no se especifica para una partitura, su valor predeterminado es el `line-width` del bloque `paper`. El hecho de fijar la longitud `line-width` de una partitura no tiene efecto sobre los márgenes del papel. Las líneas del pentagrama, de una longitud determinada por el valor `line-width` de la partitura, se alinean por la izquierda dentro del área definida por la longitud `line-width` del papel. Si la longitud `line-width` del papel y de la partitura son iguales, las líneas del pentagrama se extienden exactamente desde el margen izquierdo hasta el margen derecho, pero si la longitud `line-width` de la partitura es mayor que la `line-width` del papel, las líneas del pentagrama se imprimirán encima del espacio reservado al margen derecho.

### `left-margin` (margen izquierdo)

Margen entre el extremo izquierdo de la página y el comienzo de las líneas del pentagrama en los sistemas sin sangrado. Si el tamaño del papel se modifica, el valor predeterminado de esta dimensión se escala de acuerdo con ello. Si no se fija el valor de `left-margin`, y tanto `line-width` como `right-margin` están fijados, entonces se establece el valor

de `left-margin` a  $(\text{paper-width} - \text{line-width} - \text{right-margin})$ . Si solamente está fijado `line-width`, entonces los dos márgenes se establecen a  $((\text{paper-width} - \text{line-width}) / 2)$ , y como consecuencia los sistemas se centran sobre la página. Véase también `check-consistency`.

#### `right-margin` (margen derecho)

Margen entre el extremo derecho de la página y el final de las líneas del pentagrama en sistemas justificados. Si el tamaño de la página se modifica, el valor predeterminado de esta dimensión se escala de acuerdo con ello. Si `right-margin` está sin fijar, y tanto `line-width` como `left-margin` están fijados, el valor de `right-margin` se establece a  $(\text{paper-width} - \text{line-width} - \text{left-margin})$ . Si solamente está fijado `line-width`, los dos márgenes se establecen a  $((\text{paper-width} - \text{line-width}) / 2)$ , y como consecuencia los sistemas aparecen centrados sobre la página. Véase también `check-consistency`.

#### `check-consistency` (comprobar la consistencia)

Si esto es verdadero (su valor predeterminado), se imprime una advertencia en caso de que la suma de `left-margin`, `line-width` y `right-margin` no coincida exactamente con el valor de `paper-width`, y se sustituye cada uno de estos valores (excepto `paper-width`) con su valor predeterminado (escalado al tamaño del papel si es necesario). Si está fijado al valor falso, se ignora cualquier inconsistencia y se permite que los sistemas se salgan del borde del papel.

#### `ragged-right` (no justificar por la derecha)

Si está fijado al valor verdadero, los sistemas no cubren el ancho de la línea. En lugar de ello, los sistemas finalizan en su longitud horizontal natural. Valor predeterminado: `#t` para las partituras con un solo sistema, y `#f` para partituras que tienen dos o más sistemas. Esta variable también se puede establecer dentro de un bloque `\layout`.

#### `ragged-last` (no justificar el último)

Si está fijado al valor verdadero, el último sistema de la partitura no llena toda la anchura de la línea. En lugar de ello, el último sistema finaliza en su longitud horizontal natural. Valor predeterminado: `#f`. Esta variable también puede ajustarse dentro de un bloque `\layout`.

### Véase también

Referencia de la notación: [Escalado automático al tamaño del papel], página 569.

Archivos de inicio: `ly/paper-defaults-init.ly`.

### Advertencias y problemas conocidos

Los tamaños de papel definidos explícitamente sobrescriben cualquier ajuste de los márgenes izquierdo o derecho definidos por el usuario.

### Variables de `\paper` para el modo de doble cara

Los valores predeterminados (antes del escalado) se definen en `ly/paper-defaults-init.ly`.

#### `two-sided` (dos caras)

Si tiene el valor verdadero, utilizar `inner-margin`, `outer-margin` y `binding-offset` para determinar los márgenes dependiendo de si el número de la página es par o impar. Esto sobrescribe a `left-margin` y a `right-margin`.

#### `inner-margin` (margen interno)

Margen que todas las páginas tienen en el lado interno si forman parte de un libro. Si el tamaño del papel se modifica, el valor predeterminado de esta dimensión se escala de acuerdo con ello. Solamente funciona con `two-sided` fijado al valor verdadero.

**outer-margin (margen externo)**

Margen que todas las páginas tienen en el lado externo si forman parte de un libro. Si el tamaño del papel se modifica, el valor predeterminado de esta dimensión se escala de acuerdo con ello. Solamente funciona con `two-sided` fijado al valor verdadero.

**binding-offset (margen de encuadernación)**

Medida en que se aumenta `inner-margin` para asegurar que no se oculta nada a causa de la encuadernación. Si el tamaño del papel se modifica, el valor predeterminado de esta dimensión se escala de acuerdo con ello. Solamente funciona con `two-sided` fijado al valor verdadero.

**Véase también**

Referencia de la notación: [Escalaado automático al tamaño del papel], página 569.

Archivos de inicio: `ly/paper-defaults-init.ly`.

**Variables de `\paper` para desplazamientos y sangrados**

Los valores predeterminados (antes del escalaado) que no se relacionan aquí están definidos en el archivo `ly/paper-defaults-init.ly`.

**horizontal-shift (desplazamiento horizontal)**

Medida en que todos los sistemas (incluidos los títulos de cabecera y los separadores de sistemas) se desplazan a la derecha. Predeterminado: 0.0.

**indent (sangrado)**

Nivel de sangrado para el primer sistema de una partitura. Si se modifica el tamaño del papel, el valor predeterminado de esta dimensión se escala de acuerdo con ello. El espacio dentro de `line-width` que está disponible para el primer sistema se reduce en esta cantidad. `indent` se puede también especificar dentro de los bloques `\layout` para ajustar los sangrados partitura a partitura.

**short-indent (sangrado reducido)**

Nivel de sangrado para todos los sistemas de una partitura excepto el primero. Si el tamaño del papel se modifica, el valor predeterminado de esta dimensión se escala de acuerdo con ello. El espacio dentro de `line-width` que está disponible para los sistemas que no son el primero, se reduce en esta cantidad. También se puede especificar `short-indent` dentro de los bloques `\layout` para ajustar los sangrados reducidos partitura a partitura.

**Véase también**

Referencia de la notación: [Escalaado automático al tamaño del papel], página 569.

Archivos de inicio: `ly/paper-defaults-init.ly`.

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

**4.1.6 Otras variables de `\paper`****Variables de `\paper` para los saltos de línea****max-systems-per-page**

Número máximo de sistemas que se colocan sobre una página. Solo está contemplado por el algoritmo `ly:optimal-breaking`. Valor predeterminado: no fijado.

**min-systems-per-page**

Número mínimo de sistemas que se colocan sobre una página. Puede causar que las páginas aparezcan atiborradas si el valor es demasiado grande. En el momento

actual solo está contemplado por el algoritmo `ly:optimal-breaking`. Valor predeterminado: no fijado.

#### `systems-per-page`

Número de sistemas que se deben colocar en cada página. En el momento actual solo está contemplado por el algoritmo `ly:optimal-breaking`. Valor predeterminado: no fijado.

#### `system-count`

Número de sistemas que utilizar para una partitura. Valor predeterminado: no fijado. Esta variables también se puede ajustar dentro de un bloque `\layout`.

### Véase también

Referencia de la notación: Sección 4.3.1 [Saltos de línea], página 582.

### Variables de `\paper` para los saltos de página

Los valores predeterminados que no están relacionados aquí, se definen en el archivo `ly/paper-defaults-init.ly`

#### `page-breaking`

Algoritmo de saltos de página que utilizar. Se puede elegir entre `ly:minimal-breaking`, `ly:page-turn-breaking`, `ly:one-page-breaking`, `ly:one-line-breaking`, `ly:one-line-auto-height-breaking` y `ly:optimal-breaking` (predeterminado).

#### `page-breaking-system-system-spacing`

Engaña a la parte del programa encargada de los saltos de página para que piense que `system-system-spacing` está establecido a un valor distinto al que tiene realmente. Por ejemplo, si `page-breaking-system-system-spacing.padding` está establecido a algo que es mucho mayor que `system-system-spacing.padding`, entonces el divisor de páginas pone menos sistemas en cada página. Valor predeterminado: sin ajustar.

#### `page-count`

Número de páginas que utilizar en una partitura, no ajustado de forma predeterminada.

Las siguientes variables son efectivas solamente cuando `page-breaking` está fijado a `ly:page-turn-breaking`. Los saltos de página se seleccionan entonces de forma que se minimice el número de pasos de página. Dado que los pasos de página son necesarios cuando se pasa de una página impar a una página par, normalmente resultará favorecida aquella disposición en que la última página sea impar. Los lugares en que se prefieran pasos de página pueden indicarse manualmente insertando `\allowPageTurn` o automáticamente incluyendo el grabador `Page_turn_engraver` (véase [Paso de página óptimo], página 588).

Si no hay un número suficiente de posibilidades para escoger a la hora de insertar unos pasos de página adecuados, LilyPond puede insertar una página vacía dentro de una partitura, entre partituras (si hay dos o más partituras) o finalizando una partitura en una página de numeración par. Los valores de las tres siguientes variables pueden incrementarse para que estas acciones se vuelvan menos probables.

Los valores son penalizaciones, es decir, cuando más alto es el valor menos probables será la acción asociada relativa a otras posibilidades.

#### `blank-page-penalty` (penalización por página vacía)

Penalización por tener una página vacía en mitad de una partitura. Si `blank-page-penalty` es grande y está seleccionado `ly:page-turn-breaking`,

entonces será menos probable que LilyPond inserte una página en medio de una partitura. En lugar de ello, intercala espacio en la música hasta que se rellene la página vacía y la siguiente. Predeterminado: 5.

#### **blank-last-page-penalty** (penalización por última página vacía)

Penalización por terminar la partitura sobre una página par. Si **blank-last-page-penalty** es grande y está seleccionado **ly:page-turn-breaking**, entonces será menos probable que LilyPond produzca una partitura en que la última página tenga numeración par. En su lugar, ajustará el espaciado de manera que use una página más o una página menos. Predeterminado: 0.

#### **blank-after-score-page-penalty** (penalización por página vacía después de una partitura)

Penalización por tener una página vacía después del final de una partitura y antes de la siguiente. De forma predeterminada, esta penalización es más pequeña que **blank-page-penalty**, de forma que se insertan páginas vacías después de las partituras como preferencia sobre la inserción de páginas vacías dentro de una partitura. Predeterminado: 2.

### **Véase también**

Referencia de la notación: Sección 4.3.2 [Saltos de página], página 586, [Saltos de página óptimos], página 587, [Paso de página óptimo], página 588, [Saltos de página mínimos], página 587, [Saltos de página del tipo página única], página 588, [Saltos de página del tipo una-línea], página 588, [Saltos de página del tipo una-línea-altura-automática], página 588.

Archivos de inicio: `ly/paper-defaults-init.ly`.

### **Variables de \paper para la numeración de las páginas**

Los valores predeterminados que no se relacionan aquí están definidos en el archivo `ly/paper-defaults-init.ly`

#### **auto-first-page-number**

El algoritmo de división de páginas está afectado por el hecho de que el número de la primera página sea par o impar. Si está establecido al valor verdadero, el algoritmo de división de páginas decide si comenzar con un número par o impar. Esto hace que el número de la primera página se quede como está, o que se aumente en una unidad. Predeterminado: `#f`.

#### **first-page-number**

Valor del número de página en la primera página.

#### **print-first-page-number**

Si está fijado a verdadero, se imprime un número de página en la primera página.

#### **print-page-number**

Si está fijado a falso, no se imprimen los números de página.

#### **page-number-type**

Tipo de numeración para las páginas. Entre las posibilidades están **roman-lower** (números romanos en minúsculas), **roman-upper** (números romanos en mayúsculas) y **arabic** (cifras árabigas). Valor predeterminado: `'arabic'`.

### **Véase también**

Archivos de inicio: `ly/paper-defaults-init.ly`.

## Advertencias y problemas conocidos

Los números de página impares siempre están a la derecha. Si quiere que la música comience en la página 1 debe haber una página en blanco en el reverso de la página de cubierta de manera que la página 1 esté en el lado derecho.

## Variables de `\paper` diversas

### `page-spacing-weight`

Importancia relativa del espacio (vertical) de las páginas y el espaciado (horizontal) de las líneas. Los valores altos hacen que el espaciado de la página tenga más importancia. Predeterminado: 10.

### `print-all-headers`

Si está fijado a verdadero, imprime todos los encabezamientos para cada una de las partituras `\score` de la salida. Normalmente sólo se imprimen las variables de encabezamiento `piece` y `opus`. Valor predeterminado: `#f`.

### `system-separator-markup`

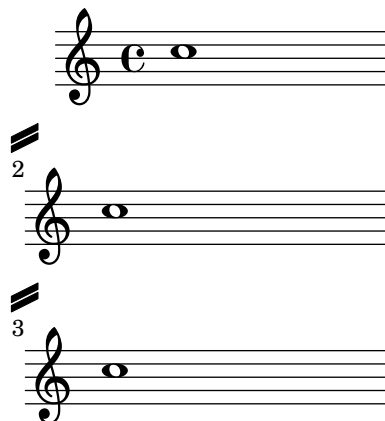
Objeto de marcado que se inserta entre los sistemas. Se suele usar para partituras orquestales. Predeterminado: sin establecer. El elemento de marcado `\slashSeparator`, definido en el archivo `ly/titling-init.ly`, se aporta como un valor predeterminado adecuado, por ejemplo:

```

#(set-default-paper-size "a8")

\book {
  \paper {
    system-separator-markup = \slashSeparator
  }
  \header {
    tagline = ##f
  }
  \score {
    \relative { c''1 \break c1 \break c1 }
  }
}

```



### `footnote-separator-markup`

A markup object that is inserted above the footnote texts at the bottom of the page. Default: a centered horizontal line, defined in `ly/paper-defaults-init.ly`.

## Véase también

Archivos de inicio: `ly/titling-init.ly`, `ly/paper-defaults-init.ly`.

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

## Advertencias y problemas conocidos

El encabezamiento de página predeterminado pone el número de página y el campo `instrument` del bloque `\header` sobre la misma línea.

## 4.2 Disposición de la partitura

Esta sección trata de las opciones de disposición de la partitura para el bloque `\layout`.

### 4.2.1 El bloque `\layout`

Así como el bloque `\paper` contiene ajustes relativos al formateo de la página para todo el documento, el bloque `\layout` contiene ajustes para la disposición específica de las partituras. Para fijar las opciones de disposición de partituras de forma global, escribálas en un bloque `\layout` del nivel jerárquico superior. Para fijar opciones de disposición para una partitura individual escribálas dentro de un bloque `\layout` dentro del bloque `\score`, después de la música. Entre los ajustes que pueden aparecer en un bloque `\layout` se encuentran los siguientes:

- la función de Scheme `layout-set-staff-size`,
- las modificaciones de contexto de los bloques `\context`, y
- las variables de `\paper` que afectan a la disposición de la partitura.

La función `layout-set-staff-size` se estudia en la sección siguiente, Sección 4.2.2 [Establecer el tamaño del pentagrama], página 581. Las modificaciones de contexto se estudian en un capítulo distinto; véase Sección 5.1.4 [Modificar los complementos (plug-ins) de contexto], página 631, y Sección 5.1.5 [Cambiar los valores por omisión de los contextos], página 633.

Las variables de `\paper` que pueden aparecer dentro de un bloque `\layout`, y cuyos valores predeterminados se toman del bloque `paper`, son:

- `line-width`, `ragged-right` y `ragged-last` (véase [Variables de `\paper` para la anchura y los márgenes], página 573)
- `indent` y `short-indent` (véase [Variables de `\paper` para desplazamientos y sangrados], página 575)
- `system-count` (véase [Variables de `\paper` para los saltos de línea], página 575)

He aquí un ejemplo de un bloque `\layout`:

```
\layout {
  indent = 2\cm
  \context {
    \StaffGroup
    \override StaffGroup.staff-staff-spacing.basic-distance = #8
  }
  \context {
    \Voice
    \override TextScript.padding = #1
    \override Glissando.thickness = #3
  }
}
```

Se pueden escribir varios bloques `\layout` como expresiones del nivel superior. Esto puede ser útil, por ejemplo, si hay distintos ajustes almacenados en diversos archivos y se incluyen



opcionalmente. Internamente, cuando se evalúa un bloque `\layout`, se hace una copia de la configuración de `\layout` actual, y después se aplican los cambios definidos en el bloque; el resultado se guarda como la nueva configuración actual. Desde el punto de vista del usuario, los bloques `\layout` se combinan, pero en situaciones de conflicto (cuando se cambia la misma propiedad en diferentes bloques) las definiciones más tardías toman precedencia.

Por ejemplo: si este bloque:

```
\layout {
  \context {
    \Voice
    \override TextScript.color = #magenta
    \override Glissando.thickness = #1.5
  }
}
```

se coloca después del del ejemplo anterior, las sobrescrituras de `'padding` y de `'color` para `TextScript` se combinan, pero la última sobrescritura `'thickness` para `Glissando` sustituye (u oculta) a la anterior.

Los bloques `\layout` se pueden asignar a variables para una reutilización posterior, pero la forma en que esto funciona es ligera y significativamente diferente del hecho de escribirlas literalmente.

Si se define una variable de la siguiente manera:

```
variable_de_Layout = \layout {
  \context {
    \Voice
    \override NoteHead.font-size = #4
  }
}
```

guardará la configuración actual de `\layout` con la sobrescritura añadida de `NoteHead.font-size`, pero esta combinación *no* se guarda como la nueva configuración actual. Tenga en cuenta que la ‘configuración actual’ se lee cuando la variable se define y no cuando se usa, de manera que el contenido de la variable depende de su posición dentro del código fuente.

La variable se puede entonces usar dentro de otro bloque `\layout`, por ejemplo:

```
\layout {
  \variable_de_layout
  \context {
    \Voice
    \override NoteHead.color = #red
  }
}
```

Un bloque `\layout` que contiene una variable, como en el ejemplo anterior, *no* copia la configuración actual, sino que en lugar de ello usa el contenido de `\variable_de_layout` como la configuración base para las adiciones posteriores. Esto significa que cualquier cambio definido entre la definición y el uso de la variable, se pierde.

Si la `variable_de_layout` se define (o se incluye con `\include`) inmediatamente antes de que se utilice, su contenido es sencillamente la configuración en curso más las sobrescrituras que se hayan definido dentro de él. Así pues, en el ejemplo anterior que muestra el uso de `\variable_de_layout` el último bloque `\layout` consistiría en:

```
TextScript.padding = #1
TextScript.color = #magenta
Glissando.thickness = #1.5
```

```
NoteHead.font-size = #4
NoteHead.color = #red
```

más las sobreescrituras de `indent` y de `StaffGrouper`.

Pero si la variable ya hubiera sido definida antes del primer bloque `\layout` la configuración actual contendría ahora solamente

```
NoteHead.font-size= #4 % (escrita en la definición de la variable)
NoteHead.color = #red % (añadida después del uso de la variable)
```

Si se planifica cuidadosamente, las variables de `\layout` pueden ser una herramienta valiosa para estructurar el diseño de la disposición de las fuentes, y también para reiniciar la configuración de `\layout` a un estado conocido.

## Véase también

Referencia de la notación: Sección 5.1.5 [Cambiar los valores por omisión de los contextos], página 633.

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

### 4.2.2 Establecer el tamaño del pentagrama

El **tamaño de pentagrama** predeterminado es de 20 puntos, que corresponde a una altura del pentagrama de 7.03 mm (un punto es 100/7227 de pulgada, o 2540/7227 mm). El tamaño del pentagrama se puede modificar de tres formas:

1. Para establecer globalmente el tamaño del pentagrama para todas las partituras de un archivo (o en un bloque `\book`, para ser exactos), utilice `set-global-staff-size`:

```
 #(set-global-staff-size 14)
```

El ejemplo de arriba establece el tamaño global predeterminado del pentagrama a 14 pt (4.92 mm) y escala todas las tipografías según corresponda.

2. Para fijar el tamaño del pentagrama para una sola partitura dentro de un libro, use `layout-set-staff-size` dentro del bloque `\layout` de dicha partitura:

```
\score {
  ...
  \layout {
    #(layout-set-staff-size 14)
  }
}
```

3. Para fijar el tamaño del pentagrama para una sola pauta dentro de un sistema, use la instrucción `\magnifyStaff`. Por ejemplo, las partituras de música de cámara editadas de la forma tradicional, con piano, usan habitualmente pautas de 7 mm para el piano, mientras que los otros pentagramas suelen estar entre 3/5 y 5/7 de reducción (entre el 60% y el 71%). Para obtener la proporción de 5/7, escriba:

```
\score {
  <<
    \new Staff \with {
      \magnifyStaff #5/7
    } { ... }
    \new PianoStaff { ... }
  >>
}
```

En caso de que usted sepa qué tamaño de `fontSize` quiere usar, puede emplear la forma siguiente:

```
\score {
```

```

<<
  \new Staff \with {
    \magnifyStaff #(magstep -3)
  } { ... }
  \new PianoStaff { ... }
>>
}

```

Para imitar la apariencia de las ediciones tradicionales, es mejor evitar reducir el grosor de las líneas del pentagrama.

## Peso automático de la tipografía a distintos tamaños

La tipografía Emmentaler ofrece el conjunto de glifos musicales *Feta* en ocho tamaños distintos, cada uno de ellos ajustado a un tamaño de pentagrama. Cuanto más pequeño es el tamaño de los glifos, más “pesados” se vuelven, de manera que armonicen con las líneas del pentagrama, que son también relativamente más gruesas. Los tamaños de glifo recomendados se relacionan en la tabla siguiente:

nombre de la fuente	altura del pentagrama (pt)	altura del pentagrama (mm)	usos
feta11	11.22	3.9	partituras de bolsillo
feta13	12.60	4.4	
feta14	14.14	5.0	
feta16	15.87	5.6	
feta18	17.82	6.3	cancioneros
feta20	20	7.0	particellas estándar
feta23	22.45	7.9	
feta26	25.2	8.9	

## Véase también

Referencia de la notación: [Seleccionar el tamaño de la tipografía para la notación], página 235, Sección A.8 [La tipografía Emmentaler], página 713.

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

## 4.3 Saltos

### 4.3.1 Saltos de línea

Normalmente los saltos de línea se determinan automáticamente. Se eligen de forma que las líneas no aparezcan demasiado apretadas ni demasiado sueltas, y que las líneas consecutivas tengan una densidad similar.

Para forzar manualmente un salto de línea sobre la línea divisoria, utilice la instrucción `\break`:

```

\relative c'' {
  c4 c c c | \break
  c4 c c c |
}

```

}



2



De forma predeterminada se ignora cualquier instrucción `\break` insertada en la mitad de un compás, y se imprime un mensaje de advertencia durante la compilación del archivo de entrada de LilyPond. Para forzar un salto de línea en medio de un compás, añada una línea divisoria invisible con `\bar ""`:

```
\relative c'' {
  c4 c c
  \bar ""
  \break
  c |
  c4 c c c |
}
```



También se ignora cualquier instrucción `\break` que se produzca en la línea divisoria si el compás anterior termina en medio de una nota, como cuando un grupo de valoración especial comienza en un compás y termina en otro distinto. En este caso, elimine el grabador `Forbid_line_break_engraver` del contexto `Voice` y use una construcción de música simultánea insertando el `break` en el lugar adecuado dentro de la segunda voz:

```
\new Voice \with {
  \remove "Forbid_line_break_engraver"
} \relative {
  <<
    { c''2. \tuplet 3/2 { c4 c c } c2. | }
    { s1 | \break s1 | }
  >>
}
```



2



De forma similar, el funcionamiento predeterminado es que los saltos de línea resulten ignorados cuando hay barras de corchea que cruzan la línea divisoria. Utilice la instrucción `\override Beam.breakable = ##t` para forzar esto:

```
\relative c'' {
  \override Beam.breakable = ##t
  c2. c8[ c | \break
  c8 c] c2. |
}
```



La instrucción `\noBreak`, evita que se produzca un salto de línea en la barra divisoria en que se inserta.

Dentro de una partitura, se evitan los saltos de línea automáticos dentro de la música que se encuentra entre las instrucciones `\autoLineBreaksOff` y `\autoLineBreaksOn`. Si también se quieren evitar los saltos de página automáticos, deben usarse las instrucciones `\autoBreaksOff` y `\autoBreaksOn`. Los saltos manuales no están afectados por estas instrucciones. Observe que el hecho de inhibir los saltos de línea automáticos puede hacer que la música se extienda sobre el margen derecho cuandoquiera que no cabe completamente dentro de una sola línea.

Los saltos automáticos de línea (no de página) se pueden habilitar sobre líneas divisorias individuales mediante el uso de `\once \autoLineBreaksOn` en una barra. La instrucción identifica un salto de línea permitido, en lugar de uno prohibido.

Los ajustes más básicos que influyen sobre el espaciado de las líneas son `indent` y `line-width`. Se establecen dentro del bloque `\layout`. Controlan el sangrado de la primera línea de música, y la longitud de las líneas.

Si se establece `ragged-right` a verdadero en el bloque `\layout`, los sistemas terminan en su longitud horizontal natural, en lugar de distribuirse horizontalmente para llenar toda la línea. Esto es de utilidad para fragmentos cortos, y para efectuar una comprobación de lo apretado que es el espaciado natural.

La opción `ragged-last` es similar a `ragged-right`, pero afecta sólo a la última línea de la pieza.

```
\layout {
  indent = 0\mm
  line-width = 150\mm
  ragged-last = ##t
}
```

Para saltos de línea a intervalos regulares utilice `\break` separado mediante desplazamientos con `\skip` y repetidos con `\repeat`. Por ejemplo, esto haría que los 28 compases siguientes (suponiendo un compás de 4/4) se dividan cada cuatro compases, y sólo en dichos lugares:

```
<<
\repeat unfold 7 {
  s1 \noBreak s1 \noBreak
  s1 \noBreak s1 \break
}
```

```
{ la música real... }
>>
```

## Instrucciones predefinidas

```
\break, \noBreak, \autoBreaksOff, \autoBreaksOn, \autoLineBreaksOff,
\autoLineBreaksOn.
```

## Fragmentos de código seleccionados

### *Using an extra voice for breaks*

Often it is easier to manage line and page-breaking information by keeping it separate from the music by introducing an extra voice containing only skips along with the `\break`, `pageBreak` and other layout information.

This pattern becomes especially helpful when overriding `line-break-system-details` and the other useful but long properties of `NonMusicalPaperColumnGrob`.

```
music = \relative c'' { c4 c c c }

\score {
  \new Staff <<
    \new Voice {
      s1 * 2 \break
      s1 * 3 \break
      s1 * 6 \break
      s1 * 5 \break
    }
    \new Voice {
      \repeat unfold 2 { \music }
      \repeat unfold 3 { \music }
      \repeat unfold 6 { \music }
      \repeat unfold 5 { \music }
    }
  }
  >>
}
```





## Véase también

Referencia de la notación: [Variables de `\paper` para los saltos de línea], página 575, `\undefined` [`\undefined`], página `\undefined`.

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “LineBreakEvent” en *Referencia de Funcionamiento Interno*.

## Advertencias y problemas conocidos

Si se escribe `\autoLineBreaksOff` o `\autoBreaksOff` antes de cualquier contenido musical, aparecerán mensajes de error. Escriba siempre estas instrucciones después de algo de música.

### 4.3.2 Saltos de página

Esta sección describe los distintos métodos para crear saltos de página, y cómo modificarlos.

#### Saltos de página manuales

Se puede sobrescribir el mecanismo predeterminado de salto de página insertando instrucciones `\pageBreak` o `\noPageBreak`. Estas instrucciones son análogas a `\break` y `\noBreak`. Se deben insertar en una línea divisoria. Estas instrucciones fuerzan y prohíben, respectivamente, la eventualidad de un salto de página en dicha línea divisoria. Por supuesto, la instrucción `\pageBreak` también fuerza un salto de línea.

Las instrucciones `\pageBreak` y `\noPageBreak` también se pueden insertar en el nivel más alto, entre las partituras y los elementos de marcado situados en el nivel superior.

Dentro de una partitura, no se producen saltos de página automáticos en el interior de la música que se encuentra entre las instrucciones `\autoPageBreaksOff` y `\autoPageBreaksOn`. Los saltos de página manuales no resultan afectados por estas instrucciones.

Hay ajustes análogos a `ragged-right` y `ragged-last` que tienen el mismo efecto sobre el espaciado vertical. Si `ragged-bottom` se establece a `#t` los sistemas no se justifican verticalmente. Cuando el valor de `ragged-last-bottom` se fija a `#t`, como lo está de forma predeterminada, se permite una separación al final de la última página (o la última página de cada uno de las partes de libro `\bookpart`). Véase Sección 4.1.3 [Variables de espaciado de `\paper` verticales fijas], página 570.

Los saltos de página se calculan por medio de la función `page-breaking`. LilyPond ofrece varios algoritmos para el cómputo de los saltos de página, entre ellos `ly:optimal-breaking`, `ly:page-turn-breaking` y `ly:minimal-breaking`. El predeterminado es `ly:optimal-breaking`, pero el valor se puede cambiar en el bloque `\paper`:

```
\paper {
  page-breaking = #ly:page-turn-breaking
}
```

Cuando un libro tiene muchas partituras y páginas, puede ser difícil resolver el problema de los saltos de página, necesitando mucha memoria y prolongados tiempos de procesamiento. Para facilitar el proceso de división en páginas, se usan los bloques `\bookpart` para dividir el libro en varias partes: los saltos de página se producen de manera independiente en cada parte. También se pueden usar diferentes funciones de división en páginas para las distintas partes del libro.

```
\bookpart {
  \header {
    subtitle = "Prefacio"
  }
}
```

```

\paper {
  %% En una parte que consiste en texto principalmente,
  %% puede ser preferible ly:minimal-breaking
  page-breaking = #ly:minimal-breaking
}
\markup { ... }
...
}
\bookpart {
  %% En esta parte, consistente en música, se usa la función
  %% óptima predeterminada de saltos de página.
  \header {
    subtitle = "Primer movimiento"
  }
  \score { ... }
  ...
}

```

## Instrucciones predefinidas

`\pageBreak`, `\noPageBreak`, `\autoPageBreaksOn`, `\autoPageBreaksOff`.

## Véase también

Referencia de la notación: [Variables de `\paper` para los saltos de página], página 576.

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

## Advertencias y problemas conocidos

El prefijo `\once` no es efectivo con `\autoPageBreaksOn` y `\autoPageBreaksOff`. Si se ha desactivado la división automática de páginas y después se vuelve a activar para permitir un salto, debe continuar activada durante algunos compases (cuyo número exacto dependerá de la partitura) antes de volverse a desactivar, pues en caso contrario podría perderse la oportunidad de que el salto se produzca.

## Saltos de página óptimos

La función `ly:optimal-breaking` es el método predeterminado de LilyPond para determinar los saltos de página. Intenta hallar una división de páginas que haga mínimos el apretujamiento y la distensión, tanto horizontal como verticalmente. A diferencia de `ly:page-turn-breaking`, no tiene un concepto de los pasos de página.

## Véase también

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

## Saltos de página mínimos

La función `ly:minimal-breaking` efectúa unos cálculos mínimos para determinar los saltos de página: completa una página con tantos sistemas como sea posible antes de continuar con la siguiente. Así, puede preferirse para partituras con muchas páginas, donde las otras funciones de salto de página pueden resultar demasiado lentas o ávidas de memoria, o con una gran cantidad de textos. Se habilita utilizando:

```

\paper {
  page-breaking = #ly:minimal-breaking
}

```



## Véase también

Snippets: Sección “Spacing” en *Fragmentos de código*.

## Saltos de página del tipo página única

La función `ly:one-page-breaking` (saltos de página única) es un algoritmo de saltos de página de propósito especial que ajusta automáticamente la altura de la página para que quepa toda la música en una sola página. La variable `paper-height` del bloque `paper` se ignora, pero los otros ajustes funcionan de la manera acostumbrada. Concretamente, el espaciado entre el último sistema (o elemento de marcado de nivel superior) y el pie de página se puede personalizar con `last-bottom-spacing` dentro del bloque `paper`. La anchura de la página se queda sin alterar por omisión, pero se puede fijar con la variable `paper-width` del bloque `paper`.

## Advertencias y problemas conocidos

`ly:one-page-breaking` no es compatible, de momento, con `\bookpart`.

## Saltos de página del tipo una-línea

La función `ly:one-line-breaking` es un algoritmo de saltos de página para uso especial que pone cada partitura en una página y en una sola línea. Esta función de salto de página no compone los títulos ni los márgenes; solamente se imprime la partitura.

El ancho de la página se ajusta de forma que la partitura más larga quepa en una línea. Concretamente, las variables `paper-width`, `line-width` y `indent` del bloque `\paper` se ignoran, aunque `left-margin` y `right-margin` aún se respetan. La altura de la página se deja sin cambios.

## Saltos de página del tipo una-línea-altura-automática

La función `ly:one-line-auto-height-breaking` funciona igual que `ly:one-line-breaking` excepto que la altura de la página se modifica automáticamente para hacerse corresponder con la altura de la música. De forma específica, la variable `paper-height` del bloque `\paper` se ajusta de tal manera que abarca los extremos superior e inferior de la partitura más alta más los márgenes superior e inferior `top-margin` y `bottom-margin`.

Observe que el ajuste `top-system-spacing` afecta a la posición vertical de la música. Fíjelo al valor `##f` dentro de un bloque `paper` para disponer la música sencillamente entre los márgenes superior e inferior.

## Paso de página óptimo

Con frecuencia es necesario encontrar una configuración de división de páginas de manera que haya un silencio al final de una página de cada dos. De esta forma, el músico puede pasar la página sin perder notas. La función `ly:page-turn-breaking` trata de encontrar una división de páginas que haga mínimos el apretujamiento y el estiramiento, pero con la restricción añadida de que sólo se permite introducir vueltas de página en los lugares especificados.

Hay dos etapas en el uso de esta función de división de páginas. En primer lugar debemos habilitarlo en el bloque `\paper`, como se explicó en Sección 4.3.2 [Saltos de página], página 586. Entonces debemos decirle a la función dónde nos gustaría permitir los saltos de página.

Hay dos formas de conseguir la segunda tarea. Primero, podemos especificar manualmente cada uno de los pasos de página potenciales, insertando `\allowPageTurn` en nuestro archivo de entrada en los lugares adecuados.

Si esto es demasiado tedioso, podemos añadir un grabador `Page_turn_engraver` a un contexto `Staff` o `Voice`. El grabador `Page_turn_engraver` analizará el contexto en busca de secciones sin notas (observe que no busca silencios, sino la ausencia de notas. Se hace así para que la polifonía en un solo pentagrama con silencios en una de las voces no arruine la labor del grabador `Page_turn_engraver`). Cuando encuentra una sección sin notas suficientemente larga, el

grabador `Page_turn_engraver` inserta un `\allowPageTurn` en la última barra de compás de dicha sección, a no ser que haya una barra ‘especial’ de compás (como una doble barra), en cuyo caso se insertará el `\allowPageTurn` en la última barra ‘especial’ de compás de la sección.

El grabador `Page_turn_engraver` lee la propiedad de contexto `minimumPageTurnLength` para determinar qué longitud debe tener una sección sin notas antes de que se considere la posibilidad de un paso de página. El valor predeterminado para `minimumPageTurnLength` es `(ly:make-moment 1/1)`. Si quiere inhabilitar las vueltas de página, fíjelo a algún valor ‘muy grande’.

```
\new Staff \with { \consists "Page_turn_engraver" }
{
  a4 b c d |
  R1 | % aquí se permite un salto de página
  a4 b c d |
  \set Staff.minimumPageTurnLength = #(ly:make-moment 5/2)
  R1 | % aquí no se permite un salto de página
  a4 b r2 |
  R1*2 | % aquí se permite un salto de página
  a1
}
```

Al usar repeticiones de primera y segunda vez, el grabador `Page_turn_engraver` solo permite un pase de página durante la repetición si hay suficiente tiempo al principio y al final de la repetición para volver a pasar la página hacia atrás. Si la repetición es demasiado corta, se puede usar el grabador `Page_turn_engraver` para *inhabilitar* los pasos de página mediante el establecimiento de un valor adecuado para la propiedad de contexto `minimumRepeatLengthForPageTurn`. En este caso el grabador `Page_turn_engraver` sólo permitirá los pases de página en las repeticiones cuya duración sea mayor que el valor especificado.

Las instrucciones de paso de página, `\pageTurn`, `\noPageTurn` y `\allowPageTurn`, se pueden usar también en el nivel más elevado del código, dentro de elementos de marcado del nivel superior y entre las partituras.

## Instrucciones predefinidas

`\pageTurn`, `\noPageTurn`, `\allowPageTurn`.

## Véase también

Referencia de la notación: [Variables de `\paper` para los saltos de línea], página 575.

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

## Advertencias y problemas conocidos

Utilice un solo grabador `Page_turn_engraver` en cada partitura. Si hay más de uno, se interferirán entre sí.

## Véase también

Notation Reference: `<undefined>` [`<undefined>`], página `<undefined>`.

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

## 4.4 Espaciado vertical

El espaciado vertical está controlado por tres factores: la cantidad de espacio disponible (es decir, el tamaño del papel y los márgenes), la separación entre los sistemas, y la separación entre los pentagramas dentro de un sistema.

### 4.4.1 Espaciado vertical flexible dentro de los sistemas

Tres mecanismos distintos controlan el espaciado vertical flexible dentro de los sistemas, uno por cada una de las categorías siguientes:

- *pautas sin agrupar*,
- *pautas agrupadas* (pautas dentro de un grupo de pentagramas como `ChoirStaff`, etc.), y
- *líneas que no son pautas* (como `Lyrics`, `ChordNames`, etc.).

La altura de cada sistema se determina en dos fases. Primero, todos los pentagramas se disponen según la cantidad de espacio vacío disponible. Después, las líneas que no son pautas se distribuyen entre las pautas.

Observe que los mecanismos de espaciado estudiados en esta sección solamente controlan el espaciado vertical de las pautas y líneas que no son pautas dentro de los sistemas individuales. El espaciado vertical entre distintos sistemas, partituras, marcados y márgenes se controla mediante variables de `\paper` que se estudian en Sección 4.1.4 [Variables de espaciado de `\paper` verticales flexibles], página 571.

### Propiedades de espaciado dentro de los sistemas

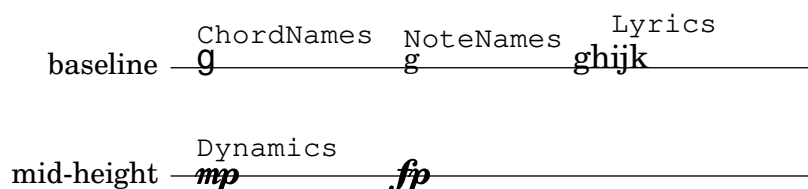
Los mecanismos de espaciado vertical dentro del sistema están controlados por dos conjuntos de propiedades de grob. El primer conjunto está asociado con el grob `VerticalAxisGroup`, que se crea por parte de todas las pautas y líneas que no son pautas. El segundo conjunto está asociado con el grob `StaffGrouper`, que puede crearse por parte de los grupos de pentagramas, pero solamente si se le llama explícitamente. Estas propiedades se describen individualmente al final de la presente sección.

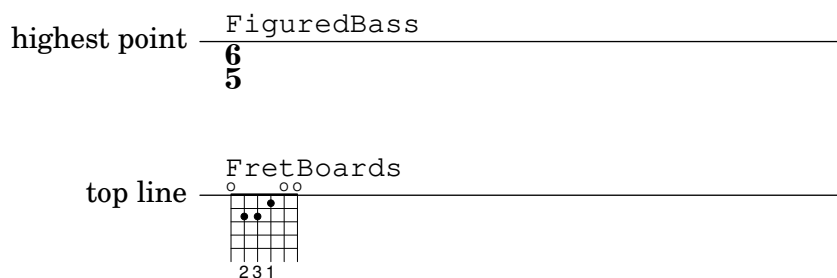
Los nombres de estas propiedades (excepto para `staff-affinity`) siguen el formato `elemento1-elemento2-spacing`, donde `elemento1` y `elemento2` son los elementos que se van a espaciar. Observe que `elemento2` no está necesariamente por debajo de `elemento1`; por ejemplo, `nonstaff-relatedstaff-spacing` mide hacia arriba a partir de la línea del tipo no-pauta si `staff-affinity` es UP.

Cada distancia se mide entre los *puntos de referencia* de los dos elementos. El *punto de referencia* para un pentagrama y otro tipo de pauta es el centro vertical de su `StaffSymbol` (es decir, la línea central si `line-count` es un número impar; el espacio central si `line-count` es par). Los puntos de referencia para las líneas individuales que no son pautas aparecen en la tabla siguiente:

Línea que no es una pauta	Punto de referencia
<code>ChordNames</code>	línea de base
<code>NoteNames</code>	línea de base
<code>Lyrics</code>	línea de base
<code>Dynamics</code>	la mitad de la altura de la ‘m’
<code>FiguredBass</code>	punto más alto
<code>FretBoards</code>	línea superior

En la siguiente imagen, las líneas horizontales indican las posiciones de estos puntos de referencia:





Cada una de las propiedades de grob del espaciado vertical (excepto `staff-affinity`) usa la misma estructura de lista-A que las variables de espaciado del bloque `\paper` que se han estudiado en Sección 4.1.4 [Variables de espaciado de `\paper` verticales flexibles], página 571. En Sección 5.3.7 [Modificación de las listas-A], página 658, se estudian métodos específicos para la modificación de las listas-A. Las propiedades de los grobs deben ajustarse con un `\override` dentro de un bloque `\score` o `\layout`, y no dentro de un bloque `\paper`.

El ejemplo que sigue muestra las dos maneras en que pueden modificarse estas listas-A. La primera declaración actualiza un par clave-valor de forma individual, y la segunda redefine la propiedad completamente:

```
\new Staff \with {
  \override VerticalAxisGroup.default-staff-staff-spacing.basic-distance = #10
} { ... }

\new Staff \with {
  \override VerticalAxisGroup.default-staff-staff-spacing =
    #'((basic-distance . 10)
      (minimum-distance . 9)
      (padding . 1)
      (stretchability . 10))
} { ... }
```

Para modificar globalmente cualquiera de los ajustes de espaciado, póngalos dentro del bloque `\layout`:

```
\layout {
  \context {
    \Staff
    \override VerticalAxisGroup.default-staff-staff-spacing.basic-distance = #10
  }
}
```

Los ajustes estándar para las propiedades de grob del espaciado vertical están relacionadas en Sección “VerticalAxisGroup” en *Referencia de Funcionamiento Interno* y en Sección “Staff-Grouper” en *Referencia de Funcionamiento Interno*. Las sobreescrituras predeterminadas para los tipos específicos de líneas que no son pautas están relacionadas en las descripciones de contexto correspondientes en Sección “Contexts” en *Referencia de Funcionamiento Interno*.

## Propiedades del grob VerticalAxisGroup

Las propiedades de `VerticalAxisGroup` se suelen ajustar con una instrucción `\override` en el nivel de `Staff` (o equivalente).

### `staff-staff-spacing`

Se utiliza para determinar la distancia entre la pauta actual y la pauta que está justo por debajo de ella dentro del mismo sistema, incluso si hay más líneas que no son pautas (tales como `Lyrics`) entre las dos pautas. No se aplica a la pauta inferior de un sistema.

Inicialmente, el valor de espaciado **staff-staff-spacing** de un grupo **VerticalAxisGroup** es una función de Scheme que aplica las propiedades del **StaffGrouper** si la pauta forma parte de un grupo, o el **default-staff-staff-spacing** de la pauta en caso contrario. Esto hace posible que las pautas estén espaciadas de forma diferente cuando están agrupadas. Para conseguir un espaciado regular sin importar el agrupamiento, esta función se puede sustituir por una lista-A de espaciado flexible, utilizando la forma de sobreescritura de redefinición completa que se ha mostrado más arriba. Si solo se especifican algunos valores en una operación de sobreescritura, los valores que faltan se toman de **default-staff-staff-spacing** (si tiene valores para ellos).

#### **default-staff-staff-spacing**

Una lista-A de espaciado flexible que define el espacio **staff-staff-spacing** que se usará para las pautas sin agrupar, a no ser que se haya sobreescrito explícitamente **staff-staff-spacing** con una instrucción **\override**.

#### **staff-affinity**

Dirección de la pauta que utilizar para el espaciado de la línea actual que no es pauta. Se puede elegir entre **UP** (arriba), **DOWN** (abajo) y **CENTER** (centro). Si es **CENTER**, la línea del tipo no-pauta se coloca equidistante entre las dos pautas contiguas a ambos lados, a no ser que lo impidan colisiones u otras restricciones de espaciado. Las líneas del tipo no-pauta adyacentes han de llevar valores de **staff-affinity** no crecientes desde la parte superior hasta la inferior, p.ej. una línea del tipo no-pauta establecida a **UP** no debería seguir inmediatamente a otra que esté establecida a **DOWN**. Las líneas del tipo no-pauta en la parte superior de un sistema deben usar **DOWN**; las de la parte inferior deben usar **UP**. El ajuste de **staff-affinity** para una pauta hace que se le trate como una línea que no es una pauta. El establecimiento de **staff-affinity** a **#f** para una línea que no es una pauta hace que se trate como una pauta. Establecer **staff-affinity** a **UP**, **CENTER**, o **DOWN** hace que el pentagrama resulte espaciado como si fuera una línea que no es una pauta.

#### **nonstaff-relatedstaff-spacing**

Distancia entre la línea del tipo no-pauta actual y la pauta más próxima en la dirección de **staff-affinity**, si no hay ninguna línea que no sea una pauta entre las dos, y **staff-affinity** es **UP** o **DOWN**. Si **staff-affinity** es **CENTER**, entonces se usa **nonstaff-relatedstaff-spacing** para las pautas más próximas *a los dos lados*, incluso aunque aparezcan otras líneas del tipo no-pauta entre la pauta actual y una de las otras. Esto significa que la colocación de una línea que no es una pauta depende tanto de las pautas que la rodean como de las líneas que no son pautas y que la rodean. Establecer la ampliabilidad **stretchability** de uno de estos tipos de espaciado a un valor más pequeño provoca que este espaciado domine. El establecimiento de **stretchability** a un valor mayor hace que dicho espaciado tenga un menor efecto.

#### **nonstaff-nonstaff-spacing**

Distancia entre la línea actual del tipo no-pauta y la siguiente línea del tipo no-pauta en la dirección de **staff-affinity**, si las dos están al mismo lado de la pauta relacionada, y **staff-affinity** es **UP** o **DOWN**.

#### **nonstaff-unrelatedstaff-spacing**

Distancia entre la línea actual del tipo no-pauta y la pauta en la dirección opuesta de **staff-affinity**, si no hay ninguna otra línea del tipo no-pauta entre las dos, y **staff-affinity** es **UP** o **DOWN**. Se puede usar, por ejemplo, para requerir una cantidad de relleno mínima entre una línea de **Lyrics** y la pauta a la que no pertenece.

## Propiedades del grob `StaffGrouper`

Las propiedades de `StaffGrouper` se suelen ajustar con una instrucción `\override` en el nivel de `StaffGroup` (o equivalente).

### `staff-staff-spacing`

Distancia entre pautas consecutivas dentro del grupo de pautas actual. La propiedad `staff-staff-spacing` del grob `VerticalAxisGroup` de una pauta individual se puede sobrescribir con distintos ajustes de espaciado para dicha pauta.

### `staffgroup-staff-spacing`

Distancia entre la última pauta del grupo en curso y la pauta que está justo por debajo de ella dentro del mismo sistema, incluso si existen una o más líneas que no son pautas (tales como `Lyrics`) entre las dos pautas. No se aplica al pentagrama o pauta inferior de un sistema. La propiedad `staff-staff-spacing` del grob `VerticalAxisGroup` de una pauta individual se puede sobrescribir con distintos ajustes de espaciado para dicha pauta.

## Véase también

Referencia de la notación: Sección 4.1.4 [Variables de espaciado de `\paper` verticales flexibles], página 571, Sección 5.3.7 [Modificación de las listas-A], página 658.

Archivos de inicio: `ly/engraver-init.ly`, `scm/define-grobs.scm`.

Referencia de funcionamiento interno: Sección “Contexts” en *Referencia de Funcionamiento Interno*, Sección “`VerticalAxisGroup`” en *Referencia de Funcionamiento Interno*, Sección “`StaffGrouper`” en *Referencia de Funcionamiento Interno*.

## Espaciado de pautas no agrupadas

Las *pautas*, tales como los pentagramas (`Staff`), pautas de percusión (`DrumStaff`) o de tablatura (`TabStaff`), etc. son contextos que pueden contener uno o más contextos de voz, pero no pueden contener otras pautas.

Las siguientes propiedades afectan al espaciado de las pautas *no agrupadas*:

- Propiedades de `VerticalAxisGroup`:
  - `default-staff-staff-spacing`
  - `staff-staff-spacing`

Estas propiedades de grob se describen individualmente más arriba; véase [Propiedades de espaciado dentro de los sistemas], página 590.

Están implicadas ciertas propiedades adicionales para las pautas que son parte de un grupo; véase [Espaciado de pautas agrupadas], página 594.

El ejemplo siguiente muestra cómo la propiedad `default-staff-staff-spacing` puede afectar al espaciado de pautas no agrupadas. Las mismas sobrescrituras aplicadas a `staff-staff-spacing` tendrían el mismo efecto, pero también se aplicaría en caso de que las pautas estuvieran combinadas en un grupo o grupos.

```
\layout {
  \context {
    \Staff
    \override VerticalAxisGroup.default-staff-staff-spacing =
      #'((basic-distance . 8)
         (minimum-distance . 7)
         (padding . 1))
  }
}
```

```
<<
% The very low note here needs more room than 'basic-distance
% can provide, so the distance between this staff and the next
% is determined by 'padding.
\new Staff { b,2 r | }

% Here, 'basic-distance provides enough room, and there is no
% need to compress the space (towards 'minimum-distance) to make
% room for anything else on the page, so the distance between
% this staff and the next is determined by 'basic-distance.
\new Staff { \clef bass g2 r | }

% By setting 'padding to a negative value, staves can be made to
% collide. The lowest acceptable value for 'basic-distance is 0.
\new Staff \with {
  \override VerticalAxisGroup.default-staff-staff-spacing =
    #'((basic-distance . 3.5)
      (padding . -10))
} { \clef bass g2 r | }
\new Staff { \clef bass g2 r | }
>>
```



## Véase también

Archivos de inicio: `scm/define-grobs.scm`.

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “VerticalAxisGroup” en *Referencia de Funcionamiento Interno*.

## Espaciado de pautas agrupadas

En partituras grandes como las orquestales, es común colocar los pentagramas en grupos. El espacio entre los grupos suele ser mayor que el espacio que hay entre los pentagramas dentro del mismo grupo.

Los grupos de pautas, *Staff-groups* (tales como `StaffGroup`, `ChoirStaff`, etc.) son contextos que pueden contener al mismo tiempo uno o más pentagramas o pautas.

Las siguientes propiedades afectan al espaciado de las pautas dentro de los grupos:

- Propiedades de `VerticalAxisGroup`:
  - `staff-staff-spacing`
- Propiedades de `StaffGroup`:
  - `staff-staff-spacing`

- **staffgroup-staff-spacing**

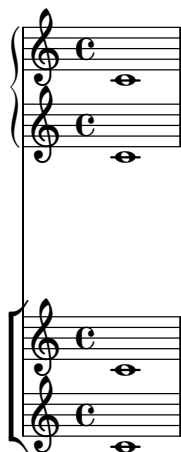
Estas propiedades de grob se describen individualmente más arriba; véase [Propiedades de espaciado dentro de los sistemas], página 590.

El ejemplo siguiente muestra cómo pueden afectar las propiedades del grob **StaffGrouper** al espaciado de las pautas agrupadas:

```
\layout {
  \context {
    \Score
    \override StaffGrouper.staff-staff-spacing.padding = #0
    \override StaffGrouper.staff-staff-spacing.basic-distance = #1
  }
}

<<
\new PianoStaff \with {
  \override StaffGrouper.staffgroup-staff-spacing.basic-distance = #20
} <<
  \new Staff { c'1 }
  \new Staff { c'1 }
>>

\new StaffGroup <<
  \new Staff { c'1 }
  \new Staff { c'1 }
>>
>>
```



## Véase también

Archivos de inicio: `scm/define-grobs.scm`.

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “VerticalAxisGroup” en *Referencia de Funcionamiento Interno*, Sección “StaffGrouper” en *Referencia de Funcionamiento Interno*.

## Espaciado de las líneas que no son pautas

Las *Líneas que no son pautas* (tales como **Lyrics**, **ChordNames**, etc.) son contextos cuyos objetos de presentación se imprimen como pentagramas (es decir, en líneas horizontales dentro de los



sistemas). Específicamente, las líneas que no son pautas son contextos del tipo no-pauta que contienen el grabador Sección “Axis\_group\_engraver” en *Referencia de Funcionamiento Interno*.

Las siguientes propiedades afectan al espaciado de las líneas que no son pautas:

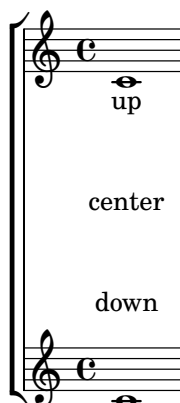
- Propiedades de `VerticalAxisGroup`:
  - `staff-affinity`
  - `nonstaff-relatedstaff-spacing`
  - `nonstaff-nonstaff-spacing`
  - `nonstaff-unrelatedstaff-spacing`

Estas propiedades de grob se describen individualmente más arriba; véase [Propiedades de espaciado dentro de los sistemas], página 590.

El ejemplo siguiente muestra cómo la propiedad `nonstaff-nonstaff-spacing` puede afectar el espaciado de líneas consecutivas que no son pautas. Aquí, mediante el establecimiento de la clave de ampliabilidad `stretchability` a un valor muy grande, la línea de letra es capaz de ampliarse mucho más de lo que es usual:

```
\layout {
  \context {
    \Lyrics
    \override VerticalAxisGroup.nonstaff-nonstaff-spacing.stretchability = #1000
  }
}

\new StaffGroup
<<
  \new Staff \with {
    \override VerticalAxisGroup.staff-staff-spacing = #'((basic-distance . 30))
  } { c'1 }
  \new Lyrics \with {
    \override VerticalAxisGroup.staff-affinity = #UP
  } \lyricmode { up }
  \new Lyrics \with {
    \override VerticalAxisGroup.staff-affinity = #CENTER
  } \lyricmode { center }
  \new Lyrics \with {
    \override VerticalAxisGroup.staff-affinity = #DOWN
  } \lyricmode { down }
  \new Staff { c'1 }
>>
```



## Véase también

Archivos de inicio: `ly/engraver-init.ly`, `scm/define-grobs.scm`.

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Contexts” en *Referencia de Funcionamiento Interno*, Sección “VerticalAxisGroup” en *Referencia de Funcionamiento Interno*.

### 4.4.2 Posicionamiento explícito de los pentagramas y los sistemas

Una forma de entender los mecanismos del espaciado vertical flexible que se han explicado más arriba es como una colección de ajustes que controlan la cantidad de relleno vertical entre los pentagramas y los sistemas.

Es posible enfrentarse al espaciado vertical de una forma distinta utilizando `NonMusicalPaperColumn.line-break-system-details`. Mientras que los mecanismos del espaciado vertical flexible especifican relleno vertical, `NonMusicalPaperColumn.line-break-system-details` puede especificar posiciones verticales exactas sobre la página.

`NonMusicalPaperColumn.line-break-system-details` acepta una lista asociativa de cuatro ajustes diferentes:

- `X-offset` (desplazamiento en X)
- `Y-offset` (desplazamiento en Y)
- `extra-offset` (desplazamiento adicional)
- `alignment-distances` (distancias de alineación)

Las sobreescrituras de los objetos gráficos, entre ellas las de `NonMusicalPaperColumn` que aparecen más abajo, pueden ocurrir en tres lugares distintos dentro de un archivo de entrada:

- directamente en medio de las notas
- en un bloque `\context`
- en el bloque `\with`

Cuando sobreescribimos `NonMusicalPaperColumn`, usamos la instrucción `\override` usual en los bloques `\context` y en el bloque `\with`. Por otra parte, cuando sobreescribimos `NonMusicalPaperColumn` en medio de las notas, debemos usar la instrucción especial `\overrideProperty`. He aquí algunas sobreescrituras de `NonMusicalPaperColumn` de ejemplo con la instrucción especial `\overrideProperty`:

```
\overrideProperty NonMusicalPaperColumn.line-break-system-details
  #'((X-offset . 20))
```

```
\overrideProperty NonMusicalPaperColumn.line-break-system-details
  #'((Y-offset . 40))
```

```
\overrideProperty NonMusicalPaperColumn.line-break-system-details
  #'((X-offset . 20)
      (Y-offset . 40))
```

```
\overrideProperty NonMusicalPaperColumn.line-break-system-details
  #'((alignment-distances . (15)))
```

```
\overrideProperty NonMusicalPaperColumn.line-break-system-details
  #'((X-offset . 20)
      (Y-offset . 40)
      (alignment-distances . (15))))
```

Para entender cómo funciona cada uno de los distintos ajustes, empezamos observando un ejemplo que no incluye absolutamente ninguna sobreescritura.

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          s1*5 \break
          s1*5 \break
          s1*5 \break
        }
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
      \new Staff {
        \repeat unfold 15 { d'4 d' d' d' }
      }
    >>
  }
}
```



Esta partitura mantiene aislada tanto la información de los saltos de línea como los de página en una voz dedicada a ello. Esta técnica de crear una voz de saltos ayuda a mantener separada de la escritura de notas conforme el ejemplo se vuelve más complicado. Consulte Sección 4.3 [Saltos], página 582.

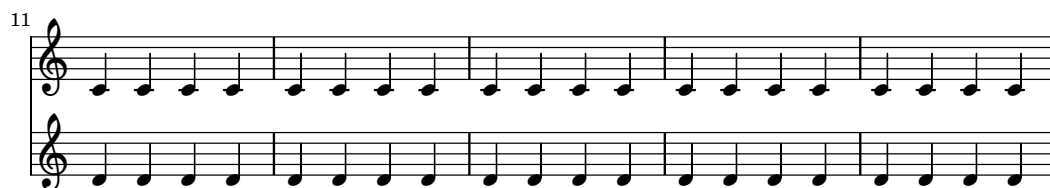
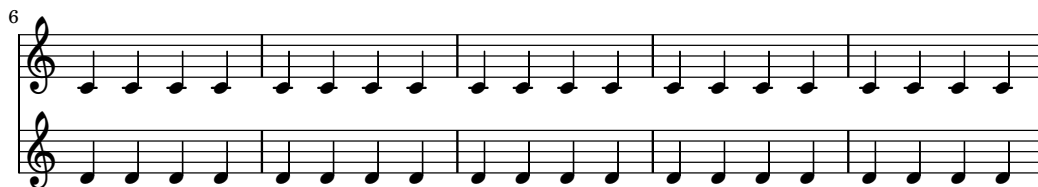
Utilizando instrucciones `\break` explícitas, la música se divide líneas de cinco compases cada una. El espaciado vertical proviene de los ajustes predeterminados de LilyPond pero el punto de origen vertical de cada sistema se fija explícitamente usando el par `Y-offset` en el atributo `line-break-system-details` del grob (objeto gráfico) `NonMusicalPaperColumn`:

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
```

```

\score {
  <<
    \new Staff <<
      \new Voice {
        \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
          #'((Y-offset . 0))
        s1*5 \break
        \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
          #'((Y-offset . 40))
        s1*5 \break
        \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
          #'((Y-offset . 60))
        s1*5 \break
      }
      \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
    >>
    \new Staff {
      \repeat unfold 15 { d'4 d' d' d' }
    }
  >>
}

```



Observe que `line-break-system-details` toma una lista asociativa de una cantidad de valores potencialmente elevada, pero aquí sólo fijamos un valor. Observe también que aquí la propiedad `Y-offset` determina la posición vertical exacta sobre la página en la que se trazará cada uno de los nuevos sistemas.

En contraste con el posicionado absoluto que está disponible a través de `Y-offset` y de `X-offset`, es posible un posicionamiento relativo con la propiedad `extra-offset` de `line-break-system-details`. La colocación es relativa a la disposición predeterminada o al posicionamiento absoluto creado mediante el establecimiento de los valores de `X-offset` y de `Y-offset`. La propiedad `extra-offset` acepta una pareja de números (un `pair`) que consiste en desplazamientos a lo largo de los ejes X e Y.

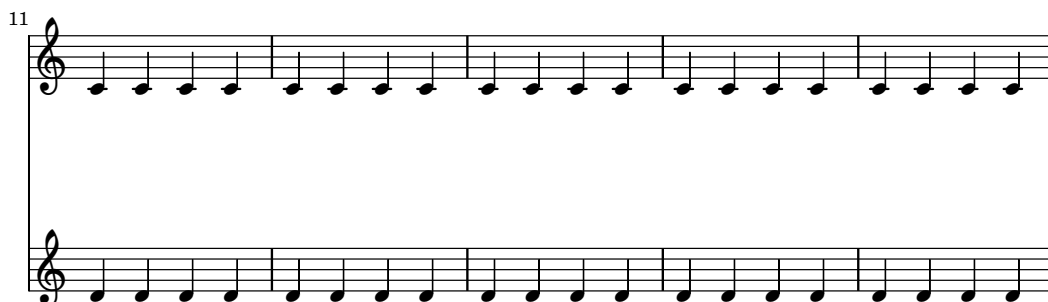
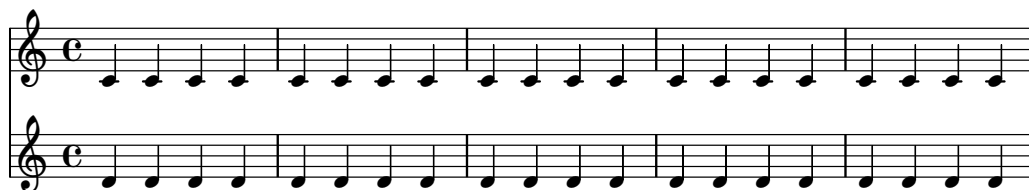
```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          s1*5 \break
          \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
            #'((extra-offset . (0 . 10)))
          s1*5 \break
          \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
            #'((extra-offset . (0 . 10)))
          s1*5 \break
        }
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
      \new Staff {
        \repeat unfold 15 { d'4 d' d' d' }
      }
    >>
  }
}
```



Ahora que hemos establecido explícitamente el punto de origen vertical de cada sistema, podemos también establecer manualmente las distancias verticales entre los pentagramas dentro de cada sistema. Lo hacemos usando la subpropiedad `alignment-distances` de `line-break-system-details`.

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
    \new Staff <<
    \new Voice {
      \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
        #'((Y-offset . 20)
          (alignment-distances . (10)))
      s1*5 \break
      \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
        #'((Y-offset . 60)
          (alignment-distances . (15)))
      s1*5 \break
      \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
        #'((Y-offset . 85)
          (alignment-distances . (20)))
      s1*5 \break
    }
    \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
  >>
  \new Staff {
    \repeat unfold 15 { d'4 d' d' d' }
  }
  >>
}
```

}



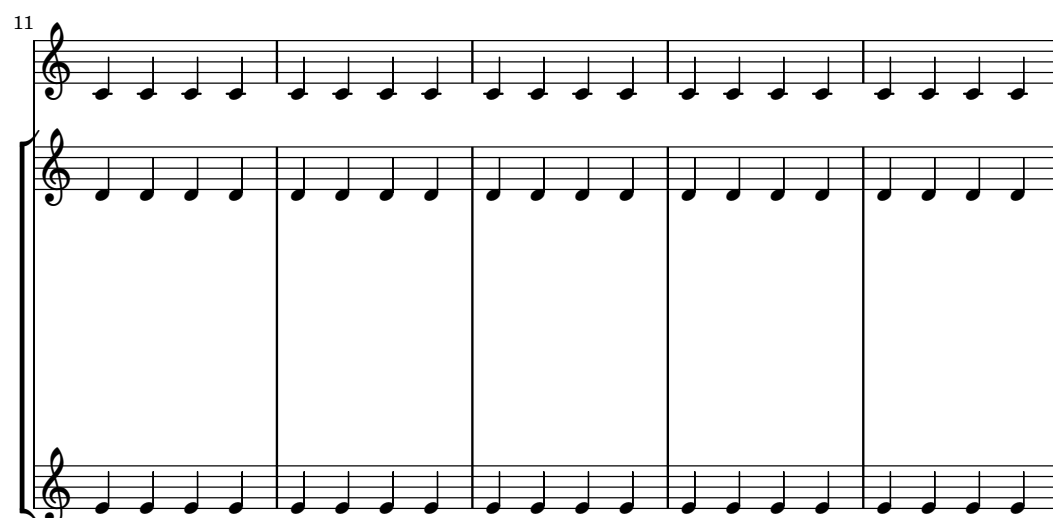
Observe que aquí asignamos dos valores distintos al atributo `line-break-system-details` del grob `NonMusicalPaperColumn`. Aunque el atributo `alist` de `line-break-system-details` acepta muchos parámetros de espaciado adicionales (entre ellos, por ejemplo, un par `X-offset` correspondiente), sólo tenemos que establecer el desplazamiento `Y-offset` y los pares `alignment-distances` para controlar el punto de origen vertical de cada sistema y pentagrama. Finalmente, observe que `alignment-distances` especifica el posicionamiento vertical de los pentagramas pero no de los grupos de pentagramas.

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
    \new Staff <<
    \new Voice {
      \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
        #'((Y-offset . 0)
          (alignment-distances . (30 10)))
      s1*5 \break
      \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
        #'((Y-offset . 60)
          (alignment-distances . (10 10)))
```

```
s1*5 \break
\overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
  #'((Y-offset . 100)
    (alignment-distances . (10 30)))
s1*5 \break
}
\new Voice { \repeat unfold 15 { c'4 c' c' c' } }
>>
\new StaffGroup <<
  \new Staff { \repeat unfold 15 { d'4 d' d' d' } }
  \new Staff { \repeat unfold 15 { e'4 e' e' e' } }
>>
>>
}
```



}



Algunos puntos que tener en cuenta:

- Al usar `alignment-distances`, la letra y otras líneas que no son pautas, no cuentan como un pentagrama.

- Las unidades de los números que se pasan a `X-offset`, `Y-offset`, `extra-offset` y `alignment-distances` se interpretan como múltiplos de la distancia entre líneas de pentagrama adyacentes. Los valores positivos mueven a los pentagramas y a la letra hacia arriba, los valores negativos mueven los pentagramas y la letra hacia abajo.
- A causa de que los ajustes a `NonMusicalPaperColumn.line-break-system-details` dados aquí permiten el posicionamiento de pentagramas y sistemas en cualquier lugar de la página, es posible violar los márgenes o los límites del papel, o incluso imprimir pentagramas o sistemas unos encima de otros. Esto se evitaría pasando a los diferentes ajustes unos valores razonables.

## Véase también

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

### 4.4.3 Evitar las colisiones verticales

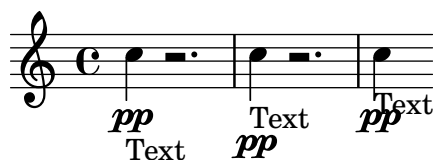
Podemos decir intuitivamente que algunos objetos de la notación musical pertenecen al pentagrama y otros se sitúan fuera del pentagrama. Entre los objetos que pertenecen al exterior del pentagrama están las marcas de ensayo, las marcas textuales y las indicaciones de dinámica (en lo sucesivo, les llamaremos objetos fuera del pentagrama). La regla de LilyPond para la colocación vertical de los objetos fuera del pentagrama es ponerlos tan cerca del pentagrama como sea posible pero no tan próximos como para que choquen con otro objeto.

LilyPond utiliza la propiedad `outside-staff-priority` para determinar si un grob es un objeto fuera del pentagrama: si `outside-staff-priority` es un número, el grob es un objeto fuera del pentagrama. Además, `outside-staff-priority` informa a LilyPond en qué orden se debe situar los objetos.

En primer lugar, LilyPond sitúa todos los objetos que no pertenecen al exterior del pentagrama. Después ordena los objetos fuera del pentagrama de acuerdo a su prioridad `outside-staff-priority` (en orden creciente). LilyPond toma los objetos fuera del pentagrama uno a uno y los coloca de forma que no choquen con ningún objeto que ya haya sido colocado. Esto es, si dos grobs fuera del pentagrama compiten por el mismo espacio, el que tiene la prioridad `outside-staff-priority` más baja se colocará más próximo al pentagrama.

Hay un listado completo de las prioridades `outside-staff-priorities` en Sección “La propiedad `outside-staff-priority` (prioridad fuera del pentagrama)” en *Manual de Aprendizaje*.

```
\relative c'' {
  c4_"Text"\pp
  r2.
  \once \override TextScript.outside-staff-priority = #1
  c4_"Text"\pp % this time the text will be closer to the staff
  r2.
  % by setting outside-staff-priority to a non-number,
  % we disable the automatic collision avoidance
  \once \override TextScript.outside-staff-priority = ##f
  \once \override DynamicLineSpanner.outside-staff-priority = ##f
  c4_"Text"\pp % now they will collide
}
```



El relleno vertical que rodea a los objetos fuera del pentagrama se puede controlar con `outside-staff-padding`.

```
\relative {
  \once \override TextScript.outside-staff-padding = #0
  a'4-"outside-staff-padding = #0"
  \once \override TextScript.outside-staff-padding = #3
  d-"outside-staff-padding = #3"
  c-"default outside-staff-padding"
  b-"default outside-staff-padding"
  R1
}
```

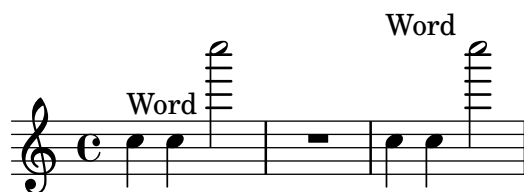


`outside-staff-padding = #3`

`default outside-staff-padding`  
`default outside-staff-padding`

De forma predeterminada, los objetos fuera del pentagrama se colocan de forma que eviten una colisión horizontal con los grobs posicionados previamente. Esto puede llevar a situaciones en las que los objetos se colocan muy próximos entre sí en el sentido horizontal. Como se muestra en el ejemplo que aparece más abajo, al establecer `outside-staff-horizontal-padding` se incrementa el espacio horizontal necesario, y en este caso mueve el texto hacia arriba para evitar que quede demasiado cerca de las líneas adicionales.

```
\relative {
  c'4~"Word" c c''2
  R1
  \once \override TextScript.outside-staff-horizontal-padding = #1
  c,,4~"Word" c c''2
}
```



## Véase también

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

## 4.5 Espaciado horizontal

### 4.5.1 Panorámica del espaciado horizontal

El motor de espaciado traduce las diferencias en las duraciones a distancias ampliables (‘muelles’) de distintas longitudes. Las duraciones más largas reciben un espacio mayor y las duraciones más cortas reciben menos. Las duraciones más breves reciben un espacio de tamaño fijo (que se controla mediante `shortest-duration-space` en el objeto Sección “SpacingSpanner” en *Referencia de Funcionamiento Interno*). Cuanto más larga es la duración, más espacio recibe: al doblar una duración se suma `spacing-increment` al espacio que ocupa la nota.

Por ejemplo, la pieza siguiente contiene numerosas blancas, negras y corcheas; la corchea va seguida de un ancho de cabeza de blanca (ACB). La negra va seguida de 2 ACB, la blanca por 3 ACB, etc.

```
\relative c' {
  c2 c4. c8
  c4. c8 c4. c8
  c8 c c4 c c
}
```



Normalmente, el valor de `spacing-increment` está establecido en 1.2 espacios de pentagrama, lo que es aproximadamente el ancho de una cabeza, y `shortest-duration-space` está establecido en 2.0, lo que significa que la nota más breve recibe 2.4 espacios de pentagrama (2.0 multiplicado por el `spacing-increment`) de espacio horizontal. Este espacio se cuenta a partir del borde izquierdo del símbolo, de manera que las notas más breves van seguidas generalmente por un espacio de 1 ACB.

Si siguiésemos el procedimiento anterior exactamente, entonces la adición de una sola fusa a una partitura que usa corcheas y semicorcheas, aumentaría enormemente la anchura de la partitura completa. La nota más breve ya no es la semicorchea, sino la fusa, añadiendo así 1 ACB a cada una de las notas. Para evitarlo, la duración más breve a efectos de espaciado no es la nota más breve de la partitura, sino la que ocurre con más frecuencia.

La duración más breve que es más común se determina de la siguiente manera: en cada compás se determina la duración más breve. La menor duración más común se toma como base para el espaciado, habiendo estipulado que esta duración menor siempre debe ser igual o menor que una corchea. La duración más breve se imprime cuando se ejecuta `lilypond` con la opción `--verbose`.

Estas duraciones también se pueden personalizar. Si establecemos la `common-shortest-duration` en Sección “SpacingSpanner” en *Referencia de Funcionamiento Interno*, entonces éste establece la duración base para el espaciado. La duración máxima para esta base (normalmente una corchea), se fija a través de `base-shortest-duration`.

Las notas que son aún más breves que la nota común más breve van seguidas por un espacio proporcional a su duración en relación con la nota común más breve. Así pues, si fuésemos a añadir tan sólo algunas semicorcheas al ejemplo anterior, irían seguidas por medio ACB:

```
\relative { c''2 c4. c8 | c4. c16[ c] c4. c8 | c8 c c4 c c }
```



Como se explicó en el *Ensayo sobre grabado musical automatizado*, las direcciones de las plicas influyen en el espaciado (véase Sección “Espaciado óptico” en *Ensayo*) y se puede ajustar con la propiedad `stem-spacing-correction` del objeto Sección “NoteSpacing” en *Referencia de Funcionamiento Interno*. Estos se generan para cada uno de los contextos de Sección “Voice” en *Referencia de Funcionamiento Interno*.

El objeto `StaffSpacing` (generado en el contexto de Sección “Staff” en *Referencia de Funcionamiento Interno*) contiene la misma propiedad para controlar el espaciado de las líneas de plica o divisorias.



```

\time 4/16
c16[ c c8]
\newSpacingSection
\override Score.SpacingSpanner.spacing-increment = #2
c16[ c c8]
\newSpacingSection
\revert Score.SpacingSpanner.spacing-increment
c16[ c c8]
}

```



## Véase también

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “SpacingSpanner” en *Referencia de Funcionamiento Interno*.

### 4.5.3 Cambiar el espaciado horizontal

Se puede alterar el espaciado horizontal con la propiedad `base-shortest-duration`. Aquí compararemos la misma música, una vez sin alterar la propiedad, y luego alterándola. Los valores mayores de `ly:make-moment` producen música más pequeña. Observe que `ly:make-moment` construye una duración, por lo que 1 4 es una duración mayor que 1 16.

```

\score {
  \relative {
    g'4 e e2 | f4 d d2 | c4 d e f | g4 g g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
    d4 d d d | d4 e f2 | e4 e e e | e4 f g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
  }
}

```



```

\score {
  \relative {
    g'4 e e2 | f4 d d2 | c4 d e f | g4 g g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
    d4 d d d | d4 e f2 | e4 e e e | e4 f g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
  }
}

```

```

}
\layout {
  \context {
    \Score
    \override SpacingSpanner.base-shortest-duration = #(ly:make-moment 1/16)
  }
}
}

```



### Estiramiento uniforme de los grupos especiales

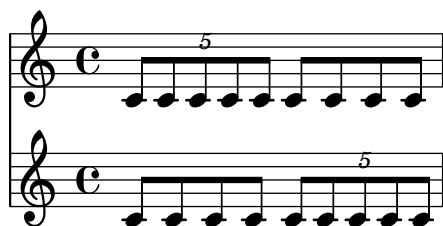
De forma predeterminada, el espaciado en los grupos de valoración especial depende de varios factores ajenos a la duración (como alteraciones, cambios de clave, etc.). Para pasar por alto estos símbolos y forzar un espaciado de duraciones iguales uniforme, use `Score.SpacingSpanner.uniform-stretching`. Esta propiedad sólo puede cambiarse al principio de la partitura:

```

\score {
  <<
    \new Staff \relative c' {
      \tuplet 5/4 { c8 c c c c } c8 c c c
    }
    \new Staff \relative c' {
      c8 c c c \tuplet 5/4 { c8 c c c c }
    }
  >>
  \layout {
    \context {
      \Score
      \override SpacingSpanner.uniform-stretching = ##t
    }
  }
}

```

}



### Espaciado estricto de las notas

Cuando se establece `strict-note-spacing`, las notas se separan sin tener en cuenta las claves, líneas divisorias ni notas de adorno:

```
\override Score.SpacingSpanner.strict-note-spacing = ##t
\new Staff \relative {
  c' '8[ c \clef alto c \grace { c16 c } c8 c c] c32[ c] }
```



### Véase también

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

#### 4.5.4 Longitud de la línea

Los ajustes más básicos que influyen en el espaciado son `indent` y `line-width`. Se definen en el bloque `\layout`. Controlan el sangrado de la primera línea de música y la longitud de las líneas.

Si se fija un valor verdadero para `ragged-right` en el bloque `\layout`, entonces los sistemas terminan en su longitud horizontal natural, en lugar de repartirse horizontalmente hasta llenar toda la línea. Esto es útil para fragmentos cortos, y para comprobar qué tan apretado es el espaciado natural. El valor normal predeterminado es falso, pero si la partitura tiene un solo sistema, el valor predeterminado es verdadero.

La opción `ragged-last` es similar a `ragged-right`, pero afecta sólo a la última línea de la pieza. No se efectúa ninguna restricción sobre dicha línea. El resultado es similar al formateo de los párrafos de texto. En un párrafo, la última línea sencillamente ocupa su longitud horizontal natural.

```
\layout {
  indent = #0
  line-width = #150
  ragged-last = ##t
}
```

### Véase también

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

#### 4.5.5 Notación proporcional

LilyPond contempla la notación proporcional, un tipo de espaciado horizontal en el que cada nota consume una medida horizontal que equivale exactamente a su duración rítmica. Este tipo de espaciado proporcional es comparable al espaciado horizontal hecho sobre un papel milimetrado. Ciertas partituras del finales del s.XX y principios del s.XXI utilizan notación proporcional para



clarificar relaciones rítmicas complejas o para facilitar la colocación de líneas cronométricas u otros gráficos directamente en la partitura.

LilyPond contempla cinco ajustes distintos para la notación proporcional, que se pueden usar solos o combinados:

- `proportionalNotationDuration`
- `uniform-stretching`
- `strict-note-spacing`
- `\remove "Separating_line_group_engraver"`
- `\override PaperColumn.used = ##t`

En los ejemplos que siguen, exploramos el uso de estos cinco ajustes de la notación proporcional y examinamos la forma en que interactúan.

Comenzamos con el siguiente ejemplo de un solo compás, que usa un espaciado clásico sin justificación por la derecha.

```
\score {
  <<
    \new RhythmicStaff {
      c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
    }
  >>
}
```

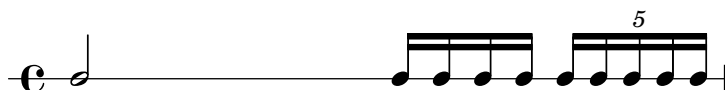


Observe que la blanca que inicia el compás ocupa mucho menos de la mitad de todo el espacio horizontal del compás. De forma similar, las semicorcheas y el cinquillo de semicorcheas con que finaliza el compás ocupan en conjunto mucho más de la mitad de todo el espacio horizontal del compás.

En el grabado clásico, este espaciado puede ser exactamente el que deseamos porque podemos tomar prestado el espacio horizontal de la blanca y conservar el espacio horizontal a lo largo del compás como un todo.

Por otro lado, si queremos insertar una línea de tiempo graduada o algún otro gráfico encima o debajo de la partitura, necesitamos la notación proporcional. Se activa la notación proporcional con el ajuste `proportionalNotationDuration`.

```
\score {
  <<
    \new RhythmicStaff {
      c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1/20)
    }
  }
}
```



La blanca al principio del compás y las notas rápidas de la segunda mitad del compás ocupan ahora cantidades iguales de espacio horizontal. Podríamos colocar una línea de tiempo graduada o un gráfico encima o debajo de este ejemplo.

El ajuste `proportionalNotationDuration` es un ajuste de contexto que reside en `Score`. Recordemos que los ajustes de contexto aparecen en uno de tres posibles lugares del archivo de entrada: en un bloque `\with`, en un bloque `\context`, o directamente entre la música precedido por la instrucción `\set`. Como con todos los ajustes de contexto, el usuario puede elegir en cuál de los tres lugares diferentes prefiere establecer el valor de `proportionalNotationDuration`.

El ajuste `proportionalNotationDuration` acepta un solo argumento, que es la duración de referencia contra el que se aplica el espaciado de toda la música. La función Scheme de LilyPond `make-moment` acepta dos argumentos: un numerador y un denominador que, juntos, expresan una cierta fracción de redonda. La llamada `(ly:make-moment 1/20)`, por tanto, produce una duración de referencia de una nota de un veinteavo de redonda (semicorcheas de cinquillo). Son también posibles valores como `(ly:make-moment 1/16)`, `(ly:make-moment 1/8)` y `(ly:make-moment 3/97)`.

¿Cómo seleccionamos la duración correcta de referencia para pasarla a `proportionalNotationDuration`? Normalmente mediante un proceso de ensayo y error, comenzando con una duración cercana a la más rápida (o más breve) duración de la pieza. Las duraciones de referencia más pequeñas aplican un espaciado más suelto; las duraciones de referencia más largas aplican un espaciado más apretado.

```
\score {
  <<
    \new RhythmicStaff {
      c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1/8)
    }
  }
}
```

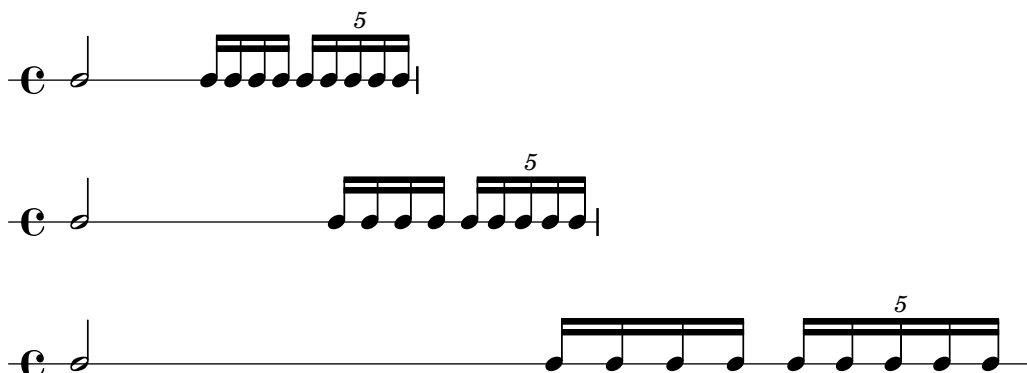
```
\score {
  <<
    \new RhythmicStaff {
      c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1/16)
    }
  }
}
```

```
\score {
  <<
    \new RhythmicStaff {
```

```

        c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
    }
>>
\layout {
  \context {
    \Score
    proportionalNotationDuration = #(ly:make-moment 1/32)
  }
}
}

```



Observe que una duración de referencia demasiado grande (como la corchea, en el ejemplo de arriba) produce un espaciado excesivamente apretado y puede ser causa de colisiones entre las cabezas de nota. Observe también que la notación proporcional en general ocupa más espacio horizontal que el espaciado clásico. El espaciado proporcional aporta claridad rítmica a expensas del espacio horizontal.

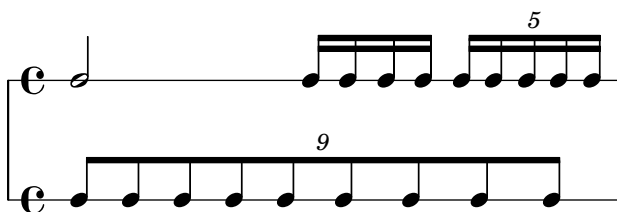
Ahora veremos cómo espaciar de forma óptima grupos de valoración especial que se superponen.

Empezamos por examinar qué le ocurre a nuestro ejemplo original, con espaciado clásico, cuando añadimos un segundo pentagrama con un tipo diferente de grupo especial.

```

\score {
  <<
    \new RhythmicStaff {
      c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
    }
    \new RhythmicStaff {
      \tuplet 9/8 { c8 8 8 8 8 8 8 8 8 }
    }
  >>
}

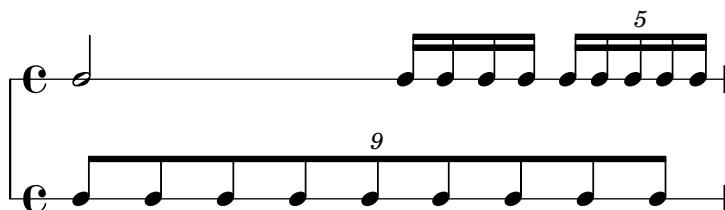
```



El espaciado es defectuoso porque las notas regularmente espaciadas del pentagrama inferior no se amplían de manera uniforme. Los grabados clásicos incluyen muy pocos tresillos complejos

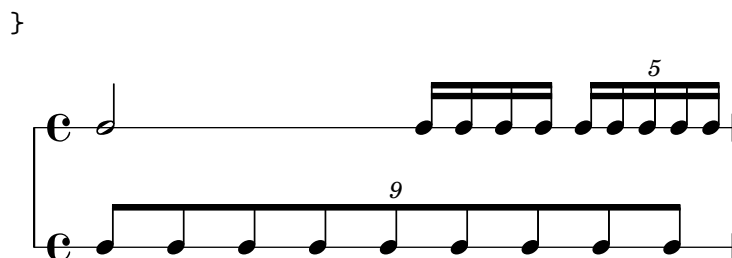
y así las reglas del grabado clásico pueden generar este tipo de resultado. El establecimiento de `proportionalNotationDuration` lo soluciona.

```
\score {
  <<
    \new RhythmicStaff {
      c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
    }
    \new RhythmicStaff {
      \tuplet 9/8 { c8 8 8 8 8 8 8 8 8 }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1/20)
    }
  }
}
```



Pero si observamos con mucho cuidado podremos ver que las notas de la segunda mitad del 9-illo están espaciadas de forma ligeramente más ancha que las de la primera mitad del 9-illo. Para asegurar una ampliación uniforme, activamos `uniform-stretching`, que es una propiedad de `SpacingSpanner`.

```
\score {
  <<
    \new RhythmicStaff {
      c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
    }
    \new RhythmicStaff {
      \tuplet 9/8 { c8 8 8 8 8 8 8 8 8 }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1/20)
      \override SpacingSpanner.uniform-stretching = ##t
    }
  }
}
```



Nuestro ejemplo de dos pentagramas ahora está espaciado exactamente, nuestras relaciones rítmicas son visualmente claras, y podemos incluir una línea de tiempo graduada o un gráfico, si queremos.

Observe que el paquete de notación proporcional de LilyPond espera que todas las partituras proporcionales establezcan el atributo `uniform-stretching` de `SpacingSpanner` al valor `##t`. El establecimiento de `proportionalNotationDuration` sin ajustar también el atributo `uniform-stretching` de `SpacingSpanner` al valor `##t` causará, por ejemplo, que los desplazamientos (`skips`) consuman una cantidad de espacio horizontal incorrecta.

El `SpacingSpanner` es un grob abstracto que reside en el contexto de `Score`. Como con nuestros ajustes de `proportionalNotationDuration`, las sobreescrituras al `SpacingSpanner` sólo pueden ocurrir en uno de estos tres lugares dentro del archivo de entrada: en el bloque `\with`, en el bloque `\context` o directamente dentro de la escritura de notas.

De manera predeterminada, sólo hay un `SpacingSpanner` por `Score`. Esto supone que, por omisión, `uniform-stretching` está activado para la partitura completa o desactivado para la partitura completa. Sin embargo, podemos sobreescribir este comportamiento y activar distintas posibilidades de espaciado en distintos lugares de la partitura. Lo hacemos con la instrucción `\newSpacingSection`. Consulte Sección 4.5.2 [Sección de espaciado nueva], página 608, para más información.

A continuación examinamos los efectos del grabador `Separating_line_group_engraver` y veremos por qué las partituras proporcionales con frecuencia eliminan este grabador. El ejemplo siguiente muestra que hay una pequeña cantidad de espacio “preliminar” justo antes de la primera nota de cada sistema.

```
\paper {
  indent = #0
}
```

```
\new Staff {
  c'1
  \break
  c'1
}
```



Esta cantidad de espacio preliminar es la misma ya sea después de una indicación de compás, una armadura o una clave. El grabador `Separating_line_group_engraver` es responsable de este espacio. La eliminación de `Separating_line_group_engraver` reduce este espacio a cero.

```
\paper {
```

```

    indent = #0
}

\new Staff \with {
  \remove "Separating_line_group_engraver"
} {
  c'1
  \break
  c'1
}

```



Los elementos no musicales como la indicación de compás, la armadura, la clave y las alteraciones son problemáticos en notación proporcional. Ninguno de estos elementos tiene duración rítmica. Pero todos ellos consumen espacio horizontal. Las distintas partituras proporcionales abordan este problema de distinta manera.

Sería posible evitar los problemas de espaciado con las armaduras, simplemente evitando tenerlas. Esta es una opción válida pues casi todas las partituras proporcionales son música contemporánea. Lo mismo puede valer para las indicaciones de compás, especialmente para las partituras que incluyen una línea de tiempo graduada u otro gráfico. Pero estas partituras son excepcionales y casi todas las partituras proporcionales incluyen al menos unas pocas indicaciones de compás. Las claves y las alteraciones son aún más esenciales.

Así pues ¿qué estrategias existen para el espaciado de los elementos no musicales en un contexto proporcional? Una buena opción es la propiedad `strict-note-spacing` de `SpacingSpanner`. Compare las dos partituras siguientes:

```

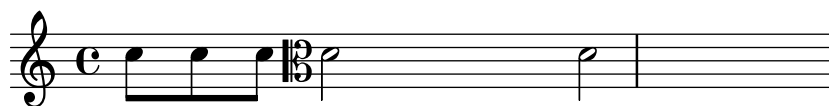
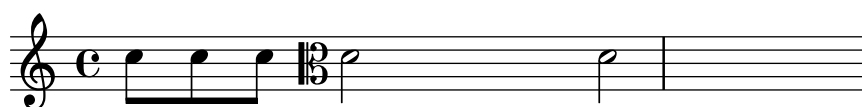
\new Staff {
  \set Score.proportionalNotationDuration = #(ly:make-moment 1/16)
  c''8 8 8 \clef alto d'2 2
}

```

```

\new Staff {
  \set Score.proportionalNotationDuration = #(ly:make-moment 1/16)
  \override Score.SpacingSpanner.strict-note-spacing = ##t
  c''8 8 8 \clef alto d'2 2
}

```



Las dos partituras son proporcionales, pero el espaciado de la primera es muy suelto a causa del cambio de clave. Sin embargo, el espaciado de la segunda partitura se mantiene estricto, porque `strict-note-spacing` está activado. La activación de `strict-note-spacing` hace que el ancho de las indicaciones de compás, armaduras, cambios de clave y alteraciones no tomen parte en el algoritmo de espaciado.

Además de los ajustes dados aquí, hay otros que aparecen con frecuencia en las partituras proporcionales. Entre ellos están:

- `\override SpacingSpanner.strict-grace-spacing = ##t`
- `\set tupletFullLength = ##t`
- `\override Beam.breakable = ##t`
- `\override Glissando.breakable = ##t`
- `\override TextSpanner.breakable = ##t`
- `\remove "Forbid_line_break_engraver" in the Voice context`

Estos ajustes dan a las notas de adorno un espaciado estricto, extienden los corchetes de grupo especial para que marquen tanto los puntos de comienzo como de final, y permiten que los elementos de extensión se dividan entre los sistemas y las páginas. Consulte las partes respectivas del manual para ver estos ajustes relacionados.

## Véase también

Referencia de la notación: Sección 4.5.2 [Sección de espaciado nueva], página 608.

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

## 4.6 Encajar la música en menos páginas

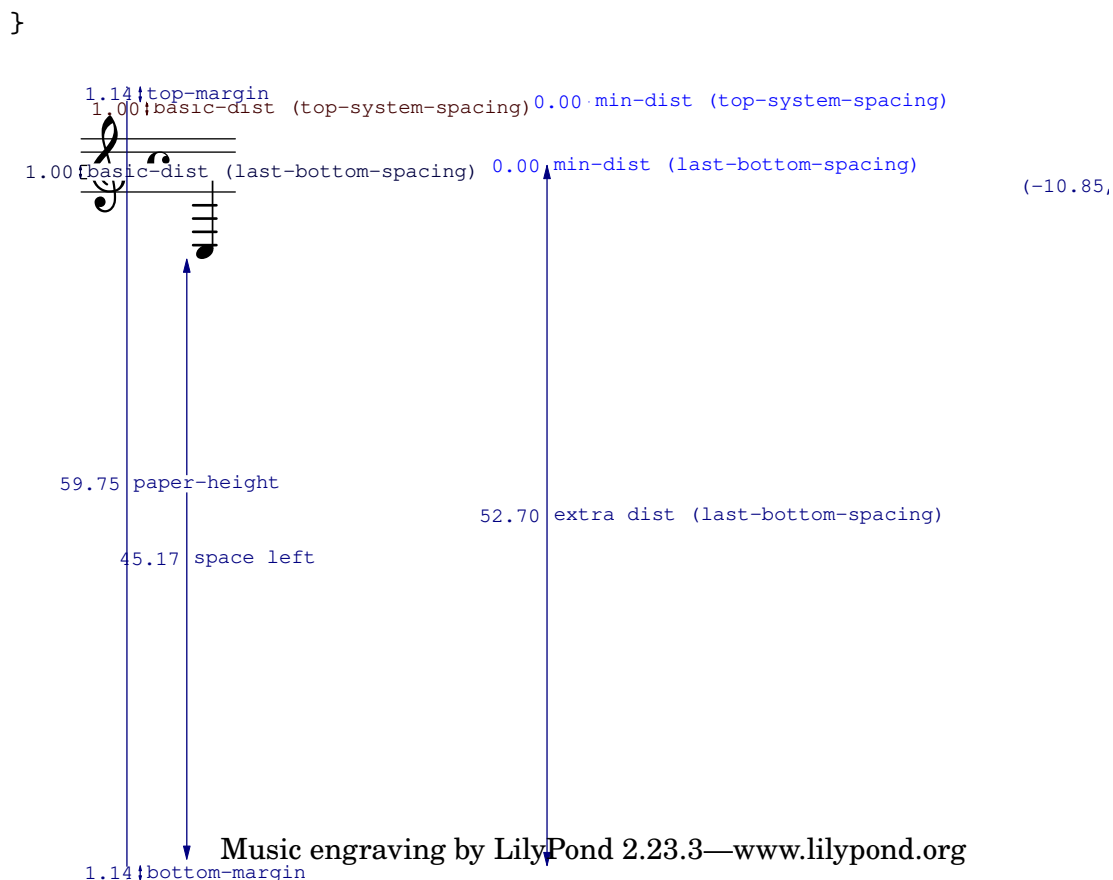
En ocasiones, podemos terminar con uno o dos pentagramas en una segunda página (o tercera, o cuarta. . .). Es fastidioso, especialmente cuando vemos que las páginas anteriores parecen tener espacio de sobra.

Al investigar los problemas de disposición, una herramienta de valor incalculable es `annotate-spacing`. Esta instrucción imprime los valores de un cierto número de variables de espaciado; para ver más detalles consulte la sección siguiente, Sección 4.6.1 [Mostrar el espaciado], página 618.

### 4.6.1 Mostrar el espaciado

Para presentar gráficamente las dimensiones de las variables de disposición vertical que pueden verse alteradas por el formato de la página, establezca `annotate-spacing` en el bloque `\paper`:

```
\book {
  \score { { c4 } }
  \paper { annotate-spacing = ##t }
```



Todas las dimensiones de disposición se muestran en espacios de pentagrama, independientemente de las unidades especificadas en los bloques `\paper` o `\layout`. En el ejemplo anterior, `paper-height` tiene un valor de 59.75 `staff-spaces` (espacios de pentagrama), y el `staff-size` (tamaño del pentagrama) es de 20 puntos, el valor predeterminado. Observe que:

$$\begin{aligned}
 1 \text{ punto} &= (25.4/72.27) \text{ mm} \\
 1 \text{ staff-space} &= (\text{staff-size})/4 \text{ pts} \\
 &= (\text{staff-size})/4 * \\
 &= (25.4/72.27) \text{ mm}
 \end{aligned}$$

En este caso, un `staff-space` equivale aproximadamente a 1.757mm. Así, la medida `paper-height` de 59.75 `staff-spaces` equivale a 105 milímetros, la altura de una hoja **a6** en orientación apaisada. Las parejas  $(a,b)$  son intervalos, donde  $a$  es el límite inferior y  $b$  es el límite superior del intervalo.

## Véase también

Referencia de la notación: Sección 4.2.2 [Establecer el tamaño del pentagrama], página 581.

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

## 4.6.2 Cambiar el espaciado

La salida de `annotate-spacing` revela las dimensiones verticales con gran detalle. Para ver más detalles acerca de la modificación de los márgenes y otras variables de diseño de la página, consulte Sección 4.1 [Disposición de la página], página 567.



Aparte de los márgenes, hay algunas otras opciones para ahorrar espacio:

- Forzar los sistemas para que se junten lo más posible entre sí (de forma que quepan la mayor cantidad de ellos en la página) y al mismo tiempo estén espaciados de forma que no exista un espacio vacío en la parte baja del papel.

```
\paper {
  system-system-spacing = #'((basic-distance . 0.1) (padding . 0))
  ragged-last-bottom = ##f
  ragged-bottom = ##f
}
```

- Forzar el número de sistemas. Puede ser de utilidad de dos formas. Si tan sólo se establece un valor, incluso el mismo valor que el número de sistemas que se están trazando de forma predeterminada, ello ocasionará a veces que se traten de introducir más sistemas en cada página, pues en este caso se omite un paso en que se produce una estimación, dando así un encaje mejor en cada página. Asimismo, el forzar una reducción neta en el número de sistemas puede producir el ahorro de una página adicional. Por ejemplo, si la disposición predeterminada tiene 11 sistemas, la asignación siguiente fuerza una disposición con 10 sistemas.

```
\paper {
  system-count = #10
}
```

- Forzar el número de páginas. Por ejemplo, la asignación siguiente fuerza una disposición con dos páginas.

```
\paper {
  page-count = #2
}
```

- Evitar (o reducir en número) los objetos que aumentan el tamaño vertical de un sistema. Por ejemplo, los corchetes de las repeticiones de primera y segunda vez (o repeticiones con finales alternativos) necesitan espacio adicional. Si estas casillas se reparten a lo largo de dos sistemas, ocupan más espacio que si estuvieran sobre el mismo sistema. Como otro ejemplo, las indicaciones dinámicas que se ‘salen’ de un sistema se pueden acercar al pentagrama:

```
\relative e' {
  e4 c g\ff c
  e4 c g-\tweak X-offset #-2.7 \ff c
}
```



- Alterar el espaciado horizontal por medio de **SpacingSpanner**. Para ver más detalles, consulte Sección 4.5.3 [Cambiar el espaciado horizontal], página 609. El ejemplo siguiente ilustra el espaciado predeterminado:

```
\score {
  \relative {
    g'4 e e2 |
    f4 d d2 |
    c4 d e f |
    g4 g g2 |
    g4 e e2 |
  }
}
```



El ejemplo siguiente modifica `common-shortest-duration` de un valor de 1/4 (negra) a 1/2 (blanca). La negra es la nota más común y la más breve del ejemplo, por lo que al agrandar esta duración se produce un efecto de ‘apretujamiento’:

```
\score {
  \relative {
    g'4 e e2 |
    f4 d d2 |
    c4 d e f |
    g4 g g2 |
    g4 e e2 |
  }
  \layout {
    \context {
      \Score
      \override SpacingSpanner.common-shortest-duration =
        #(ly:make-moment 1/2)
    }
  }
}
```



La propiedad `common-shortest-duration` no se puede modificar de manera dinámica, por lo que se debe situar siempre dentro de un bloque `\context` de forma que se aplique a la partitura completa.

## Véase también

Referencia de la notación: Sección 4.1 [Disposición de la página], página 567, Sección 4.5.3 [Cambiar el espaciado horizontal], página 609.

Fragmentos de código: Sección “Spacing” en *Fragmentos de código*.

## 5 Cambiar los valores por omisión

El objetivo del diseño de LilyPond es proporcionar la más alta calidad de los resultados, de forma predeterminada. A pesar de ello, podría tener que cambiar este resultado predeterminado. La disposición sobre el papel se controla a través de un amplio número de ‘botones e interruptores’ llamados en su conjunto ‘propiedades’. En el Manual de aprendizaje podemos encontrar una introducción en forma de tutorial al acceso y modificación de estas propiedades, véase Sección “Trucar la salida” en *Manual de Aprendizaje*. Éste debería leerse en primer lugar. Este capítulo cubre un terreno similar, pero con un estilo más adecuado para un manual de referencia.

La descripción definitiva de los controles que están disponibles para su ajuste fino están en un documento aparte: la Sección “Referencia de funcionamiento interno” en *Referencia de Funcionamiento Interno*. Dicho manual relaciona todas las variables, funciones y opciones que se encuentran disponibles en LilyPond. Está escrito como un documento HTML, que se puede encontrar en línea (<http://lilypond.org/doc/stable/Documentation/internals/>), pero que también va incluido en el paquete de la documentación de LilyPond.

Internamente, LilyPond utiliza el lenguaje Scheme (un dialecto de LISP) para aportar la infraestructura. La sobreescritura de las decisiones de disposición da acceso efectivo a las interioridades del programa, lo que requiere código de Scheme como entrada. Los elementos de Scheme se inauguran dentro de un archivo `.ly` con el símbolo de cuadrado #.<sup>1</sup>

### 5.1 Contextos de interpretación

Esta sección explica qué son los contextos y cómo modificarlos.

#### Véase también

Manual de aprendizaje: Sección “Contextos y grabadores” en *Manual de Aprendizaje*.

Archivos de inicio: `ly/engraver-init.ly`, `ly/performer-init.ly`.

Fragmentos de código: Sección “Contexts and engravers” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “Contexts” en *Referencia de Funcionamiento Interno*, Sección “Engravers and Performers” en *Referencia de Funcionamiento Interno*.

#### 5.1.1 Explicación de los contextos

Los contextos se disponen de forma jerárquica:

#### Definiciones de salida - estructura de los contextos

Esta sección explica la relevancia de las definiciones de salida cuando se trabaja con los contextos. Más adelante ofrecemos ejemplos de definiciones de salida reales (véase [Cambiar todos los contextos del mismo tipo], página 633).

Aunque la música escrita en un archivo haga referencia a tipos y nombres de contexto, los contextos se crean solamente cuando la música se está realmente interpretando. LilyPond interpreta la música bajo el control de una ‘definición de salida’ y puede hacerlo así para varias definiciones de salida distintas, dando como resultado distintas salidas también. La definición de salida que corresponde a la impresión de la música se especifica mediante `\layout`.

Una definición de salida mucho más sencilla que se usa para la producción de MIDI se especifica mediante `\midi`. Internamente, LilyPond utiliza varias otras definiciones de salida, como cuando se usa el combinador de particellas ([Combinación automática de las partes], página 192) o se crean fragmentos de notas guía citadas ([Citar otras voces], página 224).

<sup>1</sup> Sección “Tutorial de Scheme” en *Extender* contiene un breve tutorial sobre la introducción de números, listas, cadenas y símbolos en Scheme.

Las definiciones de salida establecen la relación entre los contextos así como sus respectivos valores predeterminados. Aunque la mayor parte de los cambios se suelen hacer dentro de un bloque `\layout`, los ajustes de valores relacionados con el MIDI solamente tienen efecto cuando se hacen dentro de un bloque `\midi` block.

Algunos ajustes afectan a varias salidas: por ejemplo, si se desactiva el barrado automático, `autoBeaming`, dentro de algún contexto, las barras cuentan como melismas en lo que respecta a la correspondencia entre la música y la letra, como se describe en [Duración automática de las sílabas], página 290. Esta correspondencia se hace tanto para la salida impresa como para el MIDI. Si los cambios hechos sobre el `autoBeaming` dentro de la definición de contexto de un bloque `\layout` no se repiten dentro del correspondiente bloque `\midi`, la letra y la música dejarán de estar sincronizadas dentro del MIDI.

## Véase también

Archivos instalados: `ly/engraver-init.ly`. `ly/performer-init.ly`.

## Score. El contexto maestro

Este es el contexto de notación del nivel más alto. Ningún otro contexto puede contener a un contexto Score. De forma predeterminada, el contexto Score maneja la administración de las indicaciones de compás y se asegura de que ciertos elementos como claves, compases y armaduras están siempre alineados entre los distintos pentagramas.

Se crea implícitamente una instancia del contexto Score cuando se procesa un bloque `\score {...}`.

## Contextos del nivel superior: contenedores de pentagramas

### *StaffGroup*

Agrupar pentagramas y añade un corchete en la parte izquierda, formando un grupo. Las líneas divisorias de los pentagramas contenidos se conectan verticalmente. **StaffGroup** sólo consiste en una colección de pentagramas, con un corchete delante y líneas divisorias de arriba a abajo.

### *ChoirStaff*

Idéntico a **StaffGroup** excepto que las barras de compás de los pentagramas contenidos no se conectan verticalmente.

### *GrandStaff*

Un grupo de pentagramas, con una llave en la parte izquierda que abarca el grupo. Las barras de compás de los pentagramas contenidos se conectan verticalmente.

### *PianoStaff*

Igual que **GrandStaff**, pero contempla la posibilidad de poner el nombre del instrumento a la izquierda del sistema.

## Contextos de nivel intermedio. Pentagramas

### *Staff*

Maneja claves, barras de compás, tonalidades y alteraciones accidentales. Puede contener contextos de **Voice**.

### *RhythmicStaff*

Como **Staff**, pero para imprimir ritmos. Al imprimir se ignoran las alturas de las notas; las notas se imprimen sobre una línea. La salida de MIDI conserva las alturas sin modificación.

### *TabStaff*

Contexto para generar tablaturas. De forma predeterminada dispone la expresión musical como una tablatura de guitarra, impresa sobre seis líneas.

*DrumStaff*

Maneja el tipografiado para instrumentos de percusión. Puede contener contextos `DrumVoice`.

*VaticanaStaff*

Igual que `Staff`, excepto que está pensado para tipografiar piezas en estilo gregoriano.

*MensuralStaff*

Igual que `Staff`, excepto que está diseñado para tipografiar piezas en estilo mensural.

## Contextos del nivel más bajo. Voces

Los contextos del mismo nivel que `Voice` dan un valor inicial a ciertas propiedades e inician los grabadores correspondientes. Un contexto del nivel más bajo es aquel que no tiene un contexto descendiente predeterminado `defaultchild`. Aunque es posible hacer que pueda aceptar o contener subcontextos, éstos solo se pueden crear e introducir de forma explícita.

*Voice*

Corresponde a una voz sobre un pentagrama. este contexto maneja la conversión de las indicaciones dinámicas, plicas, barras, subíndices y superíndices, ligaduras de expresión y de unión, y silencios. Tenemos que crear instancias explícitas de este contexto si necesitamos varias voces en el mismo pentagrama.

*VaticanaVoice*

Lo mismo que `Voice`, excepto que está diseñado para tipografiar piezas en estilo gregoriano.

*MensuralVoice*

Lo mismo que `Voice`, con modificaciones para el tipografiado de piezas en estilo mensural.

*Lyrics*

Corresponde a una voz con letra. Maneja la impresión de una sola línea de letra.

*DrumVoice*

El contexto de voz utilizado en una pauta de percusión.

*FiguredBass*

El contexto en que los objetos `BassFigure` se crean a partir de la entrada escrita en el modo `\figuremode`.

*TabVoice*

El contexto de voz utilizado dentro de un contexto `TabStaff`. Se suele dejar que se cree implícitamente.

*CueVoice*

El contexto de voz que se utiliza para dibujar notas de tamaño reducido, con el principal objetivo de añadir notas guía de un pentagrama a otro, véase [Formateo de las notas guía], página 227. Normalmente se deja que se cree implícitamente.

*ChordNames*

Tipografía nombres de acordes.

### 5.1.2 Crear y referenciar contextos

LilyPond crea automáticamente contextos de nivel inferior si se encuentra una expresión musical antes de que exista un contexto adecuado, pero normalmente esto sólo funciona bien para partituras sencillas o fragmentos musicales como los que aparecen en la documentación. Para partituras más complejas, se recomienda especificar explícitamente todos los contextos con las instrucciones `\new` o `\context`. La sintaxis de estas dos instrucciones es muy similar:

```
[\new | \context] Contexto [ = nombre] [expresión_musical]
```

donde se puede especificar `\new` o `\context`. *Contexto* es el tipo de contexto que se desea crear, *nombre* es un nombre opcional que se da al contexto concreto que se está creando, y *expresión\_musical* es una sola expresión musical que será interpretada por los complementos grabadores y reproductores dentro de este contexto.

El prefijo `\new` sin ningún nombre se usa con frecuencia para crear partituras con muchos pentagramas:

```
<<
  \new Staff \relative {
    % leave the Voice context to be created implicitly
    c''4 c
  }
  \new Staff \relative {
    d''4 d
  }
>>
```



y para introducir varias voces dentro de un solo pentagrama:

```
\new Staff <<
  \new Voice \relative {
    \voiceOne
    c''8 c c4 c c
  }
  \new Voice \relative {
    \voiceTwo
    g'4 g g g
  }
>>
```



`\new` debería usarse siempre para especificar contextos sin nombre.

La diferencia entre `\new` y `\context` se encuentra en la acción que se realiza:

- `\new` con un nombre o sin él, siempre crea un contexto nuevo y distinto, incluso si ya existe un contexto con el mismo nombre:

```
\new Staff <<
  \new Voice = "A" \relative {
    \voiceOne
    c''8 c c4 c c
  }
  \new Voice = "A" \relative {
    \voiceTwo
    g'4 g g g
  }
```

```
}
>>
```



- `\context` con un nombre especificado, crea un contexto nuevo solamente si no existe ya un contexto del mismo tipo y con el mismo nombre, dentro de la misma jerarquía de contextos. En caso contrario, se toma como referencia a dicho contexto creado previamente, y su expresión musical se pasa a este contexto para su interpretación.

Los contextos con nombre pueden ser útiles en casos especiales como la letra de las canciones o los bajos cifrados, como se ve en los ejemplos de [Trabajar con letra y variables], página 299, y Sección “Plantillas de conjuntos vocales” en *Manual de Aprendizaje* para el primer caso, y [Imprimir el bajo cifrado], página 460, para el segundo. De manera más general, una aplicación de los contextos con nombre es la separación entre la disposición de la partitura y el contenido musical. Son válidas cualquiera de las dos formas siguientes:

```
\score {
  <<
    % score layout
    \new Staff <<
      \new Voice = "one" {
        \voiceOne
      }
      \new Voice = "two" {
        \voiceTwo
      }
    >>

    % musical content
    \context Voice = "one" {
      \relative {
        c''4 c c c
      }
    }
    \context Voice = "two" {
      \relative {
        g'8 g g4 g g
      }
    }
  >>
}
```



```
\score {
  <<
    % score layout
    \new Staff <<
      \context Voice = "one" {
        \voiceOne
```

```

    }
    \context Voice = "two" {
      \voiceTwo
    }
  >>

  % musical content
  \context Voice = "one" {
    \relative {
      c' '4 c c c
    }
  }
  \context Voice = "two" {
    \relative {
      g'8 g g4 g g
    }
  }
  >>
}

```



De manera alternativa, se pueden utilizar variables con un efecto similar. Véase Sección “Organizar las piezas mediante variables” en *Manual de Aprendizaje*.

- `\context` sin ningún nombre corresponderá con el primer contexto que se encuentre entre los creados previamente que sean del mismo tipo dentro de la misma jerarquía de contextos, incluso si tiene nombre, y su expresión musical se pasará a dicho contexto para su interpretación. Esta forma rara vez es útil. Sin embargo, `\context` sin nombre y sin expresión musical se usa para establecer el contexto en que se ejecuta un procedimiento de Scheme especificado con `\applyContext`:

```

\new Staff \relative {
  c'1
  \context Timing
  \applyContext #(lambda (ctx)
                    (newline)
                    (display (ly:context-current-moment ctx)))
  c1
}

```

Un contexto debe tener un nombre si se va a hacer referencia a él más tarde, por ejemplo cuando se asocia la letra con la música:

```

\new Voice = "tenor" música
...
\new Lyrics \lyricsto "tenor" letra

```

Para ver más detalles sobre la asociación de letra y música, consulte [Duración automática de las sílabas], página 290.

Las propiedades de todos los contextos de un tipo en particular se pueden modificar dentro de un bloque `\layout` (con una sintaxis diferente), véase [Cambiar todos los contextos del mismo tipo], página 633.



Esta construcción también ofrece una forma de mantener las instrucciones de disposición separadas del contenido musical. Si se va a modificar un solo contexto, debe usarse un bloque `\with`, véase [Cambiar solamente un contexto determinado], página 635.

## Véase también

Manual de aprendizaje: Sección “Organizar las piezas mediante variables” en *Manual de Aprendizaje*.

Referencia de la notación: [Cambiar solamente un contexto determinado], página 635, [Duración automática de las sílabas], página 290.

### 5.1.3 Mantener vivos los contextos

Normalmente los contextos finalizan en el primer momento musical en que no tienen nada que hacer. Así, los contextos de **Voice** mueren tan pronto como ya no contienen ningún evento, los contextos de **Staff** mueren tan pronto como todos los contextos de **Voice** que contenían ya no contengan ningún evento, etc. Esto puede ocasionar dificultades si se tiene que hacer referencia a contextos anteriores que ya han muerto, por ejemplo, al cambiar de pentagramas con instrucciones `\change`, asociar letra con una voz mediante instrucciones `\lyricsto`, o cuando se añaden eventos musicales adicionales a un contexto anterior.

Existe una excepción a esta regla general: dentro de una construcción `{...}` (música secuencial), la noción que la construcción tiene del “contexto actual” descenderá un nivel cada vez que un elemento de la secuencia finaliza en un subcontexto del contexto anterior. Esto evita la creación espúrea de contextos implícitos en ciertas situaciones, pero significa que el primer contexto dentro del que se desciende, se va a mantener vivo hasta el final de la expresión.

Como contraste, los contextos de una expresión hecha con la construcción `<<...>>` (música simultánea) no se prolongan, por lo que si una instrucción que crea un contexto se encierra en otro par de `<<...>>`, se impedirá que el contexto persista durante toda la duración de la secuencia `{...}` que lo contiene.

Cualquier contexto se puede mantener vivo si nos aseguramos de que tiene algo que hacer en cualquier momento musical dado. Los contextos de **Staff** se mantienen con vida si nos aseguramos de que una de sus voces se mantiene viva. Una manera de hacerlo es añadir silencios de separación a una voz en paralelo con la música real. Éstos deben añadirse a todos y cada uno de los contextos de **Voice** que se hayan de mantener vivos. Si se van a usar esporádicamente varias voces, es más seguro mantenerlas todas vivas en lugar de tratar de confiar en las excepciones que hemos mencionado arriba.

En el ejemplo siguiente, tanto la voz A como la voz B se mantienen vivas de esta manera durante la duración de la pieza:

```
musicA = \relative { d''4 d d d }
musicB = \relative { g'4 g g g }
keepVoicesAlive = {
  <<
    \new Voice = "A" { s1*5 } % Keep Voice "A" alive for 5 bars
    \new Voice = "B" { s1*5 } % Keep Voice "B" alive for 5 bars
  >>
}

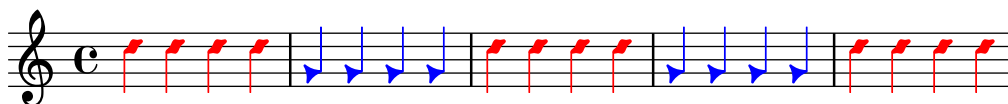
music = {
  \context Voice = "A" {
    \voiceOneStyle
    \musicA
  }
}
```

```

\context Voice = "B" {
  \voiceTwoStyle
  \musicB
}
\context Voice = "A" { \musicA }
\context Voice = "B" { \musicB }
\context Voice = "A" { \musicA }
}

\score {
  \new Staff <<
    \keepVoicesAlive
    \music
  >>
}

```



El ejemplo siguiente muestra cómo se puede escribir una línea melódica esporádica con letra utilizando este enfoque. Por supuesto, en una situación real la melodía y el acompañamiento consistirían en varias secciones diferentes.

```

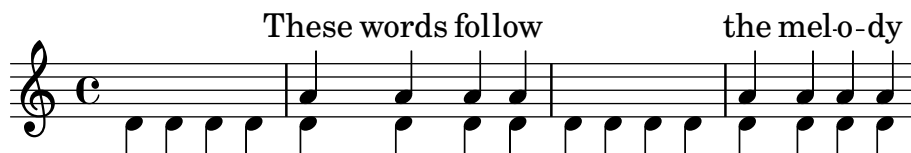
melody = \relative { a'4 a a a }
accompaniment = \relative { d'4 d d d }
words = \lyricmode { These words fol -- low the mel -- o -- dy }
\score {
  <<
    \new Staff = "music" {
      <<
        \new Voice = "melody" {
          \voiceOne
          s1*4 % Keep Voice "melody" alive for 4 bars
        }
        {
          \new Voice = "accompaniment" {
            \voiceTwo
            \accompaniment
          }
          <<
            \context Voice = "melody" { \melody }
            \context Voice = "accompaniment" { \accompaniment }
          >>
          \context Voice = "accompaniment" { \accompaniment }
          <<
            \context Voice = "melody" { \melody }
            \context Voice = "accompaniment" { \accompaniment }
          >>
        }
      >>
    }
  >>
  \new Lyrics \with { alignAboveContext = "music" }
}

```

```

\lyricsto "melody" { \words }
>>
}

```



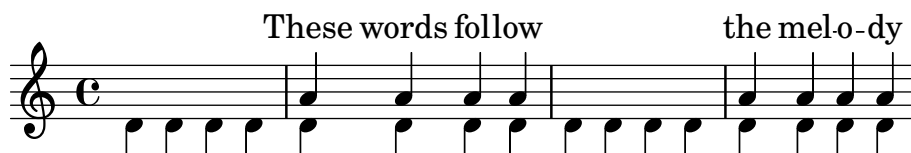
Una forma alternativa, que podría resultar mejor en muchas situaciones, es mantener con vida la línea melódica simplemente incluyendo notas espaciadoras para alinearla correctamente con el acompañamiento:

```

melody = \relative {
  s1 % skip a bar
  a'4 a a a
  s1 % skip a bar
  a4 a a a
}
accompaniment = \relative {
  d'4 d d d
  d4 d d d
  d4 d d d
  d4 d d d
}
words = \lyricmode { These words fol -- low the mel -- o -- dy }

\score {
  <<
    \new Staff = "music" {
      <<
        \new Voice = "melody" {
          \voiceOne
          \melody
        }
        \new Voice = "accompaniment" {
          \voiceTwo
          \accompaniment
        }
      >>
    }
    \new Lyrics \with { alignAboveContext = "music" }
    \lyricsto "melody" { \words }
  >>
}

```



### 5.1.4 Modificar los complementos (plug-ins) de contexto

Los contextos de notación (como `Score` y `Staff`) no sólo almacenan propiedades, también contienen «plug-ins» o complementos llamados ‘grabadores’ que crean elementos de notación. Por ejemplo, el contexto `Voice` contiene un grabador `Note_heads_engraver` que crea las cabezas de nota y el contexto `Staff` contiene un grabador `Key_engraver` que crea la armadura.

Para ver una descripción completa de todos y cada uno de los complementos, consulte Referencia de funcionamiento interno  $\mapsto$  Traducción  $\mapsto$  Grabadores. Cada contexto que se describe en Referencia de funcionamiento interno  $\mapsto$  Traducción  $\mapsto$  Contexto, relaciona los grabadores que se usan para ese contexto.

Puede ser de utilidad jugar un poco con estos complementos. Se hace iniciando un contexto nuevo con `\new` o `\context` y modificándolo:

```
\new contexto \with {
  \consists ...
  \consists ...
  \remove ...
  \remove ...
  etc.
}
{
  ...música...
}
```

donde cada uno de los ... deben ser el nombre de un grabador. Aquí tenemos un ejemplo sencillo que suprime los grabadores `Time_signature_engraver` y `Clef_engraver` de un contexto `Staff`:

```
<<
  \new Staff \relative {
    f'2 g
  }
  \new Staff \with {
    \remove "Time_signature_engraver"
    \remove "Clef_engraver"
  } \relative {
    f'2 g2
  }
>>
```



En el segundo pentagrama no hay indicación de compás ni clave. Éste es un método bastante rudimentario de hacer que desaparezcan los objetos porque afecta a todo el pentagrama. Este método también afecta al espaciado, lo que puede ser deseable o no serlo. Se muestran métodos más sofisticados para quitar objetos en Sección “Visibilidad y color de los objetos” en *Manual de Aprendizaje*.

El ejemplo siguiente muestra una aplicación práctica. Normalmente las líneas divisorias y las indicaciones de compás están sincronizadas a lo largo de toda la partitura. Lo hacen los grabadores `Timing_translator` y `Default_bar_line_engraver`. Estos complementos mantienen al día la administración de las indicaciones de compás, posición dentro del compás, etc. Moviendo

estos grabadores desde el contexto de **Score** al de **Staff**, podemos conseguir una partitura en la que cada pentagrama tiene su propio compás independiente.

```
\score {
  <<
    \new Staff \with {
      \consists "Timing_translator"
      \consists "Default_bar_line_engraver"
    }
    \relative {
      \time 3/4
      c''4 c c c c c
    }
  \new Staff \with {
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  }
  \relative {
    \time 2/4
    c''4 c c c c c
  }
  >>
  \layout {
    \context {
      \Score
      \remove "Timing_translator"
      \remove "Default_bar_line_engraver"
    }
  }
}
```



## Advertencias y problemas conocidos

El orden en que los grabadores se especifican es el orden en que se llaman para realizar su tarea de procesamiento. Normalmente, el orden en que se especifican los grabadores no tiene importancia, pero en algunos casos especiales sí la tiene, por ejemplo donde un grabador escribe una propiedad y otro la lee, o donde un grabador crea un groby otro debe procesarlo.

Las siguientes ordenaciones son importantes:

- el grabador de compases **Bar\_engraver** debe ir normalmente en primer lugar,
- el grabador de digitaciones **New\_fingering\_engraver** debe ir antes del grabador **Script\_column\_engraver** de columnas de inscripciones,
- el **Timing\_translator** debe ir antes del grabador de números de compás **Bar\_number\_engraver**.

## Véase también

Archivos instalados: `ly/engraver-init.ly`.

### 5.1.5 Cambiar los valores por omisión de los contextos

Se pueden cambiar las propiedades de contexto y de grob con las instrucciones `\set` y `\override`, tal y como se describe en Sección 5.3 [Modificar las propiedades], página 646. Estas instrucciones crean eventos musicales, haciendo que los cambios tengan efecto en el punto temporal en que la música se está procesando.

Por contra, esta sección explica la forma de cambiarlos valores *predeterminados* de las propiedades de contexto y de grob en el momento en que se crea el contexto. Existen dos formas de hacerlo. Una modifica los valores predeterminados en todos los contextos de un tipo dado, y el otro modifica los valores predeterminados solamente en una instancia concreta de un contexto.

### Cambiar todos los contextos del mismo tipo

Los ajustes de contexto predeterminados que se han de usar para la composición tipográfica dentro de `Score`, `Staff`, `Voice` y otros contextos se pueden especificar en un bloque `\context` dentro de cualquier bloque `\layout`.

Los ajustes para la salida MIDI, al contrario que para la composición tipográfica, se tendrán que especificar aparte en bloques `\midi` (véase [Definiciones de salida - estructura de los contextos], página 622).

El bloque `\layout` se debe colocar dentro del bloque `\score` al que se aplica, después de la música.

```
\layout {
  \context {
    \Voice
    [ajustes de contexto para todos los contextos Voice]
  }
  \context {
    \Staff
    [ajustes de contexto para todos los contextos Staff]
  }
}
```

Se pueden especificar los siguientes tipos de ajustes:

- Una instrucción `\override`, pero omitiendo el nombre del contexto

```
\score {
  \relative {
    a'4^"Thicker stems" a a a
    a4 a a\ff a
  }
  \layout {
    \context {
      \Staff
      \override Stem.thickness = #4.0
    }
  }
}
```



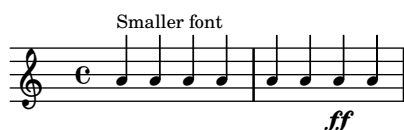
- Estableciendo una propiedad de contexto directamente

```
\score {
```

```

\relative {
  a'4^"Smaller font" a a a
  a4 a a\ff a
}
\layout {
  \context {
    \Staff
    fontSize = #-4
  }
}

```

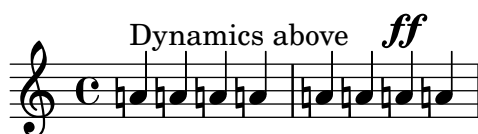


- Una instrucción predefinida tal como `\dynamicUp` o una expresión musical como `\accidentalStyle dodecaphonic`

```

\score {
  \relative {
    a'4^"Dynamics above" a a a
    a4 a a\ff a
  }
  \layout {
    \context {
      \Voice
      \dynamicUp
    }
    \context {
      \Staff
      \accidentalStyle dodecaphonic
    }
  }
}

```



- Una variable definida por el usuario que contenga un bloque `\with;` para ver detalles acerca del bloque `\with`, consulte [Cambiar solamente un contexto determinado], página 635.

```

StaffDefaults = \with {
  fontSize = #-4
}

\score {
  \new Staff {
    \relative {
      a'4^"Smaller font" a a a
      a4 a a a
    }
  }
}

```

```

\layout {
  \context {
    \Staff
    \StaffDefaults
  }
}

```



Las instrucciones de ajuste de propiedades se pueden disponer dentro de un bloque `\layout` sin que estén encerradas en un bloque `\context`. Tales ajustes son equivalentes a incluir las mismas instrucciones de ajuste de propiedades al comienzo de cada uno de los contextos del tipo especificado. Si no se especifica ningún contexto, *todos y cada uno* de los contextos del nivel inferior quedan afectados, véase [Contextos del nivel más bajo. Voces], página 624. La sintaxis de una instrucción de ajuste de propiedades dentro de un bloque `\layout` es la misma que si la misma instrucción estuviera escrita en el propio flujo musical.

```

\score {
  \new Staff {
    \relative {
      a'4^"Smaller font" a a a
      a4 a a a
    }
  }
  \layout {
    \accidentalStyle dodecaphonic
    \set fontSize = #-4
    \override Voice.Stem.thickness = #4.0
  }
}

```



## Cambiar solamente un contexto determinado

Las propiedades de contexto de una única instancia de contexto pueden cambiarse dentro de un bloque `\with`. Todas las demás instancias de contexto del mismo tipo retienen los ajustes predeterminados que LilyPond tiene programados y que se modifican por parte de cualquier bloque `\layout` que se encuentre dentro del ámbito. El bloque `\with` se debe situar inmediatamente después de las instrucciones `\new context-type`:

```

\new Staff \with { [ajustes de contexto para esta instancia de contexto solamente] }
{
  ...
}

```

De forma alternativa, si la música se escribe usando la forma corta de las instrucciones de especificación de modo, como por ejemplo `\chords` en lugar de `\chordmode`, la instrucción `\with` se debe colocar inmediatamente después de la instrucción que especifica el modo:

```

\chords \with { [ajustes de contexto para esta instancia de contexto (implícita) solam

```



```
{
  ...
}
```

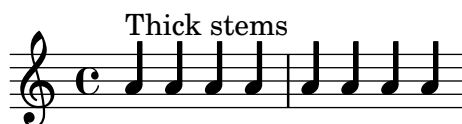
dado que es el contexto implícito creado por estas formas breves el que se quiere modificar. Se aplica la misma consideración a las otras formas cortas de especificar el modo de entrada (`\drums`, `\figures`), véase Sección 5.4.1 [Modos de entrada], página 660.

Dado las modificaciones de contexto especificadas en bloques `\with` están dentro de la música, afectarán a *todas* las salidas (tipografía *y también* el MIDI), a diferencia de los cambios que se hacen dentro de una definición de salida.

Se pueden especificar los siguientes tipos de ajustes:

- Una instrucción `\override`, pero omitiendo el nombre del contexto

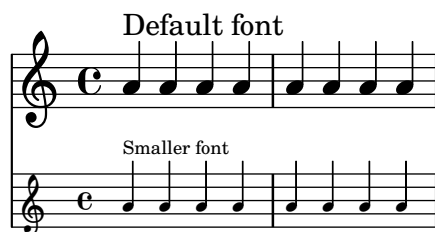
```
\score {
  \new Staff {
    \new Voice \with { \override Stem.thickness = #4.0 }
    {
      \relative {
        a'4^"Thick stems" a a a
        a4 a a a
      }
    }
  }
}
```



- Estableciendo una propiedad de contexto directamente

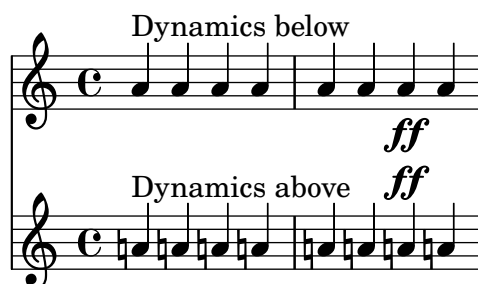
```
\score {
  <<
    \new Staff {
      \relative {
        a'4^"Default font" a a a
        a4 a a a
      }
    }
    \new Staff \with { fontSize = #-4 }
    {
      \relative {
        a'4^"Smaller font" a a a
        a4 a a a
      }
    }
  }
  >>
```

}



- Una instrucción predefinida tal como `\dynamicUp`

```
\score {
  <<
    \new Staff {
      \new Voice {
        \relative {
          a'4^"Dynamics below" a a a
          a4 a a\ff a
        }
      }
    }
    \new Staff \with { \accidentalStyle dodecaphonic }
    {
      \new Voice \with { \dynamicUp }
      {
        \relative {
          a'4^"Dynamics above" a a a
          a4 a a\ff a
        }
      }
    }
  >>
}
```



## Véase también

Referencia de la notación: Sección 5.4.1 [Modos de entrada], página 660,

## Orden de precedencia

El valor de una propiedad que se aplica en un momento determinado se determina de la siguiente forma:

- si está en efecto una instrucción `\override` o `\set` dentro del flujo musical, se usa dicho valor,
- de lo contrario, se usa el valor predeterminado tomado de un enunciado `\with` en las instrucciones de inicio del contexto,

- de lo contrario, se usa el valor tomado del bloque `\context` más reciente que corresponda dentro de los bloques `\layout` o `\midi`,
- de lo contrario se usa el valor predeterminado que LilyPond lleva preprogramado.

## Véase también

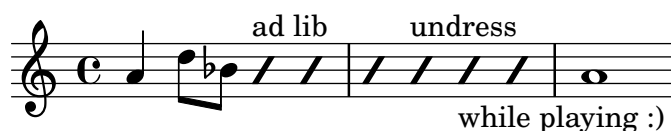
Manual de aprendizaje: Sección “Modificar las propiedades de los contextos” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.1.1 [Explicación de los contextos], página 622, [Contextos del nivel más bajo. Voces], página 624, Sección 5.3.2 [La instrucción `\set`], página 646, Sección 5.3.3 [La instrucción `\override`], página 648, Sección 4.2.1 [El bloque `\layout`], página 579.

### 5.1.6 Definir contextos nuevos

Los contextos específicos, como **Staff** y **Voice**, están contruidos a partir de bloques sencillos. Es posible crear nuevos tipos de contextos con combinaciones distintas de añadidos grabadores.

El siguiente ejemplo muestra cómo construir un tipo diferente de contexto de **Voice** partiendo de cero. Será parecido a **Voice**, pero imprime solamente cabezas centradas en forma de barra inclinada. Se puede usar para indicar improvisación en piezas de jazz,



Estos ajustes se definen dentro de un bloque `\context` que a su vez está dentro de un bloque `\layout`,

```
\layout {
  \context {
    ...
  }
}
```

En el siguiente análisis, la entrada de ejemplo que se muestra debe ir en el lugar de los puntos suspensivos ... del fragmento anterior.

En primer lugar es necesario definir un nombre para el nuevo contexto:

```
\name ImproVoice
```

Debido a que es parecido al contexto **Voice**, queremos órdenes que funcionen dentro de los contextos **Voice** (existentes) para que siga funcionando. Esto se consigue dando al contexto nuevo un alias de **Voice**,

```
\alias Voice
```

El contexto imprimirá notas y textos explicativos, por ello tenemos que añadir los grabadores que aportan esta funcionalidad, y además el grabador que agrupa las notas, plicas y silencios que están en el mismo momento musical en forma de columnas,

```
\consists "Note_heads_engraver"
\consists "Text_engraver"
\consists "Rhythmic_column_engraver"
```

Las cabezas de todas las notas se deben situar sobre la línea central,

```
\consists "Pitch_squash_engraver"
squashedPosition = #0
```

El grabador `Pitch_squash_engraver` modifica las cabezas de nota (creadas por el grabador `Note_heads_engraver`) y establece sus posiciones verticales al valor de `squashedPosition`, en este caso 0, la línea central.

Las notas parecen barras inclinadas y no tienen plica:

```
\override NoteHead.style = #'slash
\hide Stem
```

Todos estos complementos o plug-ins tienen que comunicarse bajo el control del contexto. Los mecanismos con el que se comunican los contextos se establecen mediante la declaración del `\type` (tipo) del contexto. Dentro de un bloque `\layout`, casi todos los contextos serán del tipo `Engraver_group`. Algunos contextos especiales y los contextos de los bloques `\midi` usan otros tipos. La copia y la modificación de una definición de contexto existente también cumplimentan el tipo. Como este ejemplo crea una definición partiendo de cero, tiene que ser especificada explícitamente.

```
\type "Engraver_group"
```

Al juntarlo todo, obtenemos

```
\context {
  \name ImproVoice
  \type "Engraver_group"
  \consists "Note_heads_engraver"
  \consists "Text_engraver"
  \consists "Rhythmic_column_engraver"
  \consists "Pitch_squash_engraver"
  squashedPosition = #0
  \override NoteHead.style = #'slash
  \hide Stem
  \alias Voice
}
```

Los contextos dan lugar a jerarquías. Queremos poner el contexto `ImproVoice` dentro del contexto `Staff`, igual que los contextos de voz normales. Por tanto, modificamos la definición de `Staff` con la instrucción `\accepts` (acepta),

```
\context {
  \Staff
  \accepts ImproVoice
}
```

Con frecuencia, al reutilizar una definición de contexto existente, el contexto resultante se puede usar en cualquier lugar donde el contexto original hubiera podido hacerlo.

```
\layout {
  ...
  \inherit-acceptability to from
}
```

consigue que tengamos contextos del tipo *to* aceptados por todos los contextos que asimismo aceptan *from*. Por ejemplo, usar

```
\layout {
  ...
  \inherit-acceptability "ImproVoice" "Voice"
}
```

añade un `\accepts` para `ImproVoice` tanto a la definición de `Staff` como a la de `RhythmicStaff`.

Lo opuesto a `\accepts` (acepta) es `\denies` (deniega), lo que a veces se necesita cuando se están reutilizando definiciones de contexto existentes.

Organizar las piezas requeridas dentro de un bloque `\layout` nos deja con

```
\layout {
  \context {
    \name ImproVoice
    ...
  }
  \inherit-acceptability "ImproVoice" "Voice"
}
```

Así pues, la salida que aparece al comienzo de esta sub-sección se puede escribir como

```
\relative {
  a'4 d8 bes8
  \new ImproVoice {
    c4^"ad lib" c
    c4 c^"desvístete"
    c c_"mientras tocas :)"
  }
  a1
}
```

Para completar el ejemplo, los cambios que afectan a la jerarquía de contextos se deben repetir dentro de un bloque `\midi` de manera que la salida MIDI dependa de las mismas relaciones de contexto.

## Véase también

Referencia de funcionamiento interno: Sección “`Note_heads_engraver`” en *Referencia de Funcionamiento Interno*, Sección “`Text_engraver`” en *Referencia de Funcionamiento Interno*, Sección “`Rhythmic_column_engraver`” en *Referencia de Funcionamiento Interno*, Sección “`Pitch_squash_engraver`” en *Referencia de Funcionamiento Interno*.

### 5.1.7 Orden de disposición de los contextos

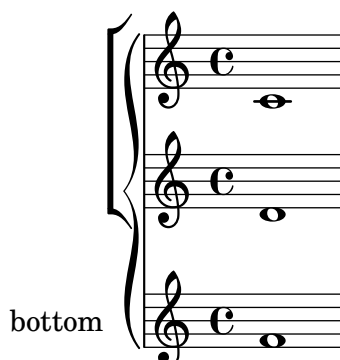
Los contextos se disponen en un sistema normalmente desde arriba hacia abajo en el orden en que se encuentran en el archivo de entrada. Cuando los contextos se anidan unos dentro de otros, el contexto exterior incluye a los contextos anidados tal y como se especifica en el archivo de entrada, siempre y cuando los contextos interiores estén incluidos en la lista “accepts” del contexto externo. Los contextos anidados que no está incluidos en la lista “accepts” del contexto externo se recolocan debajo del contexto externo en lugar de anidarse dentro de él.

La lista “accepts” de un contexto se puede cambiar con las instrucciones `\accepts` (acepta) o `\denies` (niega). `\accepts` añade un contexto a la lista “accepts” y `\denies` elimina un contexto de la lista.

Por ejemplo, un grupo de pentagramas con un corchete cuadrado no se encuentra normalmente en el interior de un pentagrama con llave curva que tenga conectadas las líneas divisorias, y un `GrandStaff` para piano no acepta un `StaffGroup` dentro de él, de forma predeterminada.

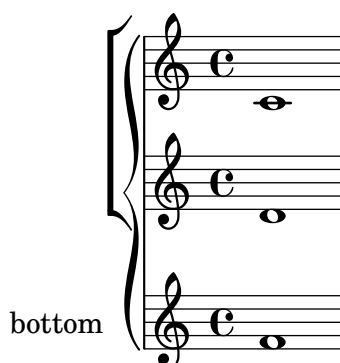
```
\score {
  \new GrandStaff <<
    \new StaffGroup <<
      \new Staff { c'1 }
      \new Staff { d'1 }
    >>
    \new Staff { \set Staff.instrumentName = bottom f'1 }
  >>
}
```

```
}
```



Sin embargo, usando la instrucción `\accepts`, se puede añadir un `StaffGroup` al contexto `GrandStaff`:

```
\score {
  \new GrandStaff <<
    \new StaffGroup <<
      \new Staff { c'1 }
      \new Staff { d'1 }
    >>
    \new Staff { \set Staff.instrumentName = bottom f'1 }
  >>
  \layout {
    \context {
      \GrandStaff
      \accepts "StaffGroup"
    }
  }
}
```



`\denies` se usa principalmente cuando un contexto nuevo se está basando en otro, pero los anidamientos requeridos difieren. Por ejemplo, el contexto `VaticanaStaff` está basado en el contexto `Staff`, pero con el contexto `VaticanaVoice` sustituido por el contexto `Voice` en la lista “accepts”.

Observe que discretamente se crea un contexto de forma implícita si se encuentra una instrucción donde no hay un contexto apropiado para contenerlo.

Dentro de una definición de contexto, el tipo de un subcontexto que se va a crear implícitamente se especifica usando `\defaultchild` (hijo predeterminado). Algunos eventos musicales requieren un contexto ‘Bottom’ (inferior): cuando se encuentra este evento, se crean

subcontextos de forma recursiva hasta que se alcanza un contexto que no tiene establecido el ‘defaultchild’.

La creación implícita de contextos puede dar lugar a pentagramas o partituras nuevos no esperados. La utilización de `\new` para crear contextos explícitamente evita esos problemas.

En ocasiones se necesita que un contexto exista durante un breve intervalo de tiempo, siendo un buen ejemplo el contexto de pentagrama de un ossia. Esto se consigue normalmente mediante la introducción de la definición del contexto en el lugar apropiado en paralelo con la sección correspondiente de la música principal. De forma predeterminada, el contexto temporal se coloca debajo de todos los contextos existentes. Para reposicionarlo por encima del contexto que tenga el nombre “principal”, debería definirse de esta forma:

```
\new Staff \with { alignAboveContext = "principal" }
```

Se presenta una situación similar cuando se posiciona un contexto temporal de letra de una canción dentro de una disposición de varios pentagramas tal como `ChoirStaff`, por ejemplo, cuando se añade una segunda estrofa a una sección que se repite. De forma predeterminada, el contexto temporal de letra se coloca debajo de los pentagramas inferiores. Mediante la definición del contexto temporal de letra con `alignBelowContext` se puede posicionar correctamente debajo del contexto de letra con nombre que contiene el texto de la primera estrofa.

En diversos lugares pueden verse ejemplos que muestran esta recolocación de contextos temporales: véase Sección “Anidado de expresiones musicales” en *Manual de Aprendizaje*, Sección 1.6.2 [Modificación de pentagramas sueltos], página 208, y Sección 2.1.2 [Técnicas específicas para la letra], página 299.

## Véase también

Manual de aprendizaje: Sección “Anidado de expresiones musicales” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 1.6.2 [Modificación de pentagramas sueltos], página 208, Sección 2.1.2 [Técnicas específicas para la letra], página 299.

Manual de utilización del programa: Sección “Aparece un pentagrama de más” en *Utilización del Programa*.

Archivos instalados: `ly/engraver-init.ly`.

## 5.2 Explicación del Manual de referencia de funcionamiento interno

### 5.2.1 Navegar por la referencia del programa

Supongamos que queremos mover la indicación de digitación del fragmento siguiente:

`c' '-2`



Si hace una visita a la documentación en busca de instrucciones de digitación (en [Indicaciones de digitación], página 240), encontrará:

#### Véase también

Referencia de funcionamiento interno: Sección “Fingering” en *Referencia de Funcionamiento Interno*.

La referencia del programador se encuentra disponible en forma de documento HTML. Se recomienda mucho que lo lea en la forma HTML, bien en línea o bien descargando los archivos

de la documentación HTML. Esta sección sería mucho más difícil de entender si está utilizando el manual en formato PDF.

Siga el enlace que lleva a Sección “Fingering” en *Referencia de Funcionamiento Interno*. Al principio de la página, puede ver

Los objetos de digitación se crean por parte de: Sección “Fingering-engraver” en *Referencia de Funcionamiento Interno* y Sección “New\_fingering-engraver” en *Referencia de Funcionamiento Interno*.

Siguiendo los enlaces relacionados dentro de la referencia del programa, podemos seguir el flujo de información dentro del programa:

- Sección “Fingering” en *Referencia de Funcionamiento Interno*: los objetos Sección “Fingering” en *Referencia de Funcionamiento Interno* se crean por parte de: Sección “Fingering-engraver” en *Referencia de Funcionamiento Interno*
- Sección “Fingering-engraver” en *Referencia de Funcionamiento Interno*: Tipos de música aceptados: Sección “fingering-event” en *Referencia de Funcionamiento Interno*
- Sección “fingering-event” en *Referencia de Funcionamiento Interno*: El tipo de evento musical **fingering-event** está descrito en Expresiones musicales con el nombre de Sección “FingeringEvent” en *Referencia de Funcionamiento Interno*

Este camino se recorre en contra de la corriente de información del programa: comienza por la salida y acaba en el evento de entrada. También podríamos haber empezado por un evento de la entrada, y leído siguiendo el flujo de información terminando en su caso en el objeto (u objetos) de la salida.

La referencia del programa también se puede examinar como un documento normal. Contiene capítulos que tratan de **Music definitions** de la Sección “Translation” en *Referencia de Funcionamiento Interno*, y del Sección “Backend” en *Referencia de Funcionamiento Interno*. Cada uno de los capítulos relaciona todas las definiciones utilizadas y todas las propiedades que se pueden ajustar.

### 5.2.2 Interfaces de la presentación

La página HTML que pudimos ver en la sección anterior describe el objeto de presentación llamado Sección “Fingering” en *Referencia de Funcionamiento Interno*. Dicho objeto es un símbolo dentro de la partitura. Tiene propiedades que guardan números (como grosores y direcciones), pero también punteros a objetos relacionados. Un objeto de presentación también se llama un *Grob*, que es una abreviatura de Graphical Object (objeto gráfico). Para ver más detalles acerca de los objetos gráficos o Grobs, consulte Sección “grob-interface” en *Referencia de Funcionamiento Interno*.

La página dedicada a **Fingering** relaciona las definiciones del objeto **Fingering**. Por ejemplo, la página dice

**padding** (dimensión, en espacios de pentagrama):  
0.5

lo que significa que el número se mantendrá a una distancia de al menos 0.5 de la cabeza de la nota.

Cada objeto de presentación puede tener varias funciones como elemento notacional o tipográfico. Por ejemplo, el objeto de digitación **Fingering** tiene los siguientes aspectos

- Su tamaño es independiente del espaciado horizontal, al contrario de las ligaduras o las barras de las figuras.
- Es un elemento de texto. Casi seguro que es un texto muy corto.
- este elemento de texto se tipografía con un tipo de letra, no como las ligaduras o las barras de las figuras.



- Horizontalmente, el centro del símbolo se debe alinear con el centro de la cabeza de la nota.
- Verticalmente, el símbolo se coloca cerca de la nota y del pentagrama.
- La posición vertical también está coordinada con otros símbolos de superíndice y de subíndice.

Cada uno de estos aspectos se capta en lo que se llaman *interfaces*, que se relacionan al final de la página dedicada a Sección “Fingering” en *Referencia de Funcionamiento Interno*

Este objeto contempla los siguientes interfaces: Sección “item-interface” en *Referencia de Funcionamiento Interno*, Sección “self-alignment-interface” en *Referencia de Funcionamiento Interno*, Sección “side-position-interface” en *Referencia de Funcionamiento Interno*, Sección “text-interface” en *Referencia de Funcionamiento Interno*, Sección “text-script-interface” en *Referencia de Funcionamiento Interno*, Sección “font-interface” en *Referencia de Funcionamiento Interno*, Sección “finger-interface” en *Referencia de Funcionamiento Interno* y Sección “grob-interface” en *Referencia de Funcionamiento Interno*.

Al pulsar sobre cualquiera de los enlaces nos desplazaremos a la página del respectivo interfaz del objeto. Cada interfaz tiene un cierto número de propiedades. Algunas de ellas no son para que el usuario las pueda ajustar (‘Propiedades internas’), pero otras sí se pueden modificar.

Hemos estado hablando de *el* objeto **Fingering**, pero realmente esto no significa mucho. El archivo de inicialización (véase Sección “Otras fuentes de información” en *Manual de Aprendizaje*) `scm/define-grobs.scm` muestra el alma del ‘objeto’,

```
(Fingering
  . ((padding . 0.5)
      (avoid-slur . around)
      (slur-padding . 0.2)
      (staff-padding . 0.5)
      (self-alignment-X . 0)
      (self-alignment-Y . 0)
      (script-priority . 100)
      (stencil . ,ly:text-interface::print)
      (direction . ,ly:script-interface::calc-direction)
      (font-encoding . fetaText)
      (font-size . -5) ; don't overlap when next to heads.
      (meta . ((class . Item)
                (interfaces . (finger-interface
                              font-interface
                              text-script-interface
                              text-interface
                              side-position-interface
                              self-alignment-interface
                              item-interface))))))
```

Como podemos ver, el objeto **Fingering** no es más que un montón de valores de variables, y la página web de la Referencia de funcionamiento interno se genera directamente a partir de esta definición.

### 5.2.3 Determinar la propiedad del grob

Recordemos que queríamos cambiar la posición del **2** en

c''-2



Puesto que el **2** se encuentra colocado verticalmente sobre su nota, tenemos que negociar con el interfaz asociado con esta colocación. Esto se hace usando **side-position-interface**. La página que describe este interface dice:

#### **side-position-interface**

Colocar un objeto víctima (este mismo) junto a otros objetos (el soporte). La propiedad **direction** significa dónde poner el objeto víctima con relación al soporte (¿a la izquierda o a la derecha, encima o debajo?)

Debajo de esta descripción, la variable **padding** (relleno) se describe como

**padding** (dimensión, en espacios de pentagrama)

Añadir esta cantidad de espacio adicional entre objetos que están unos junto a otros.

Aumentando el valor de **padding**, podemos alejar la cifra de digitación de la cabeza de la nota. La siguiente orden inserta un espacio en blanco de tres “espacios de pentagrama” entre la nota y la digitación:

```
\once \override Voice.Fingering.padding = #3
```

Al insertar este relleno antes de que se haya creado el objeto de digitación, llegamos al siguiente resultado:

```
\once \override Voice.Fingering.padding = #3
```

c''-2



En este caso, el contexto de este truco es **Voice**. Véase el apartado dedicado al añadido Sección “Fingering-engraver” en *Referencia de Funcionamiento Interno*, que dice:

El grabador **Fingering-engraver** forma parte de los contextos: ... Sección “Voice” en *Referencia de Funcionamiento Interno*

## 5.2.4 Convenciones de nombres

Se hace necesario presentar una panorámica de las diversas convenciones de nomenclatura:

- funciones de Scheme: minúsculas-con-guiones (lo que incluye a los nombres de una sola palabra)
- funciones de Scheme específicas de LilyPond: ly:más-estilo-de-scheme
- eventos, clases y propiedades musicales: como-las-funciones-de-scheme
- interfaces de Grobs: estilo-scheme
- propiedades de backend: estilo-scheme (¡pero X e Y en mayúsculas!)
- contextos (y ExpresionesMusicales y grobs): Mayúsculas o MayúsculasDeCamello
- propiedades de contexto: minúsculasSeguidoDeMayúsculasDeCamello
- grabadores: Mayúsculas\_seguido\_de\_minúsculas\_y\_con\_barras\_bajas

## 5.3 Modificar las propiedades

### 5.3.1 Panorámica de la modificación de las propiedades

Cada contexto es responsable de la creación de ciertos tipos de objetos gráficos. Los ajustes que se usan para imprimir estos objetos también se almacenan por contexto. Mediante la modificación de estos ajustes, se puede alterar la apariencia de los objetos.

Existen dos tipos diferentes de propiedades almacenadas en los contextos: las propiedades de contexto y las propiedades de grob. Las propiedades de contexto son propiedades que se aplican al contexto como un todo y controlan la forma en que el propio contexto se imprime. Por contra, las propiedades de grob se aplican a los tipos de grob específicos que se imprimirán dentro del contexto.

Las instrucciones `\set` y `\unset` se usan para cambiar los valores de las propiedades de contexto. Las instrucciones `\override` y `\revert` se usan para cambiar los valores de las propiedades de grob.

### Véase también

Referencia de funcionamiento interno: Sección “Backend” en *Referencia de Funcionamiento Interno*, Sección “All layout objects” en *Referencia de Funcionamiento Interno*, Sección “OverrideProperty” en *Referencia de Funcionamiento Interno*, Sección “RevertProperty” en *Referencia de Funcionamiento Interno*, Sección “PropertySet” en *Referencia de Funcionamiento Interno*.

### Advertencias y problemas conocidos

El «back-end» o motor de salida no es muy estricto en la comprobación de tipos de las propiedades de objetos. Las referencias cíclicas en valores Scheme de propiedades pueden producir cuelgues o salidas abruptas, o las dos cosas.

### 5.3.2 La instrucción `\set`

Cada contexto puede tener distintas *propiedades*, variables contenidas dentro de ese contexto. Se pueden cambiar mientras dura el paso de interpretación. Se consigue insertando la instrucción `\set` dentro de la música:

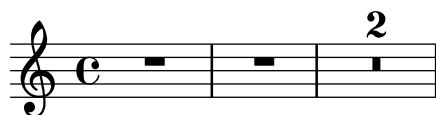
```
\set contexto.propiedad = #valor
```

*valor* es un objeto de Scheme, razón por la que va precedido del carácter almohadilla, `#`.

El nombre de las propiedades de contexto suele ir en minúsculas con mayúscula en medio. Controlan sobre todo la traducción de la música a la notación, p.ej. `localAlterations` (para determinar si hay que imprimir alteraciones o no), o `measurePosition` (para determinar cuándo hay que imprimir una línea divisoria). El valor de las propiedades de contexto puede modificarse con el tiempo durante la interpretación de la música; un ejemplo obvio es `measurePosition`. Las propiedades de contexto se modifican mediante la instrucción `\set`.

Por ejemplo, los silencios multicompas se combinan en un solo compás (como se explica en `\undefined` [`\undefined`], página `\undefined`) si el valor de la propiedad de contexto `skipBars` se establece a `#t` (verdadero):

```
R1*2
\set Score.skipBars = ##t
R1*2
```



Si se omite el argumento *context*, entonces se utiliza el contexto actual de nivel más bajo (normalmente **ChordNames**, **Voice** o **Lyrics**). En este ejemplo:

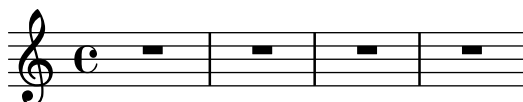
```
\set Score.autoBeaming = ##f
\relative {
  e' '8 e e e
  \set autoBeaming = ##t
  e8 e e e
} \\\
\relative {
  c' '8 c c c c8 c c c
}
```



El cambio se aplica ‘al vuelo’, mientras dura la música, de forma que el ajuste sólo afecta al segundo grupo de corcheas.

Observe que el contexto del nivel más bajo no siempre contiene la propiedad que queríamos modificar: por ejemplo, intentar ajustar el valor de la propiedad **skipBars** del contexto predeterminado del nivel más bajo, que en este caso es **Voice**, no tendrá ningún efecto, porque **skipBars** es una propiedad del contexto **Score**.

```
R1*2
\set skipBars = ##t
R1*2
```



Los contextos son jerárquicos, y si se ha especificado un contexto mayor, por ejemplo **Staff**, entonces el cambio se aplicaría también a todos los contextos **Voice** dentro del pentagrama actual.

También existe una instrucción **\unset**:

```
\unset contexto.propiedad
```

que elimina la definición de *propiedad*. Esta instrucción elimina la definición solamente si está establecida dentro de *contexto*. Properties that have been set in enclosing contexts will not be altered by an **unset** in an enclosed context:

```
\set Score.autoBeaming = ##t
\relative {
  \unset autoBeaming
  e' '8 e e e
  \unset Score.autoBeaming
  e8 e e e
} \\\
\relative {
  c' '8 c c c c8 c c c
}
```



Como `\set`, el argumento *contexto* no se tiene que especificar para un contexto del nivel más bajo, por lo que los dos enunciados

```
\set Voice.autoBeaming = ##t
\set autoBeaming = ##t
```

son equivalentes si el contexto inferior en curso es *Voice*.

Los ajustes con `\set` o `\unset` que se aplican solamente a un único paso de tiempo se pueden escribir con `\once`, por ejemplo en

```
c''4
\once \set fontSize = #4.7
c''4
c''4
```



En el manual de Referencia de funcionamiento interno hay una descripción completa de todas las propiedades de contexto disponibles, consulte Traducción  $\mapsto$  Propiedades de contexto modificables por el usuario.

## Véase también

Referencia de funcionamiento interno: Sección “Tunable context properties” en *Referencia de Funcionamiento Interno*.

### 5.3.3 La instrucción `\override`

Existe un tipo especial de propiedad de contexto: la descripción de los grobs. Las descripciones de los grobs reciben un nombre en *MayúsculasDeCamello* (empezando en mayúscula). Contienen los ‘ajustes predeterminados’ para un tipo particular de grob, en forma de lista asociativa. Consulte `scm/define-grobs.scm` para ver los ajustes de cada descripción de grob. Las descripciones de grob se modifican con `\override`.

La sintaxis de la instrucción `\override` es

```
\override [contexto.]NombreDelGrob.propiedad = #valor
```

Por ejemplo, podemos aumentar el grosor de la plica de una figura sobrescribiendo la propiedad `thickness` (grosor) del objeto `Stem` (plica):

```
c''4 c''
\override Voice.Stem.thickness = #3.0
c''4 c''
```



Si no se ha especificado ningún contexto en la instrucción `\override`, se utiliza el contexto del nivel inferior:

```
\override Staff.Stem.thickness = #3.0
<<
\relative {
  e''4 e
  \override Stem.thickness = #0.5
  e4 e
```

```

} \\  

\relative {  

  c' '4 c c c  

}  

>>

```



Algunas opciones susceptibles de trucaje, se llaman ‘subpropiedades’ y residen dentro de las propiedades. Para efectuar trujajes sobre ellas, utilice instrucciones de la forma

```
\override Stem.details.beamed-lengths = #'(4 4 3)
```

o para modificar los extremos de los objetos extensos, utilice una forma como las siguientes:

```

\override TextSpanner.bound-details.left.text = "left text"  

\override TextSpanner.bound-details.right.text = "right text"

```

El efecto de una instrucción de sobreescritura `\override` se puede deshacer con `\revert`.

La sintaxis de la instrucción `\revert` es

```
\revert [Contexto.]NombreDelGrob.propiedad
```

Por ejemplo,

```

\relative {  

  c' '4  

  \override Voice.Stem.thickness = #3.0  

  c4 c  

  \revert Voice.Stem.thickness  

  c4  

}

```



Los efectos de `\override` y `\revert` se aplican a todos los grobs del contexto afectado partiendo del momento actual y hacia adelante:

```

<<  

  \relative {  

    e' '4  

    \override Staff.Stem.thickness = #3.0  

    e4 e e  

  } \\  

  \relative {  

    c' '4 c c  

    \revert Staff.Stem.thickness  

    c4  

  }  

>>

```



Se puede usar `\once` con `\override` o `\revert` para afectar solamente al instante de tiempo actual:

```
<<
\relative c {
  \override Stem.thickness = #3.0
  e''4 e e e
} \\\
\relative {
  c''4
  \once \override Stem.thickness = #3.0
  c4 c c
}
>>
```



## Véase también

Referencia de funcionamiento interno: Sección “Backend” en *Referencia de Funcionamiento Interno*

### 5.3.4 La instrucción `\tweak`

El cambio de las propiedades de grob mediante `\override` produce la aplicación del cambio a todos los grobs dados en el contexto en el momento en que se aplica dicho cambio. Sin embargo, en ocasiones podemos desear que los cambios se apliquen a un solo grob en lugar de a todos los grobs del contexto afectado. Esto se consigue con la instrucción `\tweak`, que tiene la sintaxis siguiente:

```
\tweak [objeto-de-presentación.]propiedad-del-grob valor
```

Es opcional especificar el *objeto-de-presentación*. La instrucción `\tweak` se aplica al objeto musical que viene inmediatamente después de *valor* dentro del flujo musical.

Para ver una introducción a la sintaxis y los usos de la instrucción `\tweak`, consulte Sección “Métodos de trucaje” en *Manual de Aprendizaje*.

Si se colocan varios elementos similares en el mismo momento musical, la instrucción `\override` no se puede usar para modificar uno solo de ellos: aquí es donde se debe usar la instrucción `\tweak`. Entre los elementos que pueden aparecer más de una vez en el mismo momento musical están los siguientes:

- las cabezas de las notas de un acorde
- signos de articulación sobre la misma nota
- ligaduras de unión entre notas de un acorde
- corchetes de grupos especiales que comienzan en el mismo momento

En este ejemplo se modifican el color de una cabeza y el tipo de otra, dentro del mismo acorde:

```
< c''
\tweak color #red
d''
g''
\tweak duration-log #1
a''
```

> 4



`\tweak` se puede usar para modificar ligaduras de expresión:

```
\relative { c'-\tweak thickness #5 ( d e f) }
```



Para que funcione la instrucción `\tweak`, debe permanecer adyacente al objeto al que se ha de aplicar después de que el código de entrada se ha convertido a un flujo musical. El trucaje de un acorde completo no hace nada porque su evento musical actúa solamente como un contenedor, y todos los objetos de presentación se crean a partir de eventos dentro del `EventChord`:

```
\tweak color #red c''4
\tweak color #red <c'' e''>4
<\tweak color #red c'' e''>4
```



La instrucción `\tweak` sencilla no se puede usar para modificar ningún objeto que no se haya creado directamente a partir de la entrada. Concretamente, no afecta a las plicas, barras automáticas ni alteraciones, porque éstos se generan posteriormente por parte de objetos de presentación `NoteHead` más que por elementos musicales del flujo de entrada.

Tales objetos de presentación creados indirectamente se pueden trucar usando la forma de la instrucción `\tweak` en que el nombre del grob se especifica de forma explícita:

```
\tweak Stem.color #red
\tweak Beam.color #green c''8 e''
<c'' e'' \tweak Accidental.font-size #-3 ges''>4
```



No se puede usar `\tweak` para modificar las claves o las indicaciones de compás, porque éstos se separan de cualquier instrucción `\tweak` precedente dentro del flujo de entrada merced a la inserción automática de elementos adicionales que se requieren para especificar el contexto.

Se pueden colocar varias instrucciones `\tweak` antes de un elemento de notación; todos le afectan:

```
c'
-\tweak style #'dashed-line
-\tweak dash-fraction #0.2
-\tweak thickness #3
-\tweak color #red
\glissando
```



f''



El flujo musical que se genera a partir de una sección de un archivo de entrada (incluido cualquier elemento insertado automáticamente) puede examinarse, véase Sección “Presentación de las expresiones musicales” en *Extender*. Esto puede ser de utilidad en la determinación de lo que puede modificarse por medio de una instrucción `\tweak`.

## Véase también

Manual de aprendizaje: Sección “Métodos de trucaje” en *Manual de Aprendizaje*.

Manual de extensión: Sección “Presentación de las expresiones musicales” en *Extender*.

## Advertencias y problemas conocidos

No se puede usar la instrucción `\tweak` para modificar los puntos de control de una sola de varias ligaduras de unión dentro de un acorde, aparte de la primera que se encuentre en el código de entrada.

### 5.3.5 `\set` frente a `\override`

Tanto la instrucción `\set` como `\override` manipulan propiedades asociadas a contextos. En ambos casos las propiedades siguen una *jerarquía de contextos*: las propiedades que no se establecen dentro del contexto aún presentan los valores del contexto de su padre respectivo.

El valor y la duración en el tiempo de una propiedad de contexto son dinámicos y están disponibles solamente cuando la música se está interpretando o ‘iterando’. En el momento de la creación del contexto, se inicializan las propiedades a partir de las definiciones correspondientes (así como otras posibles modificaciones) de dicho contexto. Cualquier cambio posterior se obtiene a través de instrucciones de establecimiento de propiedades dentro de la propia música.

Las definiciones de grob (objetos gráficos) son una clase *especial* de propiedades de contexto, pues su estructura y uso son distintos de las propiedades de contexto ordinarias. A diferencia de las propiedades de los contextos normales, las definiciones de grob se subdividen en *propiedades de grob*.

Además, en contraste con las propiedades de contexto normales, las definiciones de grob tienen su propio conjunto de ‘utilidades de mantenimiento’ de las propiedades individuales y, en su caso, sub-propiedades. Esto supone que es posible definir las distintas partes dentro de diferentes contextos y aún tener la definición completa del grob en el momento de su creación, montada a partir de todas las piezas proporcionadas por el contexto actual y su ancestro o ancestros, conjuntamente.

Un grob u objeto gráfico se crea normalmente por parte de un grabador en el momento de la interpretación de una expresión musical y recibe sus propiedades iniciales de la definición de grob en curso del contexto del grabador. El grabador (u otras partes del ‘backend’ de LilyPond) pueden después modificar (o añadirse a) las propiedades iniciales del grob; sin embargo, ello no afecta a la definición de grob del contexto.

Lo que LilyPond conoce como ‘propiedades de grob’ en el contexto del trucaje a nivel de usuario son en realidad las propiedades de la definición de grob de un contexto.

Se accede a las definiciones de grob con un conjunto de instrucciones diferente. Se manipulan usando `\override` y `\revert` y tienen un nombre que empieza con una letra mayúscula (como ‘NoteHead’) mientras que las propiedades de contexto ordinarias se manipulan utilizando `\set` y `\unset` y se nombran empezando en letra minúscula.

Las instrucciones `\tweak` y `\overrideProperty` cambian las propiedades de grob pasando por encima de las propiedades de contexto completamente. En su lugar, atrapan a los grobs tan pronto se crean, y establecen propiedades sobre ellos directamente con `\tweak` cuando se originan a partir de un evento musical o, en el caso de `\overrideProperty`, para una sobreescritura específica.

### 5.3.6 La instrucción `\offset`

Aunque es posible fijar propiedades de grob a valores nuevos con las instrucciones `\override`, `\tweak` y `\overrideProperty`, muchas veces conviene más modificar dichas propiedades en una cantidad relativa al valor predeterminado. La instrucción `\offset` se encuentra disponible para este fin.

La sintaxis de `\offset` es

```
[-]\offset propiedad incrementos elemento
```

La instrucción funciona añadiendo el contenido de *incrementos* al ajuste predeterminado de la *propiedad* del grob indicado por *elemento*.

Dependiendo de la formulación de la instrucción, `\offset` puede actuar como un `\tweak` (trucaje) o como un `\override` (sobreescritura). Las variaciones en su uso se explicarán después de haber examinado las propiedades de grob que se pueden usar con `\offset`.

#### *Propiedades que admiten el desplazamiento*

Muchas, pero no todas, las propiedades de grob admiten la aplicación de un desplazamiento. Si *propiedad* no se puede desplazar, el objeto permanece sin cambios y se emite una advertencia. En estos casos se deben usar en su lugar `\override` o `\tweak` para modificar el objeto.

Se puede trabajar por ensayo y error y dejar que las advertencias nos sirvan de guía para saber lo que puede o no puede ser desplazado. Sin embargo, es posible un enfoque más sistemático.

Los criterios siguientes determinan si una propiedad se puede modificar con `\offset`:

- La propiedad tiene un ‘ajuste predeterminado’ en la descripción del grob. Tales propiedades aparecen listadas para cada grob en Sección “All layout objects” en *Referencia de Funcionamiento Interno* (y también están en `scm/define-grobs.scm`).
- La propiedad admite un valor numérico. Entre los valores numéricos se encuentran `number`, una lista de `numbers`, `number-pair`, y `number-pair-list`. Las páginas de Sección “All layout objects” en *Referencia de Funcionamiento Interno* contienen una lista de los tipos de datos característicos de cada propiedad. No importa si el ajuste predeterminado es una función.
- La propiedad no puede ser una ‘subpropiedad’ (una propiedad que reside dentro de otra propiedad).
- Las propiedades establecidas a valores infinitos no se pueden desplazar. No hay ninguna forma sensata de incrementar los valores infinitos positivos o negativos.

Los ejemplos siguientes tratan diversas propiedades de grob frente a los criterios que se han expuesto.

- Propiedades que se pueden desplazar

`Hairpin.height`

Esta propiedad no es una subpropiedad, y está en la lista que aparece en Sección “Hairpin” en *Referencia de Funcionamiento Interno*. En cuanto a su valor, admite ‘dimension, in staff space’ (dimensión, en espacios de pentagrama) fijado a 0.6666 (que claramente es un `number` no infinito).

`Arpeggio.positions`

En la lista de la página que describe Sección “Arpeggio” en *Referencia de Funcionamiento Interno* aparece una propiedad `positions` que admite una ‘pareja

de números’. Su valor predeterminado es `ly:arpeggio::positions`, que es un ‘callback’ que se evalúa durante la fase de tipografiado para dar como resultado una pareja de números para cualquier objeto `Arpeggio` dado.

- Propiedades que no admiten el desplazamiento

`Hairpin.color`

No hay un listado para `color` en Sección “Hairpin” en *Referencia de Funcionamiento Interno*.

`Hairpin.circled-tip`

La lista de `Hairpin.circled-tip` en la página de Sección “Hairpin” en *Referencia de Funcionamiento Interno* muestra que admite un valor `boolean`. Los valores booleanos so no numéricos.

`Stem.details.lengths`

Aunque está listado en Sección “Stem” en *Referencia de Funcionamiento Interno* y su valor por omisión es una lista de `numbers`, es una ‘subpropiedad’. Por el momento no se contemplan las ‘propiedades anidadas’.

## ***\offset como sobreescritura***

Si *elemento* es un nombre de grob como `Arpeggio` o `Staff.OttavaBracket`, el resultado es un `\override` o sobreescritura de el tipo de grob especificado.

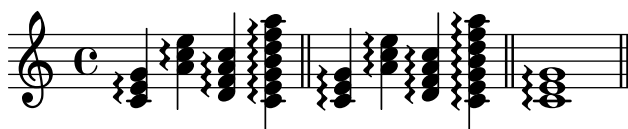
`\offset propiedad incremento [contexto.]NombreDeGrob`

Observe que el guión precedente *nunca* se utiliza con la forma ‘override’, de igual manera que nunca se usa con la propia instrucción `\override`.

El ejemplo siguiente usa la forma ‘override’, de sobreescritura, para alargar los arpeggios predeterminados que aparecen en el primer compás para que cubran todo el tamaño de los acordes. Los arpeggios se amplían en medio espacio de pentagrama hacia arriba y hacia abajo. También se muestra la misma operación efectuada sobre el primer acorde con una sobreescritura normal de la propiedad `the positions`. Este método no es en absoluto expresivo de la acción de ‘ampliar en medio espacio de pentagrama’, porque los extremos se tienen que especificar mediante coordenadas absolutas, no relativas. Además, se necesitarían sobreescrituras individuales para los otros acordes, ya que varían en tamaño y posición.

```
arpeggioMusic = {
  <c' e' g'>\arpeggio <a' c' e'>\arpeggio
  <d' f' a' c'>\arpeggio <c' e' g' b' d' f' a'>\arpeggio
}

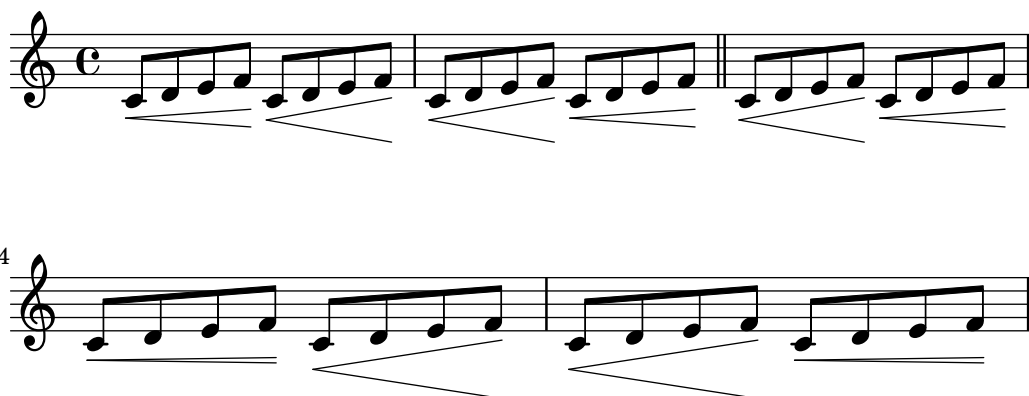
{
  \arpeggioMusic
  \bar "||"
  \offset positions #'(-0.5 . 0.5) Arpeggio
  \arpeggioMusic
  \bar "||"
  \once \override Arpeggio.positions = #'(-3.5 . -0.5)
  <c' e' g'>1\arpeggio
  \bar "||"
}
```



En su uso como ‘override’, `\offset` se puede hacer preceder por `\once` o por `\temporary` y revertirse mediante `\revert` con la *propiedad*. (véase Sección “Funciones de sustitución intermedias” en *Extender*). Esto se sigue a partir del hecho de que `\offset` en realidad crea un `\override` para la *propiedad*.

```
music = { c'8\< d' e' f'\! }

{
  \music
  \offset height 1 Hairpin
  \music
  \music
  \revert Hairpin.height
  \music
  \bar "||"
  \once \offset height 1 Hairpin
  \music \music
  \bar "||"
  \override Hairpin.height = 0.2
  \music
  \temporary \offset height 2 Hairpin
  \music
  \music
  \revert Hairpin.height
  \music
  \bar "||"
}
```



Así como `\override`, la forma ‘override’ de `\offset` se puede usar con `\undo` y con `\single`.

```
longStem = \offset length 6 Stem

{
  \longStem c'4 c''' c' c''
  \bar "||"
  \undo \longStem c'4 c''' c' c''
  \bar "||"
  \single \longStem c'4 c''' c' c''
  \bar "||"
}
```

}



### *\offset como trucaje*

Si *elemento* es una expresión musical como ( o `\arpeggio`, el resultado es la misma expresión musical con un trucaje aplicado.

```
[-]\offset [NombreDeGrob.]propiedad incrementos expresión-musical
```

La sintaxis de `\offset` en forma de ‘tweak’ es análoga a la propia instrucción `\tweak`, tanto en orden como en cuanto a la presencia o ausencia del guión inicial.

El ejemplo siguiente usa la forma ‘tweak’ para ajustar la posición vertical del objeto `BreathingSign`. Compárelo con la instrucción `\tweak` normal que también aparece. La sintaxis es equivalente; sin embargo, la salida de `\tweak` es menos intuitiva, ya que `BreathingSign.Y-offset` se calcula a partir de la tercera línea del pentagrama. No es necesario saber cómo se calcula `Y-offset`, al usar `\offset`.

```
{
  c''4
  \breathe
  c''4
  \offset Y-offset 2 \breathe
  c''2
  \tweak Y-offset 3 \breathe
}
```



En el ejemplo anterior, los objetos trucados se crearon directamente a partir de la entrada del usuario: la instrucción `\breathe` era una instrucción explícita para devolver un objeto `BreathingSign`. Dado que el objeto de la instrucción no es ambiguo, no había necesidad de especificar el nombre del grob. Sin embargo, cuando un objeto se crea *indirectamente*, es necesario incluir el nombre del grob. Es lo mismo que para la instrucción `\tweak`.

En el ejemplo siguiente, el objeto `Beam` se baja en dos espacios de pentagrama aplicando `\offset` a la propiedad `positions`.

La primera aplicación de `\offset` exige que se incluya el nombre del grob, porque nada en el código de entrada crea explícitamente la barra. En la segunda aplicación, la barra se crea manualmente con la expresión musical `[]`; por tanto, el nombre del grob no es necesario (también aparece en la ilustración una abreviatura: un solo número se aplica a los dos miembros de una pareja).

```
{
  c''8 g'' e'' d''
  \offset Beam.positions #'(-2 . -2)
  c''8 g'' e'' d''
  c''8 g'' e'' d''
  c''8-\offset positions #-2 [ g'' e'' d'']
}
```

}



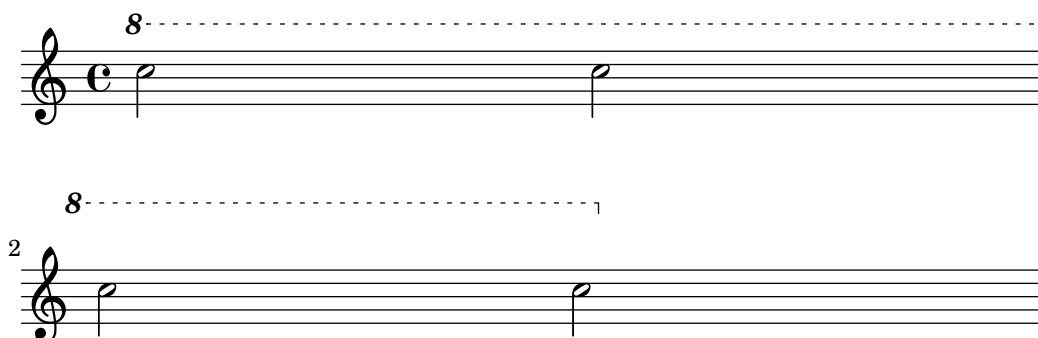
### *\offset con objetos extensos divididos*

También es posible modificar segmentos de un objeto que se extiende sobre un salto o saltos de línea. En este caso, *offsets* admite una lista de valores del tipo de datos requerido por la propiedad.

La instrucción *\offset* usada de esta forma es similar a la instrucción *\alterBroken* (véase Sección 5.5.5 [Modificación de objetos de extensión divididos], página 688). Sin embargo, a diferencia de *\alterBroken*, los valores que se le dan a *\offset* son relativos, no absolutos.

El ejemplo siguiente desplaza el objeto ‘dividido’ *OttavaBracket* a través de su propiedad *staff-padding*. Dado que la propiedad toma un *number*, se le proporciona a *incrementos* una lista de números para aplicarlos a los dos segmentos creados por el salto de línea. La parte de corchete de la primera línea queda, en realidad, inalterada, ya que se añade 0 a su valor por omisión de *staff-padding*. El segmento que está en la segunda línea se eleva tres espacios de pentagrama a partir de su altura por omisión. La altura predeterminada resulta ser 2, aunque no es necesario saberlo para conseguir la posición deseada.

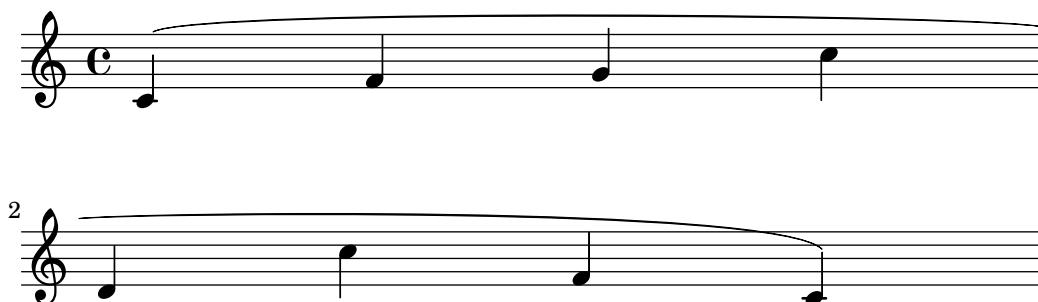
```
{
  \offset staff-padding #'(0 3) Staff.OttavaBracket
  \ottava #1
  c'''2 c'''
  \break
  c'''2 c'''
}
```



El ejemplo siguiente copia el efecto de la instrucción *\shape* mediante el incremento de la propiedad *control-points* del objeto *Slur*. Aquí, *incrementos* es una lista de parejas de números, una por cada segmento de la ligadura. Este ejemplo produce un resultado idéntico a la ilustración correspondiente que aparece en Sección 5.5.4 [Modificación de las formas], página 684.

```
{
  c'4-\offset control-points #'(
    ((0 . 0) (0 . 0) (0 . 0) (0 . 1))
    ((0.5 . 1.5) (1 . 0) (0 . 0) (0 . -1.5))
  ) ( f'4 g' c'
  \break
  d'4 c' f' c')
```

}



### 5.3.7 Modificación de las listas-A

Ciertas propiedades configurables por parte del usuario se representan internamente como *listas-A* (listas asociativas), que almacenan duplas de *claves* y *valores*. La estructura de una lista-A es la siguiente:

```
'((clave1 . valor1)
  (clave2 . valor2)
  (clave3 . valor3)
  ...)
```

Si una lista-A es una propiedad de un grob o una variable de `\paper`, sus claves se pueden modificar individualmente sin que afecte a las otras claves.

Por ejemplo, para reducir el espacio entre pentagramas adyacentes dentro de un grupo, use la propiedad `staff-staff-spacing` del grob `StaffGrouper`. La propiedad es una lista-A con cuatro claves: `basic-distance` (distancia básica), `minimum-distance` (distancia mínima), `padding` (relleno) y `stretchability` (ampliabilidad). Los ajustes estándar para esta propiedad se relacionan en la sección “Backend” de la Referencia de Funcionamiento Interno (véase Sección “StaffGrouper” en *Referencia de Funcionamiento Interno*):

```
'((basic-distance . 9)
  (minimum-distance . 7)
  (padding . 1)
  (stretchability . 5))
```

Una forma de acercar los pentagramas entre sí es reducir el valor de la clave `basic-distance` (9) para que se corresponda con el valor de `minimum-distance` (7). Para modificar una única clave de forma individual, utilice una *declaración anidada*:

```
% default space between staves
\new PianoStaff <<
  \new Staff { \clef treble c''1 }
  \new Staff { \clef bass c1 }
>>

% reduced space between staves
\new PianoStaff \with {
  % this is the nested declaration
  \override StaffGrouper.staff-staff-spacing.basic-distance = #7
} <<
  \new Staff { \clef treble c''1 }
  \new Staff { \clef bass c1 }
```

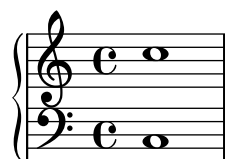
&gt;&gt;



La utilización de una declaración anidada actualiza la clave especificada (como **basic-distance** en el ejemplo anterior) sin alterar ninguna de las otras claves que ya se habían establecido para la misma propiedad.

Ahora, supongamos que deseamos que los pentagramas estén tan próximos como sea posible sin que se superpongan. La manera más sencilla de hacerlo es establecer las cuatro claves de la lista-A a cero. Sin embargo, no es necesario escribir cuatro declaraciones anidadas, una por cada clave. En lugar de eso, se puede redefinir completamente la propiedad con una sola declaración, como una lista-A:

```
\new PianoStaff \with {
  \override StaffGrouper.staff-staff-spacing =
    #'((basic-distance . 0)
      (minimum-distance . 0)
      (padding . 0)
      (stretchability . 0))
} <<
  \new Staff { \clef treble c''1 }
  \new Staff { \clef bass   c1   }
>>
```



Observe que cualquier clave que no haya sido relacionada explícitamente en la definición de la lista-A, será reiniciada a sus valores *predeterminados si no se han fijado*. En el caso de **staff-staff-spacing**, el valor de cualquier clave no fijada se reiniciaría a cero (excepto **stretchability**, que toma el valor de **basic-distance** si no se fija). Así, las dos declaraciones siguientes son equivalentes:

```
\override StaffGrouper.staff-staff-spacing =
  #'((basic-distance . 7))

\override StaffGrouper.staff-staff-spacing =
  #'((basic-distance . 7)
    (minimum-distance . 0)
    (padding . 0))
```



```
(stretchability . 7))
```

Una consecuencia de esto (posiblemente no intencionada) es la eliminación de cualquier valor estándar que se establezca en un archivo de inicio y que se carga cada vez que se compila un archivo de entrada. En el ejemplo anterior, los ajustes estándar para `padding` y `minimum-distance` (definidos en `scm/define-grobs.scm`) se reinician a sus valores predeterminados si no se han fijado (cero para las dos claves). La definición de una propiedad o variable como una lista-A (de cualquier tamaño) siempre reinicia todos los valores de clave no establecidos a sus valores predeterminados si no se han fijado. Ano ser que este sea el resultado deseado, es más seguro actualizar los valores de clave individualmente con una declaración anidada.

**Nota:** Las declaraciones anidadas no funcionan para las listas-A de propiedades de contexto (como `beamExceptions`, `keyAlterations`, `timeSignatureSettings`, etc.). Estas propiedades sólo se pueden modificar redefiniéndolas completamente como listas-A.

## 5.4 Conceptos y propiedades útiles

### 5.4.1 Modos de entrada

La forma en que se interpreta la notación contenida dentro de un archivo de entrada, está determinada por el modo de entrada en curso. In general, there are two ways of specifying the mode: a long form, e.g. `\chordmode`, and a short form, e.g. `\chords`. The long form is typically used when supplying input to a variable or when entering input directly into an explicitly created context. The short form implicitly creates a context of the correct type for the input and passes the input directly to it. It is useful in simple situations when there is no requirement to explicitly create the receiving context.

#### *Modo de acordes*

Se activa con la instrucción `\chordmode` y produce que la entrada se interprete con al sintaxis de la notación de acordes, véase Sección 2.7 [Notación de acordes], página 442. Music in chord mode is rendered as chords on a staff when entered into a **Staff** context, as chord names when entered into a **ChordNames** context or as fret boards when entered into a **FretBoards** context.

El modo de acordes se activa también con la instrucción `\chords`. Esto también produce que el código que sigue se interprete con la sintaxis de la notación de acordes but in addition it implicitly creates a new **ChordNames** context and renders the input into it as chord names, véase [Impresión de los nombres de acorde], página 448.

#### *Modo de percusión*

Se activa con la instrucción `\drummode` y produce que el código de entrada se interprete con la sintaxis de la notación de percusión, véase [Notación básica de percusión], página 420. Music in drum mode is rendered as percussion notes when entered into a **DrumStaff** context.

El modo de percusión también se activa con la instrucción `\drums`. También hace que el código que sigue se interprete con la sintaxis de la notación de percusión but in addition it implicitly creates a new **DrumStaff** context and renders the input into it as percussion notes, véase [Notación básica de percusión], página 420.

#### *Modo de cifras*

Se activa con la instrucción `\figuremode` y hace que el código de entrada se interprete con la sintaxis del bajo cifrado, véase [Introducir el bajo cifrado], página 457. Music in figure mode is rendered as figured bass when entered into a **FiguredBass** context or a **Staff** context.

El modo de cifras también se activa con la instrucción `\figures`. También hace que el código que viene a continuación se interprete con la sintaxis del bajo cifrado but in addition it implicitly creates a new `FiguredBass` context and renders the input into it as figured bass, véase [Introducción al bajo cifrado], página 456.

### *Modos de traste y tablatura*

No existen modos de entrada especiales para introducir símbolos de trastes y de tablatura.

Para crear diagramas de trastes, escriba las notas o acordes en el modo de notas e imprímalos dentro de un contexto `TabStaff`, véase [Tablaturas predeterminadas], página 371.

Para crear diagramas de trastes encima de un pentagrama, enter notes or chords in either note mode or chord mode and render them in a `FretBoards` context, véase [Diagramas de traste automáticos], página 411. Alternatively, los diagramas de trastes se pueden introducir como elementos de marcado encima de las notas utilizando la instrucción `\fret-diagram`, véase [Marcas de diagramas de trastes], página 390.

### *Modo de letra*

Se activa con la instrucción `\lyricmode`, y hace que la entrada se interprete como sílabas de la letra de la canción con duraciones opcionales y modificadores de letra asociados, véase Sección 2.1 [Música vocal], página 286. Input in lyric mode is rendered as lyric syllables when entered into a `Lyrics` context.

El modo de letra también se habilita con la instrucción `\addlyrics`. This also causes the following input to be interpreted as lyric syllables but in addition it implicitly creates a new `Lyrics` context and renders the input into it as lyric syllables.

Lyric mode is also activated with the `\addlyrics` command. Esto también crea un contexto `Lyrics` nuevo y además añade una instrucción `\lyricsto` implícita que asocia la letra que viene a continuación con la música precedente, véase `\lyricsto` [Introducción], página 371.

### *Modo de marcado*

Se activa con la instrucción `\markup`, y hace que la entrada se interprete con la sintaxis del marcado, véase `\markup` [Introducción], página 371.

### *Modo de notas*

Es el modo predeterminado o se puede activar con la instrucción `\notemode`. La entrada se interpreta como alturas, duraciones, marcado, etc. y se imprime como notación musical sobre un pentagrama.

Normalmente no es necesario especificar el modo de notas de forma explícita, pero puede ser útil hacerlo en ciertas situaciones, por ejemplo si estamos en el modo de letra, en el modo de acordes o en otro modo y queremos insertar algo que solamente se puede hacer con la sintaxis del modo de notas.

## 5.4.2 Dirección y posición

Al tipografiar música, la dirección y colocación de muchos elementos es cuestión de elección. Por ejemplo, las plicas de las notas se pueden dirigir hacia arriba o hacia abajo; la letra, las indicaciones dinámicas y otras marcas expresivas se pueden colocar encima o debajo del pentagrama; el texto se puede alinear a la izquierda, a la derecha o centrado; etc. La mayoría de estas elecciones pueden dejarse que LilyPond las determine automáticamente, pero en ciertos casos puede ser deseable forzar una dirección o colocación concreta.

## Indicadores de dirección de las articulaciones

De forma predeterminada algunas direcciones siempre son hacia arriba o siempre hacia abajo (p. ej. los matices o el calderón), mientras que otras cosas pueden alternar entre arriba y abajo en función de la dirección de las plicas (como las ligaduras o los acentos).

Se puede sobrescribir la acción predeterminada mediante el prefijado de la articulación por un *indicador de dirección*. Están disponibles tres indicadores de dirección: `^` (que significa “arriba”), `_` (que significa “abajo”) o `-` (que significa “usar la dirección predeterminada”). El indicador de dirección se puede normalmente omitir, en cuyo caso se supone el indicador predeterminado `-`, pero se necesita un indicador de dirección **siempre** antes de:

- las instrucciones `\tweak`
- las instrucciones `\markup`
- las instrucciones `\tag`
- los marcados de cadena, p.ej. `-"cadena"`
- las instrucciones de digitación, p.ej. `-1`
- las abreviaturas de articulación, p.ej. `-. , -> , --`

Estas indicaciones afectan sólo a la nota siguiente.

```
\relative {
  c' '2( c)
  c2_( c)
  c2( c)
  c2^( c)
}
```



## La propiedad de dirección

La posición o dirección de muchos objetos de presentación está controlada por la propiedad `direction`.

El valor de la propiedad `direction` se puede establecer al valor `1`, con el significado de “hacia arriba” o “encima”, o a `-1`, con el significado de “hacia abajo” o “debajo”. Se pueden usar los símbolos `UP` y `DOWN` en sustitución de `1` y `-1` respectivamente. La dirección predeterminada se puede especificar estableciendo `direction` a `0` ó a `CENTER`. De forma alternativa, en muchos casos existen instrucciones predefinidas para especificar la dirección. Todas ellas son de la forma:

```
\xxxUp, \xxxDown o \xxxNeutral
```

donde `\xxxNeutral` significa “utilizar la dirección predeterminada”. Véase Sección “Objetos interiores al pentagrama” en *Manual de Aprendizaje*.

En alguna que otra ocasión como en el arpeggio, el valor de la propiedad `direction` puede especificar si el objeto se debe colocar a la izquierda o a la derecha del objeto padre. En este caso `-1` ó `LEFT` significan “a la izquierda” y `1` ó `RIGHT` significan “a la derecha”. `0` ó `CENTER` significan “utilizar la dirección predeterminada”.

Estas indicaciones afectan a todas las notas hasta que son canceladas.

```
\relative {
  c' '2( c)
  \slurDown
  c2( c)
  c2( c)
```

```
\slurNeutral
c2( c)
}
```



En música polifónica, en general es mejor especificar una voz explícita que cambiar la dirección de un objeto. Para ver más información, véase Sección 1.5.2 [Varias voces], página 181.

## Véase también

Manual de aprendizaje: Sección “Objetos interiores al pentagrama” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 1.5.2 [Varias voces], página 181.

### 5.4.3 Distancias y medidas

Las distancias en LilyPond son de dos tipos: absolutas y escaladas.

Las distancias absolutas se usan para especificar márgenes, sangrados y otros detalles de diseño de página, y de forma predeterminada se especifican en milímetros. Las distancias se pueden especificar en otras unidades escribiendo después de la cifra indicativa de la cantidad, `\mm`, `\cm`, `\in` (pulgadas), o `\pt` (puntos, 1/72.27 pulgadas). Las distancias de diseño de página se pueden especificar también en unidades escalables (véase el párrafo siguiente) adjuntando `\staff-space` a la cantidad. La disposición de página se describe en detalle en Sección 4.1 [Disposición de la página], página 567.

Las distancias escaladas siempre se especifican en unidades de un espacio del pentagrama o, más raramente, medio espacio del pentagrama. El espacio de pentagrama es la distancia entre dos líneas del pentagrama adyacentes. El valor predeterminado se puede cambiar globalmente fijando el tamaño global del pentagrama, o se puede sobrescribir localmente cambiando la propiedad `staff-space` del objeto `StaffSymbol`. Las distancias escaladas se escalan automáticamente con cualquier cambio al tamaño global del pentagrama o a la propiedad `staff-space` del objeto `StaffSymbol`, pero las fuentes tipográficas se escalan solamente con los cambios efectuados al tamaño global del pentagrama. Así, el tamaño global del pentagrama posibilita la fácil variación del tamaño general de una partitura impresa. Para ver los métodos de establecimiento del tamaño global del pentagrama, véase Sección 4.2.2 [Establecer el tamaño del pentagrama], página 581.

Si se necesita dibujar sólo una sección de una partitura a una escala distinta, por ejemplo una sección ossia o una nota al pie, no se puede simplemente cambiar el tamaño global del pentagrama porque esto afectaría a toda la partitura. En tales casos, el cambio de tamaño se hace sobrescribiendo tanto la propiedad `staff-space` de `StaffSymbol` como el tamaño de las fuentes tipográficas. Está a nuestra disposición una función de Scheme, `magstep`, para convertir de un cambio en el tamaño de la fuente al cambio equivalente en `staff-space`. Para ver una explicación y un ejemplo de su utilización, consulte Sección “Longitud y grosor de los objetos” en *Manual de Aprendizaje*.

## Véase también

Manual de aprendizaje: Sección “Longitud y grosor de los objetos” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 4.1 [Disposición de la página], página 567, Sección 4.2.2 [Establecer el tamaño del pentagrama], página 581.

### 5.4.4 Dimensiones

Las dimensiones de un objeto gráfico especifican las posiciones de los límites izquierdo y derecho, así como los límites inferior y superior de las cajas limítrofes de los objetos como distancias a partir del punto de referencia del objeto en unidades de espacios de pentagrama. Estas posiciones se codifican frecuentemente como dos parejas de valores de Scheme. Por ejemplo, la instrucción de marcado de texto `\with-dimensions` toma tres argumentos, de los cuales los dos primeros son una pareja de Scheme que da las posiciones de los bordes izquierdo y derecho y otra pareja de Scheme que da las posiciones de los límites inferior y superior:

```
\with-dimensions #'(-5 . 10) #'(-3 . 15) arg
```

Esto especifica una caja circundante para *arg* con su límite izquierdo en -5, su límite derecho en 10, el inferior en -3 y el superior en 15, todos ellos medidos a partir del punto de referencia del objeto en unidades de espacios de pentagrama.

### Véase también

Referencia de la notación: Sección 5.4.3 [Distancias y medidas], página 663.

### 5.4.5 Propiedades del símbolo del pentagrama

Se puede definir al mismo tiempo la posición vertical de las líneas de la pauta y el número de líneas de la misma. Como muestra el siguiente ejemplo, las posiciones de las notas no están influidas por las posiciones de las líneas de la pauta.

**Nota:** La propiedad `'line-positions` sobrescribe a la propiedad `'line-count`. El número de líneas de la pauta está definido implícitamente por el número de elementos de la lista de valores de `'line-positions`.

```
\new Staff \with {
  \override StaffSymbol.line-positions = #'(7 3 0 -4 -6 -7)
}
\relative { a4 e' f b | d1 }
```



Se puede modificar la anchura de la pauta. Las unidades son espacios de pentagrama. El espaciado de los objetos dentro del pentagrama no resulta afectado por este ajuste.

```
\new Staff \with {
  \override StaffSymbol.width = #23
}
\relative { a4 e' f b | d1 }
```



### 5.4.6 Objetos de extensión

Muchos objetos de notación musical abarcan varias notas o incluso varios compases. Son ejemplos los crescendi, trinos, corchetes de grupo especial y corchetes de primera y segunda vez. Estos objetos se llaman “spanners” u «objetos de extensión», y tienen propiedades especiales para

controlar su apariencia y comportamiento. Algunas de estas propiedades son comunes a todos los objetos de extensión; otras se limitan a un subconjunto de los extensores.

Todos los objetos de extensión contemplan el interface `spanner-interface`. Algunos, básicamente aquellos que trazan una línea recta entre los dos objetos, contemplan también el interface `line-spanner-interface`.

## Uso del `spanner-interface`

Este interface proporciona dos propiedades que se aplican a varios extensores.

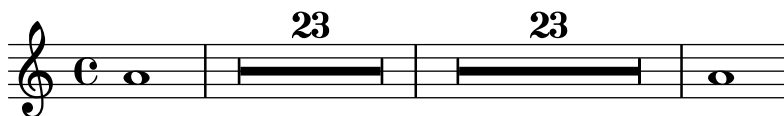
### *La propiedad `minimum-length` (longitud mínima)*

La longitud mínima del objeto de extensión se especifica a través de la propiedad `minimum-length`. Su aumento suele producir el efecto necesario de aumentar el espaciado de las notas entre los dos puntos extremos. Sin embargo, esta sobreescritura no tiene ningún efecto sobre muchos extensores, pues su longitud está determinada por otras consideraciones. Más abajo se muestran algunos ejemplos de dónde es efectiva.

```
a' ~ a'
a'
% increase the length of the tie
-\tweak minimum-length #5
~ a'
```



```
\relative \compressMMRests {
  a'1
  R1*23
  % increase the length of the rest bar
  \once \override MultiMeasureRest.minimum-length = #20
  R1*23
  a1
}
```



```
\relative {
  a' \< a a a \!
  % increase the length of the hairpin
  \override Hairpin.minimum-length = #20
  a \< a a a \!
}
```



Esta sobreescritura se puede usar también para aumentar la longitud de las ligaduras de expresión y de fraseo:

```
\relative {
```

```

a' ( g)
a
-\tweak minimum-length #5
( g)

a\ ( g\ )
a
-\tweak minimum-length #5
\ ( g\ )
}

```



Para algunos objetos de presentación, la propiedad `minimum-length` es efectiva sólo si se llama explícitamente al procedimiento `set-spacing-rods`. Para hacerlo, se debe fijar la propiedad `springs-and-rods` al valor `ly:spanner::set-spacing-rods`. Por ejemplo, la longitud mínima de un glissando no tiene efecto a no ser que se establezca la propiedad `springs-and-rods`:

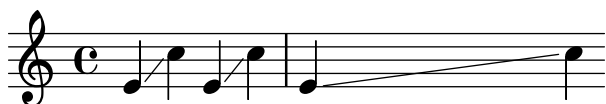
```

% default
e' \glissando c''

% not effective alone
\once \override Glissando.minimum-length = #20
e' \glissando c''

% effective only when both overrides are present
\once \override Glissando.minimum-length = #20
\once \override Glissando.springs-and-rods = #ly:spanner::set-spacing-rods
e' \glissando c''

```



Lo mismo se puede decir del objeto `Beam`:

```

% not effective alone
\once \override Beam.minimum-length = #20
e'8 e' e' e'

% effective only when both overrides are present
\once \override Beam.minimum-length = #20
\once \override Beam.springs-and-rods = #ly:spanner::set-spacing-rods
e'8 e' e' e'

```

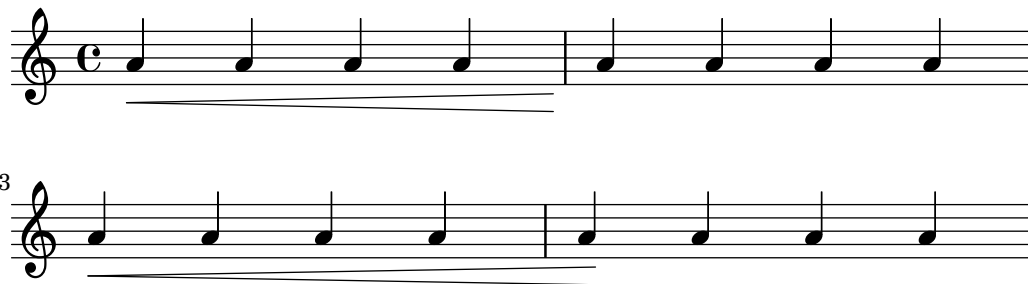


### *La propiedad to-barline*

La segunda propiedad útil del `spanner-interface` es `to-barline`. De forma predeterminada tiene el valor cierto, haciendo que los reguladores y otros objetos de extensión que terminan

sobre la primera nota de un compás, en vez de eso terminen en la línea divisoria inmediatamente precedente. Si se establece al valor falso, el extensor llegará más allá de la barra de compás y terminará exactamente sobre la nota:

```
\relative {
  a' \< a a a a \! a a a \break
  \override Hairpin.to-barline = ##f
  a \< a a a a \! a a a
}
```



Esta propiedad no es efectiva para todos los extensores. Por ejemplo, su establecimiento a `##t` no tienen ningún efecto sobre las ligaduras de expresión o de fraseo, o sobre otros extensores para los que terminar en la barra de compás no tendría ningún significado.

## Uso del line-spanner-interface

Entre los objetos que contemplan el interface `line-spanner-interface` se encuentran

- `DynamicTextSpanner`
- `Glissando`
- `TextSpanner`
- `TrillSpanner`
- `VoiceFollower`

La rutina responsable de dibujar los sellos de estos extensores es `ly:line-spanner::print`. esta rutina determina la localización exacta de los dos puntos extremos y traza una línea entre ellos, en el estilo solicitado. Las posiciones de los dos puntos extremos del extensor se calculan al vuelo, pero es posible sobrescribir sus coordenadas Y. Las propiedades que se deben especificar están anidadas a dos niveles de profundidad en la jerarquía de propiedades, pero la sintaxis de la instrucción `\override` es bastante sencilla:

```
e''2 \glissando b'
\once \override Glissando.bound-details.left.Y = #3
\once \override Glissando.bound-details.right.Y = #-2
e''2 \glissando b'
```



Las unidades para la propiedad Y son `staff-spaces`, siendo el punto del cero la línea central del pentagrama. Para el glissando, esto es el valor de Y en la coordenada X que corresponde al punto central de cada cabeza de nota si nos imaginamos que la línea se extiende hasta allí.

Si no está fijado Y, su valor se calcula a partir de la posición vertical del punto de anclaje correspondiente del extensor.



En caso de salto de línea, los valores para los puntos extremos se especifican por las sub-listas `left-broken` y `right-broken` de `bound-details`. Por ejemplo:

```
\override Glissando.breakable = ##t
\override Glissando.bound-details.right-broken.Y = #-3
c''1 \glissando \break
f''1
```



Un número de propiedades adicionales de las sub-listas `left` y `right` de la propiedad `bound-details` se pueden especificar de la misma forma que `Y`:

**Y** Establece la coordenada Y del punto extremo, en desplazamientos de `staff-spaces` desde la línea central del pentagrama. De forma predeterminada es el centro del objeto ancla, y así un glissando apunta al centro vertical de la cabeza de la nota. Para extensores horizontales como los extensores de texto y los trinos, está inamoviblemente codificado como 0.

**attach-dir** (dirección de anclaje)

Determina dónde comienza y termina la línea en la dirección X, con relación al objeto ancla. Sí, un valor de -1 (o `LEFT`, izquierda) hace que la línea comience o termine en el lado izquierdo de la cabeza de la nota a la que está anclado.

**X** Es la coordenada X absoluta del punto extremo. Se suele calcular al vuelo, y su sobreescritura no tiene un efecto útil.

**stencil** (sello)

Los extensores de línea pueden tener símbolos al comienzo o al final, lo que está contenido en esta sub-propiedad. Esto es para uso interno; se recomienda en su lugar el uso de `text`.

**text** (texto)

Es un elemento de marcado que se evalúa para dar lugar al sello. Se usa para escribir *cresc.*, *tr* y otros textos sobre los objetos de extensión horizontales.

```
\override TextSpanner.bound-details.left.text
= \markup { \small \bold Slower }
\relative { c''2\startTextSpan b c a\stopTextSpan }
```



**stencil-align-dir-y** (alineación del sello en y)

**stencil-offset** (desplazamiento del sello)

Si no se modifican uno u otro, el sello se coloca sencillamente en el punto extremo, centrado sobre la línea, como viene definido por las subpropiedades `X` e `Y`. Si se fijan `stencil-align-dir-y` o `stencil-offset` se moverá el símbolo del borde verticalmente con relación al extremo de la línea:

```
\override TextSpanner.bound-details.left.stencil-align-dir-y = #-2
```

```

\override TextSpanner.bound-details.right.stencil-align-dir-y = #UP

\override TextSpanner.bound-details.left.text = "ggg"
\override TextSpanner.bound-details.right.text = "hhh"

\relative { c'4^\startTextSpan c c c \stopTextSpan }

```



Observe que los valores negativos mueven el texto *hacia arriba*, al contrario de lo que podría esperarse, pues el valor de `-1` o `DOWN` (abajo) significa alinear el borde *inferior* del texto con la línea de extensión. Un valor de `1` o `UP` (arriba) alinea el borde superior del texto con la línea extensora.

#### arrow (flecha)

Al establecer esta sub-propiedad a `#t` se produce una punta de flecha en el extremo de la línea.

#### padding (relleno)

Esta sub-propiedad controla el espacio entre el punto extremo especificado de la línea y el extremo real. Sin relleno, un glissando empezaría y terminaría en el centro de la cabeza de las notas.

La función musical `\endSpanners` finaliza de forma prematura el extensor que comienza sobre la nota que sigue inmediatamente a continuación. Se termina después de una nota exactamente, o en la siguiente barra de compás si `to-barline` es verdadero y se produce una divisoria antes de la siguiente nota.

```

\relative c' {
  \endSpanners
  c2 \startTextSpan c2 c2
  \endSpanners
  c2 \< c2 c2
}

```



Si se usa `\endSpanners` no es necesario cerrar `\startTextSpan` con `\stopTextSpan`, ni cerrar los reguladores con `\!`.

## Véase también

Referencia de funcionamiento interno: Sección “TextSpanner” en *Referencia de Funcionamiento Interno*, Sección “Glissando” en *Referencia de Funcionamiento Interno*, Sección “VoiceFollower” en *Referencia de Funcionamiento Interno*, Sección “TrillSpanner” en *Referencia de Funcionamiento Interno*, Sección “line-spanner-interface” en *Referencia de Funcionamiento Interno*.

### 5.4.7 Visibilidad de los objetos

Hay cuatro formas principales en que se puede controlar la visibilidad de los objetos de presentación: se puede eliminar su sello, se pueden volver transparentes, se pueden pintar de blanco, o se puede sobrescribir su propiedad `break-visibility`. Las tres primeras se aplican a todos los objetos de presentación; la última sólo a unos pocos: los objetos *divisibles*. El Manual de aprendizaje introduce estas cuatro técnicas, véase Sección “Visibilidad y color de los objetos” en *Manual de Aprendizaje*.

Hay también algunas otras técnicas que son específicas de ciertos objetos de presentación. Se estudian bajo Consideraciones especiales.

#### Eliminar el sello

Todo objeto de presentación tiene una propiedad `stencil` (sello). De forma predeterminada está establecida a la función específica que dibuja ese objeto. Si se sobrescribe esta propiedad a `#f` no se llama a ninguna función y el objeto no se dibuja. La acción predeterminada se puede recuperar con `\revert`.

```
a1 a
\override Score.BarLine.stencil = ##f
a a
\revert Score.BarLine.stencil
a a a
```



Esta operación, bastante común, tiene la forma abreviada `\omit` como atajo:

```
a1 a
\omit Score.BarLine
a a
\undo \omit Score.BarLine
a a a
```



#### Hacer transparentes los objetos

Todo objeto de presentación tiene una propiedad `transparent` (transparente) que de forma predeterminada está establecida a `#f`. Si se fija a `#t` el objeto aún ocupa espacio pero es invisible.

```
a'4 a'
\once \override NoteHead.transparent = ##t
a' a'
```



Esta operación, bastante común, tiene la forma abreviada `\hide` como atajo:

```
a'4 a'
\once \hide NoteHead
```

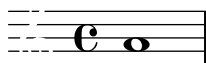
a' a'



## Pintar los objetos de blanco

Todo objeto de presentación tiene una propiedad de color que de forma predeterminada está establecida a **black** (negro). Si se sobrescribe a **white** (blanco) el objeto será indistinguible del fondo blanco. Sin embargo, si el objeto cruza a otros objetos, el color de los puntos de cruce queda determinado por el orden en que se dibujan estos objetos, lo que puede dejar una imagen fantasma del objeto blanco, como puede verse aquí:

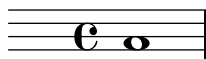
```
\override Staff.Clef.color = #white
a'1
```



Se puede evitar esto cambiando el orden de impresión de los objetos. Todos los objetos de presentación tienen una propiedad **layer** (capa) que se debe establecer a un valor entero. Los objetos con el valor de **layer** más bajo se dibujan primero, después se dibujan los objetos con valores progresivamente mayores, y así los objetos con valores más altos se dibujan encima de los que tienen valores más bajos. De forma predeterminada, casi todos los objetos tienen asignado un valor **layer** de 1, aunque algunos objetos, entre ellos el pentagrama y las líneas divisorias, **StaffSymbol** y **BarLine**, tienen asignado un valor de 0. El orden de impresión de los objetos con el mismo valor de **layer** es indeterminado.

En el ejemplo de arriba, la clave blanca, con un valor **layer** predeterminado de 1, se dibuja después de las líneas del pentagrama (valor **layer** predeterminado de 0), sobreimpresionándolas. Para cambiarlo, se debe dar al objeto **Clef** un valor de **layer** más bajo, digamos -1, para que se dibuje antes:

```
\override Staff.Clef.color = #white
\override Staff.Clef.layer = #-1
a'1
```



## Uso de break-visibility (visibilidad en el salto)

Casi todos los objetos de presentación se imprimen una sola vez, pero algunos como las líneas divisorias, claves, indicaciones de compás y armaduras de tonalidad, se pueden tener que imprimir dos veces cuando se produce un salto de línea: una vez al final de la línea y otra al comienzo de la siguiente. Estos objetos reciben el nombre de *divisibles*, y tienen una propiedad, **break-visibility** (visibilidad en el salto), para controlar su visibilidad en las tres posiciones en que pueden aparecer: al comienzo de una línea, dentro de la línea si se produce un cambio, y al final de la línea si el cambio se produce en ese lugar.

Por ejemplo, la indicación de compás se imprime de forma predeterminada al comienzo de la primera línea y en ningún otro lugar a no ser que cambie, en cuyo caso se imprime en el punto en que se produce el cambio. Si este cambio se produce al final de una línea, la nueva indicación de compás se imprime al principio de la línea siguiente y también al final de la línea anterior como indicación de precaución.

Este comportamiento se controla por medio de la propiedad **break-visibility**, que se explica en Sección “Visibilidad y color de los objetos” en *Manual de Aprendizaje*. Esta propiedad toma un vector de tres valores booleanos que, por orden, determinan si el objeto se imprime al final, dentro, o al principio de la línea. O, para ser más exactos: antes del salto de línea, si no hay salto, o después del salto.

Como alternativa se puede especificar cualquiera de las ocho combinaciones mediante funciones predefinidas cuya definición está en `scm/output-lib.scm`, donde las tres últimas columnas indican si los objetos de presentación serán visibles en las posiciones que se muestran en el encabezamiento de cada columna:

Forma de función	Forma de vector	Antes del salto	Si no hay salto	Después del salto
<code>all-visible</code>	<code>##(t t t)</code>	sí	sí	sí
<code>begin-of-line-visible</code>	<code>##(f f t)</code>	no	no	sí
<code>center-visible</code>	<code>##(f t f)</code>	no	sí	no
<code>end-of-line-visible</code>	<code>##(t f f)</code>	sí	no	no
<code>begin-of-line-invisible</code>	<code>##(t t f)</code>	sí	sí	no
<code>center-invisible</code>	<code>##(t f t)</code>	sí	no	sí
<code>end-of-line-invisible</code>	<code>##(f t t)</code>	no	sí	sí
<code>all-invisible</code>	<code>##(f f f)</code>	no	no	no

Los ajustes predeterminados de **break-visibility** dependen del objeto de presentación. La tabla siguiente muestra todos los objetos de presentación de interés que resultan afectados por **break-visibility** y el ajuste predeterminado de esta propiedad:

Objeto	Contexto usual	Valor predet.
<code>BarLine</code>	<code>Score</code>	calculado
<code>BarNumber</code>	<code>Score</code>	<code>begin-of-line-visible</code>
<code>BreathingSign</code>	<code>Voice</code>	<code>begin-of-line-invisible</code>
<code>Clef</code>	<code>Staff</code>	<code>begin-of-line-visible</code>
<code>Custos</code>	<code>Staff</code>	<code>end-of-line-visible</code>
<code>DoublePercentRepeat</code>	<code>Voice</code>	<code>begin-of-line-invisible</code>
<code>KeyCancellation</code>	<code>Staff</code>	<code>begin-of-line-invisible</code>
<code>KeySignature</code>	<code>Staff</code>	<code>begin-of-line-visible</code>
<code>ClefModifier</code>	<code>Staff</code>	<code>begin-of-line-visible</code>
<code>RehearsalMark</code>	<code>Score</code>	<code>end-of-line-invisible</code>
<code>TimeSignature</code>	<code>Staff</code>	<code>all-visible</code>

El ejemplo de abajo muestra el uso de la forma de vector para controlar la visibilidad de las líneas divisorias:

```
\relative {
  f'4 g a b
  f4 g a b
  % Remove bar line at the end of the current line
  \once \override Score.BarLine.break-visibility = ##(f t t)
  \break
  f4 g a b
  f4 g a b
}
```



Aunque los tres componentes del vector utilizado para sobrescribir **break-visibility** deben estar presentes, no todos son efectivos para todos los objetos de presentación, y algunas combinaciones pueden incluso dar errores. Son de aplicación las siguientes limitaciones:

- Las líneas divisorias no se pueden imprimir al principio de la línea.
- No se puede imprimir el número de compás al principio de la *primera* línea a no ser que su valor establecido sea distinto de 1.
- Clave: véase la sección siguiente.
- Las repeticiones dobles de tipo porcentaje se imprimen todas siempre o bien no se imprimen nunca. Utilice **begin-of line-invisible** para imprimirlas y **all-invisible** para suprimirlas.
- Armadura: véase la sección siguiente.
- ClefModifier: véase la sección siguiente.

## Consideraciones especiales

### *Visibilidad después de un cambio explícito*

La propiedad **break-visibility** controla la visibilidad de las armaduras y cambios de clave sólo al principio de las líneas, es decir, después de un salto. No tiene ningún efecto sobre la visibilidad de la armadura o la clave después de un cambio explícito de tonalidad o de clave dentro o al final de una línea. En el ejemplo siguiente la armadura que sigue al cambio explícito de tonalidad a Si bemol mayor es visible incluso con **all-invisible** establecido.

```
\relative {
  \key g \major
  f'4 g a b
  % Try to remove all key signatures
  \override Staff.KeySignature.break-visibility = #all-invisible
  \key bes \major
  f4 g a b
  \break
  f4 g a b
  f4 g a b
}
```



La visibilidad de estos cambios explícitos de tonalidad y de clave se controla por medio de las propiedades `explicitKeySignatureVisibility` y `explicitClefVisibility`. Son los equivalentes de la propiedad `break-visibility` y las dos toman un vector de tres valores booleanos o las funciones predefinidas relacionadas anteriormente, exactamente igual que `break-visibility`. Las dos son propiedades del contexto `Staff`, no de los propios objetos de presentación, y por tanto se establecen utilizando la instrucción `\set`. Las dos están establecidas de forma predeterminada al valor `all-visible`. Estas propiedades controlan sólo la visibilidad de las armaduras y las claves que resultan de los cambios explícitos y no afectan a las armaduras y tonalidades que están al principio de las líneas; para quitarlas, aún se debe sobrescribir la propiedad `break-visibility` en el objeto correspondiente.

```
\relative {
  \key g \major
  f'4 g a b
  \set Staff.explicitKeySignatureVisibility = #all-invisible
  \override Staff.KeySignature.break-visibility = #all-invisible
  \key bes \major
  f4 g a b \break
  f4 g a b
  f4 g a b
}
```



### *Visibilidad de las alteraciones de precaución*

Para eliminar las alteraciones de cancelación que se imprimen en un cambio de tonalidad explícito, establezca la propiedad `printKeyCancellation` del contexto `Staff` a `#f`:

```
\relative {
  \key g \major
  f'4 g a b
  \set Staff.explicitKeySignatureVisibility = #all-invisible
  \set Staff.printKeyCancellation = #f
  \override Staff.KeySignature.break-visibility = #all-invisible
  \key bes \major
  f4 g a b \break
  f4 g a b
  f4 g a b
}
```



Con estas sobreescrituras solamente permanecen las alteraciones accidentales delante de las notas para indicar el cambio de tonalidad.

Observe que al modificar la tonalidad a Do mayor o La menor, las alteraciones de cancelación serían *la única* indicación del cambio de armadura. En este caso, el establecimiento de `printKeyCancellation` al valor `#f` no tiene ningún efecto:

```
\relative {
  \key g \major
  f'4 g a b
  \set Staff.explicitKeySignatureVisibility = #all-invisible
  \set Staff.printKeyCancellation = ##f
  \key c \major
  f4 g a b \break
  f4 g a b
  f4 g a b
}
```



En lugar de ello, para suprimir las alteraciones de cancelación aún cuando la armadura cambia a Do mayor o a La menor, sobreescriba la visibilidad del grob `KeyCancellation`:

```
\relative {
  \key g \major
  f'4 g a b
  \set Staff.explicitKeySignatureVisibility = #all-invisible
  \override Staff.KeyCancellation.break-visibility = #all-invisible
  \key c \major
  f4 g a b \break
  f4 g a b
  f4 g a b
}
```



### *Líneas divisorias automáticas*

Como caso especial, la impresión de las líneas divisorias también se puede inhabilitar estableciendo la propiedad `automaticBars` en el contexto `Score`. Si se fija a `#f`, las barras de compás no se imprimen automáticamente; se deben crear explícitamente con una instrucción `\bar`. A diferencia de la instrucción predefinida `\cadenzaOn`, los compases se siguen contando. La generación de compases continúa de acuerdo a esta cuenta si esta propiedad se establece posteriormente a `#t`. Si se fija al valor `#f`, sólo pueden producirse saltos de línea en instrucciones `\bar` explícitas.



### Claves transportadas

El pequeño símbolo de transposición sobre una clave se produce por parte del objeto `ClefModifier`. Su visibilidad se hereda automáticamente de la del objeto `Clef`, así que no es necesario aplicar las sobreescrituras correspondientes `break-visibility` a los objetos `ClefModifier` para suprimir los símbolos de transposición u octavación para las claves invisibles.

Para los cambios de clave explícitos, la propiedad `explicitClefVisibility` controla tanto el símbolo de clave como el símbolo de transposición o de octava asociado.

### Véase también

Manual de aprendizaje: Sección “Visibilidad y color de los objetos” en *Manual de Aprendizaje*

### 5.4.8 Estilos de línea

Ciertas indicaciones de ejecución, p.ej., *rallentando*, *accelerando* y los *trinos* se escriben como texto y se extienden sobre muchos compases mediante líneas, a veces punteadas u onduladas.

Todos ellos usan las mismas rutinas que el glissando para dibujar los textos y las líneas, y por ello el ajuste fino de su comportamiento se hace de la misma manera. Se hace con un `spanner` (un objeto de extensión), y la rutina responsable de dibujar los objetos de extensión es `ly:line-spanner::print`. Esta rutina determina la colocación exacta de los dos *extremos del objeto de extensión* y dibuja una línea entre ellos, en el estilo solicitado.

He aquí un ejemplo de los distintos estilos de línea disponibles, y cómo aplicarles ajustes finos.

```
\relative {
  d''2 \glissando d'2
  \once \override Glissando.style = #'dashed-line
  d,2 \glissando d'2
  \override Glissando.style = #'dotted-line
  d,2 \glissando d'2
  \override Glissando.style = #'zigzag
  d,2 \glissando d'2
  \override Glissando.style = #'trill
  d,2 \glissando d'2
}
```



Las posiciones de los puntos extremos del objeto de extensión se computan al vuelo para cada uno de los objetos gráficos, pero es posible sobreescribirlos:

```
\relative {
  e''2 \glissando f
  \once \override Glissando.bound-details.right.Y = #-2
  e2 \glissando f
}
```



El valor de `Y` está establecido a `-2` para el extremo derecho. El lado izquierdo se puede ajustar de forma similar especificando `left` en vez de `right`.

Si no está establecido `Y`, el valor se computa a partir de la posición vertical de los puntos de anclaje izquierdo y derecho del extensor.

Son posibles otros ajustes de los extensores, para ver más detalles consulte Sección 5.4.6 [Objetos de extensión], página 664.

### 5.4.9 Rotación de objetos

Tanto los objetos de presentación como los elementos de texto de marcado se pueden girar cualquier ángulo respecto a cualquier punto, pero difiere el método de hacerlo.

#### Rotación de objetos de presentación

Todos los objetos de presentación que contemplan el interface `grob-interface` se pueden rotar estableciendo su propiedad `rotation`. Acepta una lista de tres elementos: el ángulo de la rotación en sentido antihorario, y las coordenadas `x` e `y` del punto con relación al punto de referencia del objeto, alrededor del que se va a realizar la rotación. El ángulo de rotación se especifica en grados y las coordenadas en espacios de pentagrama.

El ángulo de rotación y las coordenadas del punto de rotación se deben determinar por ensayo y error.

Solamente en ciertas ocasiones es útil la rotación de objetos de presentación; el ejemplo siguiente muestra una situación en que puede serlo:

```
g4\< e' d' f'\!
\override Hairpin.rotation = #'(15 -1 0)
g4\< e' d' f'\!
```



#### Rotación de elementos de marcado

Todos los textos de marcado se pueden rotar para que se dispongan en cualquier ángulo prece-diéndolos de la instrucción `\rotate`. La instrucción acepta dos argumentos: el ángulo de rotación en grados en sentido antihorario, y el texto que rotar. Los límites que ocupa el texto no se rotan: toman su valor a partir de los extremos de las coordenadas `x` e `y` del texto rotado. En el ejemplo siguiente la propiedad `outside-staff-priority` del texto se establece a `##f` para desactivar la evitación automática de colisiones, lo que empuja al texto a una posición muy alta.

```
\override TextScript.outside-staff-priority = ##f
g4^\markup { \rotate #30 "a G" }
b^\markup { \rotate #30 "a B" }
des'^\markup { \rotate #30 "a D-Flat" }
fis'^\markup { \rotate #30 "an F-Sharp" }
```



## 5.5 Trucos avanzados

Esta sección trata sobre distintos enfoques en la realización de ajustes finos a la apariencia de la partitura impresa.

### Véase también

Manual de aprendizaje: Sección “Trucar la salida” en *Manual de Aprendizaje*, Sección “Otras fuentes de información” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.2 [Explicación del Manual de referencia de funcionamiento interno], página 642, Sección 5.3 [Modificar las propiedades], página 646.

Manual de extensión: Sección “Interfaces para programadores” en *Extender*.

Archivos de inicio: `scm/define-grobs.scm`.

Fragmentos de código: Sección “Tweaks and overrides” en *Fragmentos de código*.

Referencia de funcionamiento interno: Sección “All layout objects” en *Referencia de Funcionamiento Interno*.

### 5.5.1 Alineación de objetos

Los objetos gráficos que soportan el interface `self-alignment-interface` y/o el `side-position-interface` se pueden alinear contra un objeto colocado previamente, de diversas maneras. Para ver una lista de estos objetos, consulte Sección “self-alignment-interface” en *Referencia de Funcionamiento Interno* y Sección “side-position-interface” en *Referencia de Funcionamiento Interno*.

Todos los objetos gráficos tienen un punto de referencia, una extensión horizontal y una extensión vertical. La extensión horizontal es una pareja de números que dan los desplazamientos a partir del punto de referencia de los bordes izquierdo y derecho, siendo negativos los desplazamientos hacia la izquierda. La extensión vertical es una pareja de números que dan el desplazamiento a partir del punto de referencia hasta los bordes inferior y superior, siendo negativos los desplazamientos hacia abajo.

La posición de un objeto sobre el pentagrama viene dada por los valores de las propiedades `X-offset` e `Y-offset`. El valor de `X-offset` da el desplazamiento desde la coordenada X del punto de referencia del objeto padre, y el valor de `Y-offset` da el desplazamiento a partir de la línea central del pentagrama. Los valores de `X-offset` y `Y-offset` se pueden establecer directamente o se puede dejar que se calculen por parte de procedimientos para conseguir una alineación con el objeto padre.

**Nota:** Muchos objetos tienen consideraciones de posicionamiento especiales que hacen que se ignore o se modifique cualquier ajuste realizado a `X-offset` o a `Y-offset`, a pesar de que el objeto contemple el interface `self-alignment-interface`. La sobreescritura de las propiedades `X-offset` ó `Y-offset` a un valor fijo hace que se descarte la respectiva propiedad `self-alignment`.

Por ejemplo, una alteración accidental se puede repositionar verticalmente estableciendo `Y-offset` pero los cambios a `X-offset` no tienen ningún efecto.

Las letras de ensayo se pueden alinear con objetos divisibles como líneas divisorias, claves, armaduras e indicaciones de compás. Hay propiedades especiales que se encuentran en `break-aligned-interface` para el posicionamiento de las letras de ensayo sobre dichos objetos.

## Véase también

Referencia de la notación: [Uso del `break-alignable-interface`], página 681.

Manual de extensión: Sección “Funciones de callback” en *Extender*.

## Establecer directamente X-offset y Y-offset

Se pueden dar valores numéricos a las propiedades `X-offset` y `Y-offset` de muchos objetos. El ejemplo siguiente muestra tres notas con una digitación en su posición predeterminada y con los valores `X-offset` y `Y-offset` modificados.

```
a'-3
a'
-\tweak X-offset #0
-\tweak Y-offset #0
-3
a'
-\tweak X-offset #-1
-\tweak Y-offset #1
-3
```



## Uso del side-position-interface

Un objeto que contempla el `side-position-interface` se puede colocar junto a su objeto padre de forma que los bordes especificados de los dos objetos se toquen. El objeto se puede situar encima, debajo, a la derecha o a la izquierda del objeto padre. El padre no se puede especificar; viene determinado por el orden de los elementos en el flujo de entrada. Casi todos los objetos tienen la cabeza de la nota asociada como padre.

Los valores de las propiedades `side-axis` y `direction` determinan dónde colocar el objeto, como sigue:

<code>side-axis</code> propiedad	<code>direction</code> propiedad	colocación
0	-1	izquierda
0	1	derecha
1	-1	debajo
1	1	encima

Si `side-axis` es 0, `X-offset` se debe establecer al procedimiento `ly:side-position-interface::x-aligned-side`. Este procedimiento devuelve el valor correcto de `X-offset` para situar el objeto al lado izquierdo o derecho del padre de acuerdo con el valor de `direction`.

Si `side-axis` es 1, `Y-offset` se debe establecer al procedimiento `ly:side-position-interface::y-aligned-side`. Este procedimiento devuelve el valor correcto de `Y-offset` para situar el objeto encima o debajo del padre de acuerdo con el valor de `direction`.

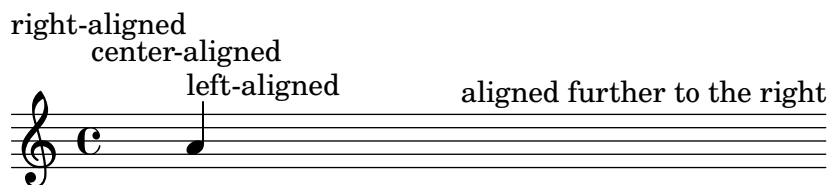
## Uso del self-alignment-interface

### *Auto-alineación horizontal de los objetos*

La alineación horizontal de un objeto que contempla el interface `self-alignment-interface` está controlada por el valor de la propiedad `self-alignment-X`, siempre y cuando la propiedad `X-offset` de este objeto esté establecida a `ly:self-alignment-interface::x-aligned-on-self`. Se le puede dar a `self-alignment-X` cualquier valor real, en unidades de la mitad de la extensión X total del objeto. Los valores negativos mueven el objeto a la derecha, los positivos hacia la izquierda. Un valor de 0 centra el objeto sobre el punto de referencia de su padre, un valor de -1 alinea el borde izquierdo del objeto sobre el punto de referencia de su padre, y un valor de 1 alinea el borde derecho del objeto sobre el punto de referencia de su padre. Se pueden usar los símbolos `LEFT`, `CENTER` y `RIGHT` en sustitución de los valores -1, 0 y 1, respectivamente.

Normalmente se usaría la instrucción `\override` para modificar el valor de `self-alignment-X`, pero se puede usar la instrucción `\tweak` para alinear varias anotaciones por separado sobre una sola nota:

```
a'
-\tweak self-alignment-X #-1
^"left-aligned"
-\tweak self-alignment-X #0
^"center-aligned"
-\tweak self-alignment-X #RIGHT
^"right-aligned"
-\tweak self-alignment-X #-2.5
^"aligned further to the right"
```



### *Auto-alineación vertical de los objetos*

Los objetos se pueden alinear verticalmente en una forma análoga a la alineación horizontal si la propiedad `Y-offset` está establecida a `ly:self-alignment-interface::y-aligned-on-self`. Sin embargo, a menudo se encuentran implicados otros mecanismos en la alineación vertical: el valor de `Y-offset` es tan sólo una variable que se tiene en cuenta. Esto puede hacer que ajustar el valor de ciertos objetos sea una tarea difícil. Las unidades son sólo la mitad de las dimensiones verticales del objeto, que suele ser bastante pequeño, por lo que pueden requerirse números bastante grandes. Un valor de -1 alinea el borde inferior del objeto con el punto de referencia del objeto padre, un valor de 0 alinea el centro del objeto con el punto de referencia del padre, y un valor de 1 alinea el borde superior del objeto con el punto de referencia del padre. Se pueden usar los símbolos `DOWN`, `CENTER`, `UP` en sustitución de -1, 0 y 1, respectivamente.

#### *Auto-alineación de objetos en las dos direcciones*

Estableciendo tanto `X-offset` como `Y-offset`, se puede alinear un objeto en las dos direcciones simultáneamente.

El ejemplo siguiente muestra cómo ajustar una digitación de forma que se acerque a la cabeza de la nota.

```
a'
-\tweak self-alignment-X #0.5 % move horizontally left
-\tweak Y-offset #ly:self-alignment-interface::y-aligned-on-self
-\tweak self-alignment-Y #-1 % move vertically up
```

```
-3 % third finger
```

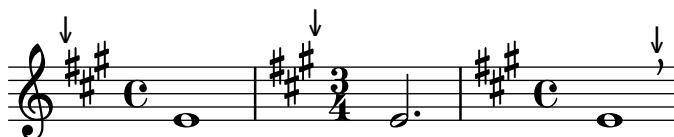


## Uso del break-alignable-interface

Las letras de ensayo se pueden alinear con objetos de notación distintos a las barras de compás. Estos objetos son **ambitus**, **breathing-sign**, **clef**, **custos**, **staff-bar**, **left-edge**, **key-cancellation**, **key-signature** y **time-signature**.

Cada tipo de objeto tiene su propio punto de referencia predeterminado, con el que se alinean las marcas de ensayo:

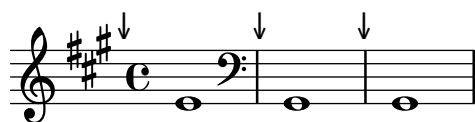
```
% The rehearsal mark will be aligned to the right edge of the Clef
\override Score.RehearsalMark.break-align-symbols = #'(clef)
\key a \major
\clef treble
\mark "↓"
e'1
% The rehearsal mark will be aligned to the left edge of the Time Signature
\override Score.RehearsalMark.break-align-symbols = #'(time-signature)
\key a \major
\clef treble
\time 3/4
\mark "↓"
e'2.
% The rehearsal mark will be centered above the Breath Mark
\override Score.RehearsalMark.break-align-symbols = #'(breathing-sign)
\key a \major
\clef treble
\time 4/4
\mark "↓"
\breath
\mark "↓"
```



Se puede especificar una lista de posibles objetos para la alineación. Si algunos de los objetos son invisibles en ese punto debido al valor de **break-visibility** o a valores de visibilidad explícitos para las armaduras y las claves, la letra de ensayo o número de compás se alinean con el primer objeto de la lista que sea visible. Si ningún objeto de la lista es visible, el objeto se alinea con la línea divisoria. Si la línea divisoria es invisible, el objeto se alinea con el punto en el que se encontraría la línea divisoria.

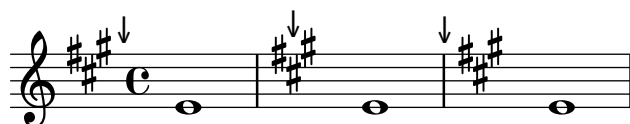
```
% The rehearsal mark will be aligned to the right edge of the Key Signature
\override Score.RehearsalMark.break-align-symbols = #'(key-signature clef)
\key a \major
\clef treble
\mark "↓"
e'1
```

```
% The rehearsal mark will be aligned to the right edge of the Clef
\set Staff.explicitKeySignatureVisibility = #all-invisible
\override Score.RehearsalMark.break-align-symbols = #'(key-signature clef)
\key a \major
\clef bass
\mark "↓"
gis,1
% The rehearsal mark will be centered above the Bar Line
\set Staff.explicitKeySignatureVisibility = #all-invisible
\set Staff.explicitClefVisibility = #all-invisible
\override Score.RehearsalMark.break-align-symbols = #'(key-signature clef)
\key a \major
\clef treble
\mark "↓"
e'1
```



La alineación de la letra de ensayo con relación al objeto de notación se puede cambiar, como se ve en el ejemplo siguiente. En una partitura con varios pentagramas, este ajuste se debe hacer para todos los pentagramas.

```
% The RehearsalMark will be aligned with the right edge of the Key Signature
\override Score.RehearsalMark.break-align-symbols = #'(key-signature)
\key a \major
\clef treble
\time 4/4
\mark "↓"
e'1
% The RehearsalMark will be centered above the Key Signature
\once \override Score.KeySignature.break-align-anchor-alignment = #CENTER
\mark "↓"
\key a \major
e'1
% The RehearsalMark will be aligned with the left edge of the Key Signature
\once \override Score.KeySignature.break-align-anchor-alignment = #LEFT
\key a \major
\mark "↓"
e'1
```



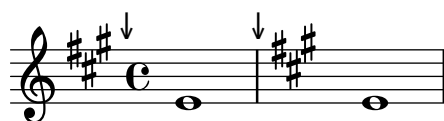
La letra de ensayo también se puede desplazar al borde derecho o izquierdo en una medida arbitraria. Las unidades son espacios de pentagrama:

```
% The RehearsalMark will be aligned with the left edge of the Key Signature
% and then shifted right by 3.5 staff-spaces
\override Score.RehearsalMark.break-align-symbols = #'(key-signature)
\once \override Score.KeySignature.break-align-anchor = #3.5
```

```

\key a \major
\mark "↓"
e'1
% The RehearsalMark will be aligned with the left edge of the Key Signature
% and then shifted left by 2 staff-spaces
\once \override Score.KeySignature.break-align-anchor = #-2
\key a \major
\mark "↓"
e'1

```



### 5.5.2 Agrupación vertical de objetos gráficos

Los grobs (objetos gráficos) `VerticalAlignment` y `VerticalAxisGroup` trabajan de manera coordinada. `VerticalAxisGroup` agrupa distintos grobs como `Staff`, `Lyrics`, etc. Después, `VerticalAlignment` alinea los distintos grobs agrupados previamente por `VerticalAxisGroup`. Normalmente sólo existe un `VerticalAlignment` por cada partitura, pero cada `Staff`, `Lyrics`, etc. tiene su propio `VerticalAxisGroup`.

### 5.5.3 Modificación de los sellos

Todos los objetos de presentación tienen una propiedad `stencil` que es parte del `grob-interface`. De forma predeterminada, esta propiedad suele estar establecida a una función específica del objeto que está hecha a medida para disponer el símbolo que lo representa en la salida. Por ejemplo, el ajuste estándar para la propiedad `stencil` del objeto `MultiMeasureRest` es `ly:multi-measure-rest::print`.

El símbolo estándar de cualquier objeto se puede sustituir modificando la propiedad `stencil` para que haga referencia a un procedimiento diferente escrito especialmente. Esto requiere un alto nivel de conocimiento del funcionamiento interno de LilyPond, pero hay una forma más fácil que a menudo puede dar resultados adecuados.

El procedimiento es establecer la propiedad `stencil` al procedimiento que imprime texto (`ly:text-interface::print`) y añadir una propiedad `text` al objeto ajustada para que contenga el texto de marcado que produce el símbolo requerido. Debido a la flexibilidad del marcado, se pueden conseguir muchas cosas; en particular, consulte [Notación gráfica dentro de elementos de marcado], página 274.

El ejemplo siguiente muestra esto cambiando el símbolo de la cabeza de la nota a unas aspas dentro de una circunferencia.

```

Xin0 = {
  \once \override NoteHead.stencil = #ly:text-interface::print
  \once \override NoteHead.text = \markup {
    \combine
      \halign #-0.7 \draw-circle #0.85 #0.2 ##f
      \musicglyph "noteheads.s2cross"
  }
}
\relative {
  a' a \Xin0 a a

```



}



Cualquiera de los glifos de *Feta* usados en la fuente tipográfica Emmentaler se puede pasar a la instrucción de marcado `\musicglyph`: véase Sección A.8 [La tipografía Emmentaler], página 713.

Se pueden insertar en línea tanto imágenes en formato EPS como instrucciones de Postscript usando las instrucciones de marcado `\epsfile` y `\postscript` respectivamente; véase Sección A.11.3 [Graphic], página 765.

## Véase también

Referencia de la notación: [Notación gráfica dentro de elementos de marcado], página 274, Sección 1.8.2 [Formatear el texto], página 264, `<undefined>` [`<undefined>`], página `<undefined>`, Sección A.8 [La tipografía Emmentaler], página 713, Sección A.11.3 [Graphic], página 765.

### 5.5.4 Modificación de las formas

#### Modificación de ligaduras de unión y de expresión

Las ligaduras de unión (`Ties`), de expresión (`Slurs`), de fraseo (`PhrasingSlurs`), de dejar vibrar (`LaissezVibrerTies`) y de repetición (`RepeatTies`) se trazan como curvas de Bézier de tercer orden. Si la forma de la ligadura calculada automáticamente no resulta óptima, se puede modificar su forma manualmente de dos formas:

- mediante la especificación de los desplazamientos que se quieren realizar sobre los puntos de control de la curva de Bézier calculada automáticamente, o bien
- mediante la especificación explícita de las posiciones de los cuatro puntos de control que se requieren para definir la curva deseada.

A continuación se explican ambos métodos. El primer método es más adecuado cuando solo se requieren ligeros ajustes de la curva; el segundo puede ser mejor para crear curvas que tienen relación con una única nota.

#### *Curvas de Bézier cúbicas*

Las curvas Bézier de tercer orden o cúbicas están definidas por cuatro puntos de control. El primer y cuarto puntos de control son exactamente los puntos extremos de comienzo y de final de la curva. Los dos puntos de control intermedios definen la forma. Se pueden encontrar en la web animaciones que muestran cómo se traza la curva, pero la descripción siguiente puede ser de ayuda. La curva comienza a partir del primer punto de control dirigiéndose directamente hacia el segundo, curvándose progresivamente para dirigirse hacia el tercero y continuando la curva hacia el cuarto, llegando a éste en viaje directo desde el tercer punto de control. La curva está contenida enteramente dentro del cuadrilátero definido por los cuatro puntos de control. Todas las traslaciones, rotaciones y escalado de los puntos de control producen exactamente las mismas operaciones sobre la curva.

#### *Especificar desplazamientos a partir de los puntos de control actuales*

En este ejemplo, la colocación automática de la ligadura no resulta óptima, y `\tieDown` no es la solución.

&lt;&lt;

{ e'1~ 1 }

```

\\
\relative { r4 <g' c,> <g c,> <g c,> }
>>

```



El ajuste de los puntos de control de la ligadura mediante la instrucción `\shape` permite evitar las colisiones.

La sintaxis de `\shape` es:

```
[-]\shape desplazamientos elemento
```

Esta instrucción recoloca los puntos de control de *elemento* en las cantidades dadas por *desplazamientos*. El argumento *desplazamientos* es una lista de parejas de números o una lista de tales listas. Cada elemento de una pareja representa el desplazamiento de una de las coordenadas de un punto de control. Si el *elemento* es una cadena de caracteres, el resultado es `\once\override` para el tipo de grob especificado. Si *elemento* es una expresión musical, el resultado es la misma expresión musical con la consiguiente modificación aplicada.

En otras palabras, la función `\shape` puede actuar como una instrucción `\once\override` o como una instrucción `\tweak` dependiendo de si el argumento *elemento* es el nombre de un grob, como “Slur”, o una expresión musical, como “(”. El argumento *desplazamientos* especifica los desplazamientos de los cuatro puntos de control como una lista de cuatro parejas de valores (dx . dy) en unidades de espacios de pentagrama (o una lista de tales listas si la curva tiene más de un segmento).

El guión precedente se exige si, y sólo si, se usa la forma `\tweak`.

Así, usando el mismo ejemplo de arriba y la forma `\once\override` de la instrucción `\shape`, lo siguiente tiene el efecto de elevar la ligadura de unión en la mitad de un espacio de pentagrama:

```

<<
{
  \shape #'((0 . 0.5) (0 . 0.5) (0 . 0.5) (0 . 0.5)) Tie
  e'1~ 1
}
\\
\relative { r4 <g' c,> <g c,> <g c,> }
>>

```



Este posicionamiento de la ligadura de unión es mejor, pero quizá debería elevarse más en la parte central. El ejemplo siguiente hace esto, esta vez usando la forma alternativa `\tweak`:

```

<<
{
  e'1-\shape #'((0 . 0.5) (0 . 1) (0 . 1) (0 . 0.5)) ~ e'
}
\\
\relative { r4 <g' c,> <g c,> <g c,> }
>>

```

&gt;&gt;



Los cambios en las posiciones horizontales de los puntos de control se pueden hacer de la misma forma, y dos curvas distintas que dan comienzo en el mismo momento musical se pueden conformar también:

```
\relative {
  c''8(\( a) a'4 e c\)
  \shape #'((0.7 . -0.4) (0.5 . -0.4) (0.3 . -0.3) (0 . -0.2)) Slur
  \shape #'((0 . 0) (0 . 0.5) (0 . 0.5) (0 . 0)) PhrasingSlur
  c8(\( a) a'4 e c\)
}
```



La función `\shape` también puede desplazar los puntos de control de las curvas que se extienden atravesando saltos de línea. Cada pieza de la curva dividida puede recibir su propia lista de desplazamientos. Si no se necesita hacer cambios a uno de los segmentos en particular, se puede usar una lista vacía como contenedor. En este ejemplo, el salto de línea hace que la única ligadura de expresión aparezca como si fueran dos:

```
\relative {
  c'4( f g c
  \break
  d,4 c' f, c)
}
```



Los cambios en la forma de las dos mitades de la ligadura de expresión deja más claro que la ligadura continúa más allá del salto de línea:

```
% ( ) may be used as a shorthand for ((0 . 0) (0 . 0) (0 . 0) (0 . 0))
% if any of the segments does not need to be changed
\relative c' {
  \shape #'(
    (( 0 . 0) (0 . 0) (0 . 0) (0 . 1))
    ((0.5 . 1.5) (1 . 0) (0 . 0) (0 . -1.5))
  ) Slur
  c4( f g c
  \break
  d,4 c' f, c)
```

}



En una curva en forma de «S» siempre es necesario ajustar manualmente los puntos de control: LilyPond nunca escoge tales formas automáticamente.

```
\relative c'' {
  c8( e b-> f d' a e-> g)
  \shape #'((0 . -1) (5.5 . -0.5) (-5.5 . -10.5) (0 . -5.5)) PhrasingSlur
  c8\ ( e b-> f d' a e-> g\ )
}
```

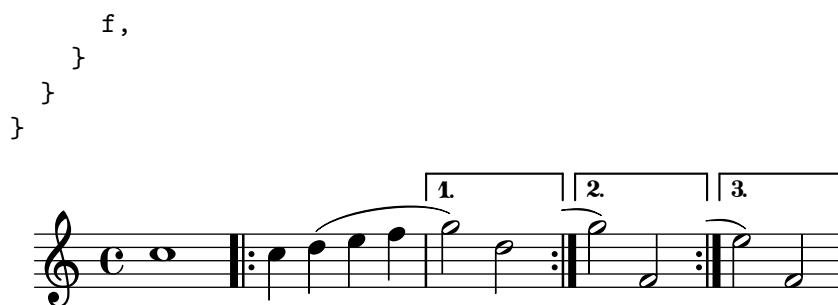


### *Especificar los puntos de control explícitamente*

Las coordenadas de los puntos de control de Bézier se especifican en unidades de espacios de pentagrama. La coordenada X es relativa al punto de referencia al que se adjunta la ligadura, y la coordenada Y es relativa a la línea central del pentagrama. Las coordenadas se especifican como una lista de cuatro parejas de números decimales (reales). Un enfoque consiste en estimar las coordenadas de los dos extremos, y después adivinar los dos puntos intermedios. Los valores óptimos se encuentran después por ensayo y error. Tenga en cuenta que estos valores podrían requerir un ajuste manual si se hace posteriormente cualquier cambio manual a la música o a la disposición.

Una situación en la que es preferible especificar los puntos de control explícitamente a especificar los desplazamientos es cuando se necesita especificarlos de forma relativa a una nota única. A continuación presentamos un ejemplo de esto. Muestra una forma de indicar una ligadura que se extiende hacia el interior de las casillas de primera y segunda vez de una repetición.

```
\relative {
  c''1
  \repeat volta 3 { c4 d( e f }
  \alternative {
    { g2) d }
    {
      g2
      % create a slur and move it to a new position
      % the <> is just an empty chord to carry the slur termination
      -\tweak control-points #'((-2 . 3.8) (-1 . 3.9) (0 . 4) (1 . 3.4)) ( <> )
      f,
    }
  }
  {
    e'2
    % create a slur and move it to a new position
    -\tweak control-points #'((-2 . 3) (-1 . 3.1) (0 . 3.2) (1 . 2.4)) ( <> )
  }
}
```



## Advertencias y problemas conocidos

No es posible modificar la forma de las ligaduras de unión o de expresión cambiando la propiedad `control-points` si hay más de una en el mismo momento musical, ni siquiera usando la instrucción `\tweak`. Sin embargo, se puede sobrescribir la propiedad `tie-configuration` de `TieColumn` para fijar la línea de inicio y la dirección según se requiera.

## Véase también

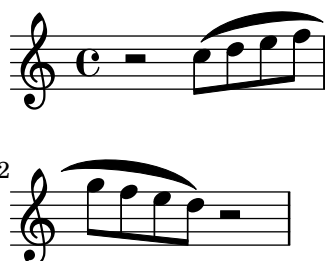
Referencia de funcionamiento interno: Sección “TieColumn” en *Referencia de Funcionamiento Interno*.

## 5.5.5 Modificación de objetos de extensión divididos

### Uso de `\alterBroken`

Cuando un “spanner” u objeto extenso atraviesa uno o varios saltos de línea, cada fragmento hereda los atributos del objeto de extensión original. Así, el trucado ordinario de un objeto dividido aplica las mismas modificaciones a todos y cada uno de sus segmentos. En el ejemplo que aparece a continuación, la sobrescritura de `thickness` (grosor) afecta a la ligadura a ambos lados del salto de línea.

```
\relative c'' {
  r2
  \once\override Slur.thickness = 10
  c8( d e f
  \break
  g8 f e d) r2
}
```



Es posible modificar de forma independiente el aspecto de los fragmentos individuales de un objeto de extensión dividido, con la instrucción `\alterBroken`. Esta instrucción puede producir un `\override` o bien un `\tweak` de una propiedad del objeto de extensión.

La sintaxis de `\alterBroken` es

```
[-]\alterBroken propiedad valores elemento
```

El argumento *valores* es una lista de valores, uno por cada fragmento. Si *elemento* es el nombre de un grob como `Slur` o `Staff.PianoPedalBracket`, el resultado es una instrucción de

sobreescritura `\override` del tipo de grob especificado. Si *elemento* es una expresión musical como “(” o “[” el resultado es la misma expresión musical con el correspondiente truco aplicado.

El guión inicial se debe utilizar con la forma de trucaje, `\tweak`. No lo escriba cuando utilice `\alterBroken` como una operación de sobreescritura, `\override`.

En su uso como `\override`, `\alterBroken` puede ir precedido de `\once` o `\temporary` y ser revertido con `\revert` con el argumento *propiedad* (véase Sección “Funciones de sustitución intermedias” en *Extender*).

EL código siguiente aplica un `\override` independiente a cada uno de los fragmentos de la ligadura del ejemplo anterior:

```
\relative c'' {
  r2
  \alterBroken thickness #'(10 1) Slur
  c8( d e f
  \break
  g8 f e d) r2
}
```



La instrucción `\alterBroken` se puede usar con cualquier objeto de extensión, incluidos Tie, PhrasingSlur, Beam y TextSpanner. Por ejemplo, un editor que está preparando una edición académica podría desear indicar la ausencia de parte de una ligadura de fraseo en una fuente haciendo discontinuo solamente el segmento que se ha añadido. El siguiente ejemplo ilustra de qué manera se puede hacer esto, en este caso usando la forma `\tweak` de la instrucción:

```
% The empty list is conveniently used below, because it is the
% default setting of dash-definition, resulting in a solid curve.
\relative {
  c''2-\alterBroken dash-definition #'(( (0 1.0 0.4 0.75))) \(\e
  \break
  g2 e\
}
```



Es importante comprender que `\alterBroken` fija cada pieza del objeto fragmentado al valor correspondiente dentro del argumento *valores*. Cuando hay menos valores que piezas, cualquier fragmento adicional recibe la asignación de la lista vacía. Esto puede conllevar resultados no deseados si la propiedad de disposición no está establecida a la lista vacía de forma predeterminada. En estos casos, se debe asignar a cada segmento un valor adecuado.

## Advertencias y problemas conocidos

Los saltos de línea pueden ocurrir en distintos lugares según cambia la disposición de la música. Los ajustes elegidos para `\alterBroken` podrían no ser adecuados para un objeto de extensión que ya no está dividido o que se fragmenta en más piezas que antes. El uso explícito (manual) de saltos de línea mediante `\break` puede prevenir situaciones como ésta.

La instrucción `\alterBroken` no es efectiva para propiedades de los objetos extensos accedidas antes del salto de línea, como `direction`.

## Véase también

Extender LilyPond: Sección “Trucos difíciles” en *Extender*.

### 5.5.6 Contenedores unpure-pure

Los contenedores ‘unpure-pure’ (pura y no pura) son útiles para la sobrescritura de los cálculos del espaciado en el *eje Y* (concretamente `Y-offset` e `Y-extent`) con una función de Scheme en lugar de un literal, es decir, un número o una pareja de números.

Para ciertos objetos gráficos, las dimensiones `Y-extent` están basadas en la propiedad `stencil`, la sobrescritura de la propiedad de sello de éstos requiere una sobrescritura adicional de `Y-extent` con un contenedor unpure-pure. Cuando una función sobrescribe una dimensión `Y-offset` y/o `Y-extent` se supone que esto dispara los cálculos de los saltos de línea prematuramente durante la compilación. Así pues, la función no se evalúa en absoluto (devolviendo por lo general un valor de ‘0’ o ‘(0 . 0)’) lo que puede dar lugar a colisiones. Una función ‘pura’ no afecta a las propiedades, objetos o suicidios de grobs, y por ello siempre ven sus valores relacionados con el eje Y evaluados correctamente.

Actualmente hay unas treinta funciones que ya se consideran ‘puras’ y los contenedores Unpure-pure son una manera de establecer funciones que no están en esta lista como ‘puras’. La función ‘pura’ se evalúa *antes* de cualquier salto de línea y así el espaciado horizontal se puede ajustar ‘a tiempo’. La función ‘impura’ se evalúa entonces *después* del salto de línea.

**Nota:** Dado que es difícil saber siempre qué funciones están en esta lista, recomendamos que cualquier función ‘pura’ que estemos creando no utilice los grobs `Beam` o `VerticalAlignment`.

Un contenedor ‘unpure-pure’ se contruye de la manera siguiente:

```
(ly:make-unpure-pure-container f0 f1)
```

donde `f0` es una función que toma  $n$  argumentos ( $n \geq 1$ ) y el primer argumento siempre debe ser el grob. Ésta es la función que da el resultado real. `f1` es la función que se etiqueta como ‘pura’ que toma  $n + 2$  argumentos. De nuevo, el primer argumento debe ser siempre el grob pero los argumentos segundo y tercero son argumentos de ‘inicio’ y de ‘final’.

*inicio* y *final* son, a todos los efectos, valores mudos que sólo tienen importancia para los **objetos de extensión** (o sea: `Hairpin`, regulador, o `Beam`, barra), que pueden devolver distintas estimaciones de altura basadas en una columna de inicio y una de final.

El resto son los otros argumentos para la primera función (que puede no ser ninguno si  $n = 1$ ).

El resultado de la segunda función se usa como una aproximación del valor necesario, que se usa entonces por la primera función para obtener el valor real que se usa a continuación para el ajuste de precisión mucho más tardío durante el proceso de espaciado.

```
#(define (square-line-circle-space grob)
  (let* ((pitch (ly:event-property (ly:grob-property grob 'cause) 'pitch))
        (notename (ly:pitch-notename pitch)))
```

```

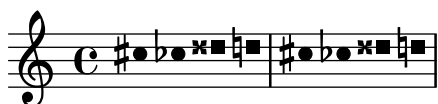
(if (= 0 (modulo notename 2))
  (make-circle-stencil 0.5 0.0 #t)
  (make-filled-box-stencil '(0 . 1.0)
    '(-0.5 . 0.5))))))

squareLineCircleSpace = {
  \override NoteHead.stencil = #square-line-circle-space
}

smartSquareLineCircleSpace = {
  \squareLineCircleSpace
  \override NoteHead.Y-extent =
    #(ly:make-unpure-pure-container
      ly:grob::stencil-height
      (lambda (grob start end) (ly:grob::stencil-height grob)))
}

\new Voice \with { \remove "Stem_engraver" }
\relative c'' {
  \squareLineCircleSpace
  cis4 ces disis d
  \smartSquareLineCircleSpace
  cis4 ces disis d
}

```



En el primer compás, sin el contenedor *unpure-pure*, el motor de espaciado no conoce la anchura de la cabeza de la nota y permite que colisione con las alteraciones accidentales. En el segundo compás, el motor de espaciado conoce la anchura de las cabezas de las notas y evita la colisión mediante el alargamiento de la línea en la medida adecuada.

Normalmente, para cálculos simples se pueden usar funciones casi idénticas tanto para las partes ‘no pura’ y ‘pura’, simplemente cambiando el número de argumentos que se pasan a, y el ámbito de, la función. Este caso de utilización es lo bastante frecuente como para que `ly:make-unpure-pure-container` construya dicha segunda función de forma predeterminada cuando se llama con solo un argumento de función.

**Nota:** Si una función está caracterizada como ‘pura’ y resulta que no lo es, el resultado puede ser inesperado.

## 5.6 Uso de las funciones musicales

Dondequiera que se necesite reutilizar trucos con distintas expresiones musicales, con frecuencia conviene hacer que el truco forme parte de una *función musical*. En esta sección estudiaremos solamente las funciones de *sustitución*, en las que el objeto es sustituir una variable dentro de un fragmento de código de entrada de LilyPond. Se describen otras funciones más complejas en Sección “Funciones musicales” en *Extender*.



### 5.6.1 Sintaxis de las funciones de sustitución

Es fácil hacer una función que sustituya una variable dentro de código de LilyPond. La forma general de estas funciones es

```
funcion =
#(define-music-function
  (arg1 arg2 ...)
  (type1? type2? ...)
  #{
    ...música...
  #})
```

donde

*argN* *n*-ésimo argumento

*typeN?* *predicado de tipo* de Scheme para el que *argN* debe devolver *#t*.

*...música...* código de entrada normal de LilyPond, que utiliza *\$* (en los lugares en que sólo se permiten construcciones de LilyPond) o *#* (para usarlo como un valor de Scheme o un argumento de función musical o música dentro de listas de música) para referenciar argumentos (p.ej. *#arg1*).

Es necesaria la lista de predicados de tipo. Algunos de los predicados de tipo más comunes que se utilizan en las funciones musicales son:

```
boolean?
cheap-list?  (utilizar en lugar de 'list?' para un procesamiento más rápido)
ly:duration?
ly:music?
ly:pitch?
markup?
number?
pair?
string?
symbol?
```

Para ver una lista de los predicados de tipo disponibles, consulte Sección A.21 [Predicados de tipo predefinidos], página 856. También se permiten los predicados de tipo definidos por el usuario.

### Véase también

Referencia de la notación: Sección A.21 [Predicados de tipo predefinidos], página 856.

Expansión de LilyPond: Sección “Funciones musicales” en *Extender*.

Archivos instalados: `lily/music-scheme.cc`, `scm/c++.scm`, `scm/lily.scm`.

### 5.6.2 Ejemplos de funciones de sustitución

Esta sección presenta algunos ejemplos de funciones de sustitución. No pretenden ser exhaustivas, sino demostrar algunas de las posibilidades de las funciones de sustitución sencillas.

En el primer ejemplo se define una función que simplifica el ajuste del relleno de un elemento de inscripción de texto TextScript:

```
padText =
#(define-music-function
  (padding)
```

```

      (number?)
    #{
      \once \override TextScript.padding = #padding
    #})

\relative {
  c''4^"piu mosso" b a b
  \padText #1.8
  c4^"piu mosso" b a b
  \padText #2.6
  c4^"piu mosso" b a b
}

```



Además de números, podemos usar expresiones musicales, como notas, para los argumentos de las funciones musicales:

```

custosNote =
#(define-music-function
  (note)
  (ly:music?)
  #{
    \tweak NoteHead.stencil #ly:text-interface::print
    \tweak NoteHead.text
      \markup \musicglyph "custodes.mensural.u0"
    \tweak Stem.stencil ##f
    #note
  #})

\relative { c'4 d e f \custosNote g }

```



Ambas funciones son expresiones únicas sencillas en las que solo falta el último elemento de una llamada de función o sobreescritura. Para estas definiciones de función en particular, existe una sintaxis alternativa más simple que consiste simplemente en escribir la parte constante de la expresión y sustituir el elemento final que falta por `\etc`:

```

padText =
  \once \override TextScript.padding = \etc

\relative {
  c''4^"piu mosso" b a b
  \padText #1.8
  c4^"piu mosso" b a b
  \padText #2.6
  c4^"piu mosso" b a b
}

```

```
}
```



```
custosNote =
  \tweak NoteHead.stencil #ly:text-interface::print
  \tweak NoteHead.text
    \markup \musicglyph "custodes.mensural.u0"
  \tweak Stem.stencil ##f
  \etc
```

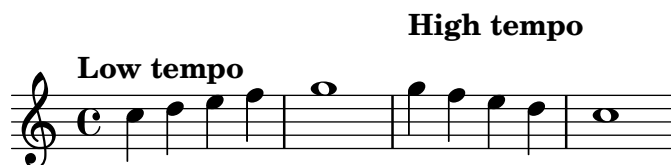
```
\relative { c'4 d e f \custosNote g }
```



Se pueden definir funciones de sustitución con más de un argumento:

```
tempoPadded =
#(define-music-function
  (padding tempotext)
  (number? markup?)
  #{
    \once \override Score.MetronomeMark.padding = #padding
    \tempo \markup { \bold #tempotext }
  #})

\relative {
  \tempo \markup { "Low tempo" }
  c''4 d e f g1
  \tempoPadded #4.0 "High tempo"
  g4 f e d c1
}
```



## Apéndice A Tablas del manual sobre notación

### A.1 Cuadro de nombres de acordes



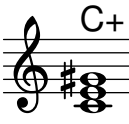


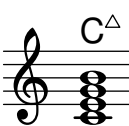

La siguiente tabla muestra el sistema estándar para imprimir los nombres de los acordes, junto a las notas que representa. Se muestran también otros sistemas de nomenclatura (no soportados) en el fragmento de código “Chord names alternative” de la sección Sección “Chords” en *Fragmentos de código*, incluida la notación inspirada por Harald Banter (1982) que se utilizaba de forma predeterminada en algunas de las primeras versiones de LilyPond (hasta la 1.7).

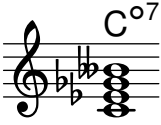


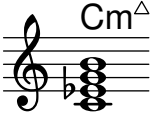
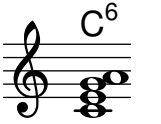

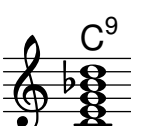

The following table lists the chord names shown in the musical score, ordered from top-left to bottom-right:


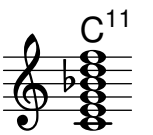
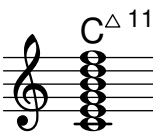
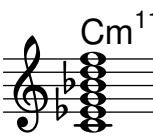

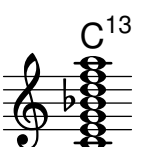
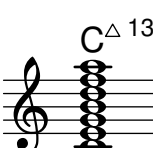

C	Cm	C+	C°	C <sup>7</sup>	Cm <sup>7</sup>	C <sup>Δ</sup>	C <sup>°7</sup>	Cm <sup>Δ</sup> b5
C <sup>7</sup> #5	Cm <sup>Δ</sup>	C <sup>Δ</sup> #5	C <sup>°</sup>	C <sup>6</sup>	Cm <sup>6</sup>	C <sup>9</sup>	Cm <sup>9</sup>	
Cm <sup>13</sup>	Cm <sup>11</sup>	Cm <sup>7</sup> b5 9	C <sup>7</sup> b9	C <sup>7</sup> #9	C <sup>11</sup>	C <sup>7</sup> #11	C <sup>13</sup>	
C <sup>7</sup> #11 b13	C <sup>7</sup> #5 #9	C <sup>7</sup> #9 #11	C <sup>7</sup> b13					
C <sup>7</sup> b9 b13	C <sup>7</sup> #11	C <sup>Δ</sup> 9	C <sup>7</sup> b13					
C <sup>7</sup> b9 b13	C <sup>7</sup> b9 13	C <sup>Δ</sup> 9	C <sup>Δ</sup> 13	C <sup>Δ</sup> #11	C <sup>7</sup> b9 13			
C <sup>sus4</sup>	C <sup>7</sup> sus4	C <sup>9</sup> sus4	C <sup>9</sup>	Cm <sup>11</sup>	C <sup>lyd</sup>	C <sup>alt</sup>		

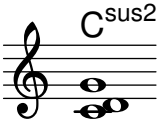
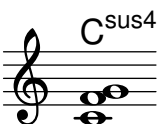


## A.2 Modificadores de acorde más usuales

La tabla siguiente presenta los modificadores de acorde que se pueden usar para generar las estructuras de acordes estándar.

Tipo	Intervalo	Modificador	Ejemplo	Salida
Mayor	Tercera quinta justa	mayor y 5 o nada	c1:5	
Menor	Tercera quinta justa	menor y m o m5	c1:m	
Aumentado	Tercera quinta aumentada	mayor y aug	c1:aug	
Disminuido	Tercera quinta disminuida	menor y dim	c1:dim	
Séptima dominante	Tríada séptima menor	mayor con 7	c1:7	
Séptima mayor	Tríada séptima mayor	mayor y maj7 ó maj	c1:maj7	
Séptima menor	Tríada séptima menor	menor y m7	c1:m7	

Séptima disminuida		Tríada disminuida y séptima disminuida	dim7	c1:dim7	
Aumentado séptima	con	Tríada aumentada y séptima menor	aug7	c1:aug7	
Séptima semi-disminuido		Tríada disminuida y séptima menor	m7.5-	c1:m7.5-	
Menor con séptima mayor		Tríada menor con séptima mayor	m7+	c1:m7+	
Mayor con sexta		Tríada mayor y sexta	6	c1:6	
Menor con sexta		Tríada menor y sexta	m6	c1:m6	
Novena dominante		Séptima dominante con novena mayor	9	c1:9	
Mayor con novena		Séptima mayor y no- vena mayor	maj9	c1:maj9	

Menor con novena	Séptima menor y novena mayor	m9	c1:m9	
Oncena dominante	Novena dominante con oncena justa	11	c1:11	
Oncena mayor	Novena mayor y oncena justa	maj11	c1:maj11	
Menor con oncena	Novena menor y oncena justa	m11	c1:m11	
Trecena dominante	Novena dominante y trecena mayor	13	c1:13	
Trecena dominante	Oncena dominante y trecena mayor	13.11	c1:13.11	
Mayor con trecena	Oncena mayor y trecena mayor	maj13.11	c1:maj13.11	
Menor con trecena	Oncena menor y trecena mayor	m13.11	c1:m13.11	

Segunda suspendida	Segunda mayor y quinta justa	sus2	c1:sus2	
Cuarta suspendida	Cuarta justa y quinta justa	sus4	c1:sus4	
Acorde de potencia o de quinta vacía (de dos notas)	Quinta justa	1.5	c1:5	
Acorde de potencia o de quinta vacía (de tres notas)	Quinta justa, octava	1.5.8	c1:5.8	

### A.3 Afinaciones de cuerdas predefinidas

El cuadro de abajo muestra las afinaciones predefinidas de cuerdas.

**Guitar tunings**

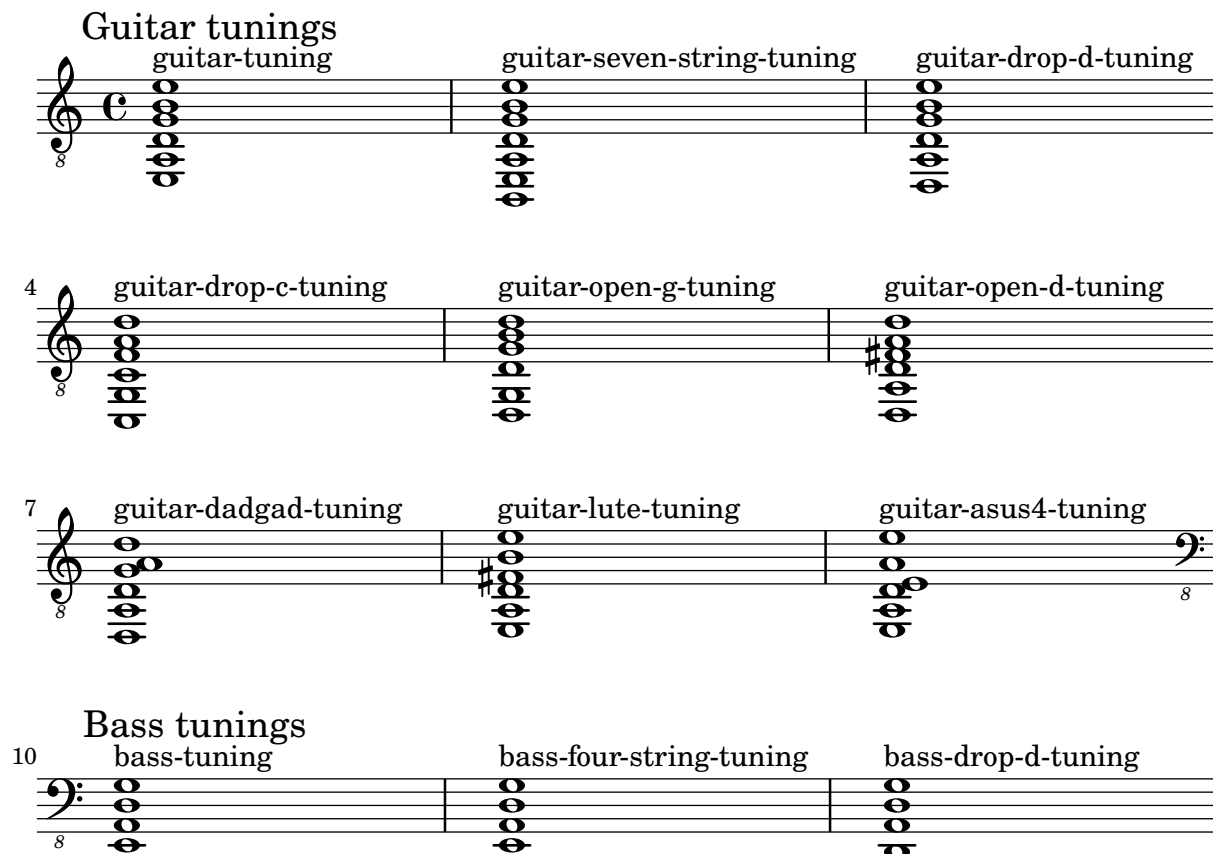
guitar-tuning      guitar-seven-string-tuning      guitar-drop-d-tuning

guitar-drop-c-tuning      guitar-open-g-tuning      guitar-open-d-tuning

guitar-dadgad-tuning      guitar-lute-tuning      guitar-asus4-tuning

**Bass tunings**

bass-tuning      bass-four-string-tuning      bass-drop-d-tuning





13      bass-five-string-tuning      bass-six-string-tuning

Mandolin tunings

15      mandolin-tuning

Banjo tunings

16      banjo-open-g-tuning      banjo-c-tuning

18      banjo-modal-tuning      banjo-open-d-tuning

20      banjo-open-dm-tuning      banjo-double-c-tuning      banjo-double-d-tuning

Ukulele tunings

23      ukulele-tuning      ukulele-d-tuning

25      tenor-ukulele-tuning      baritone-ukulele-tuning

Orchestral string tunings

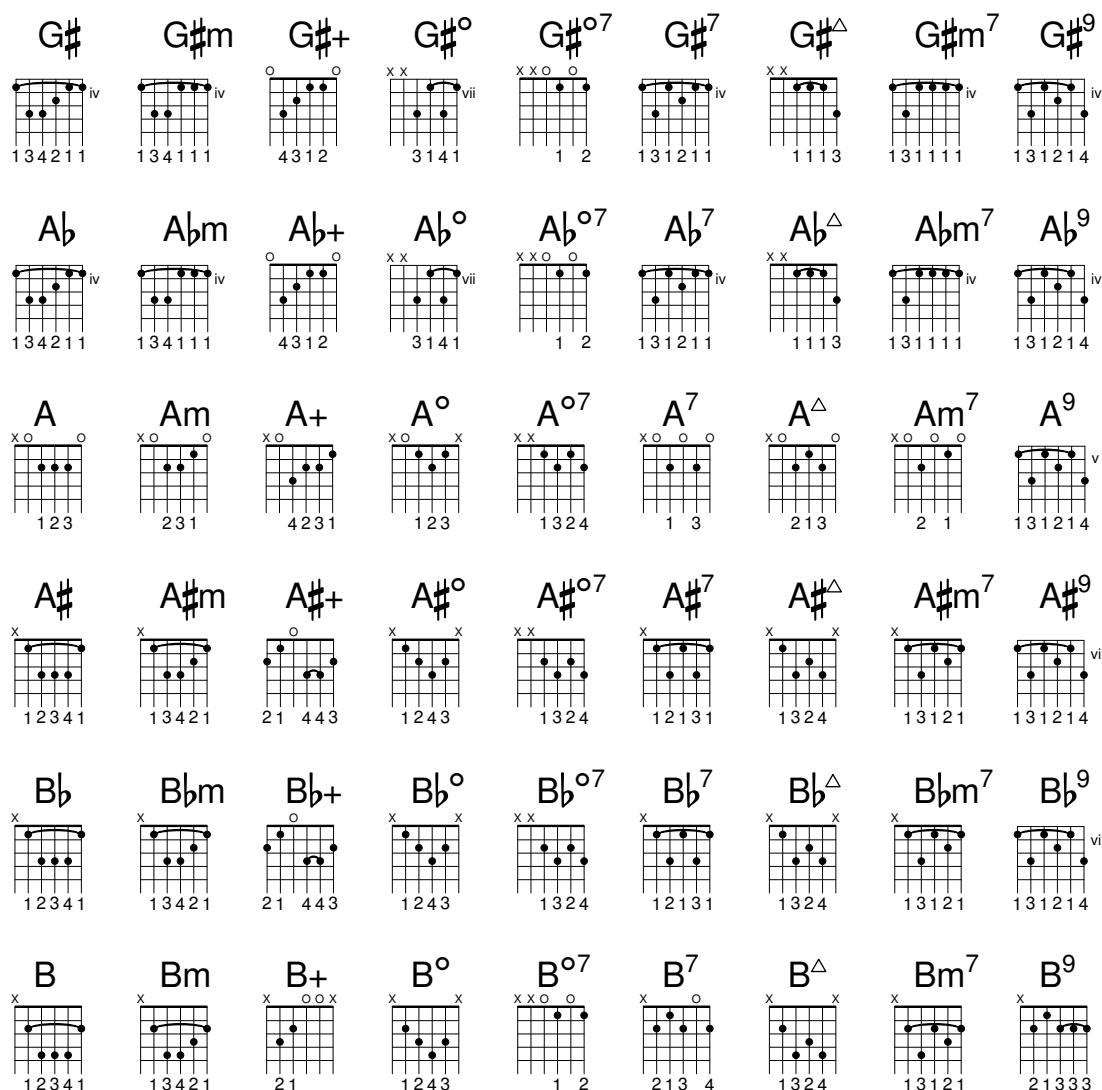
27      violin-tuning      viola-tuning      cello-tuning      double-bass-tuning

## A.4 Diagramas predefinidos de posiciones

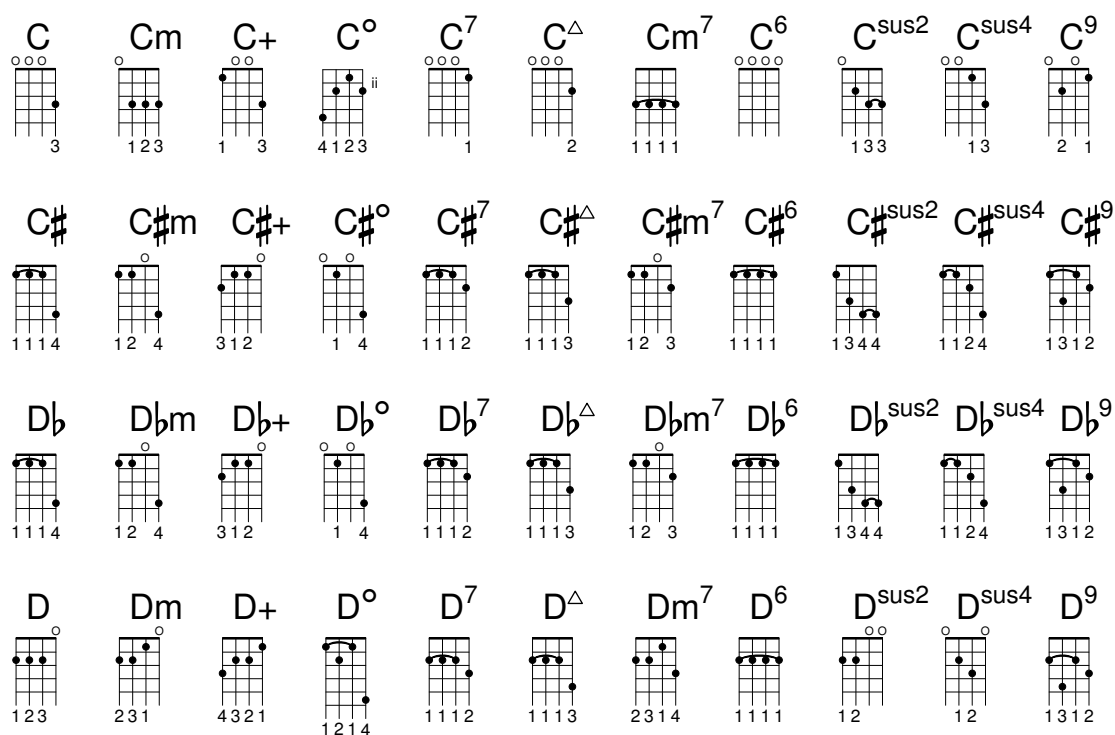
### Diagramas para guitarra

C      Cm      C+      C°      C°7      C7      C△      Cm7      C9

3 1 2 1	2 1 3	4 3 1 2	3 4	1 3 2 4	2 3 1 4	4 3 1 1 1	4 2 1	2 1 3 3 3
3 1 2 1	2 1 3	4 3 1 2	3 4	1 3 2 4	2 3 1 4	4 3 1 1 1	4 2 1	2 1 3 3 3
1 3 2	2 3 1	2 3 1	1 3 1	1 2	2 1 3	1 2 3	2 1 1	2 1 3 3 3
3 1 2 1	3 2 4 1	3 2 1 4	3 1 4 1	1 3 2 4	1 3 2 4	1 2 3 4	1 2 3 4	2 1 3 3 3
3 1 2 1	3 2 4 1	3 2 1 4	3 1 4 1	1 3 2 4	1 3 2 4	1 2 3 4	1 2 3 4	2 1 3 3 3
2 3 1	2 3	3 2 1	3 1 4 1	1 3 2 4	2 1	3 1 2	2	2 1 3
1 3 4 2 1 1	1 3 4 1 1 1	1 3 4 2	3 1 4 1	1 2	1 3 1 2 1 1	3 2 1	1 3 1 1 1 1	1 3 1 2 1 4
1 3 4 2 1 1	1 3 4 1 1 1	2 1 4 4 3	3 1 4 1	1 3 2 4	1 3 1 2 1 1	4 3 2 1	1 3 1 1 1 1	1 3 1 2 1 4
1 3 4 2 1 1	1 3 4 1 1 1	2 1 4 4 3	3 1 4 1	1 3 2 4	1 3 1 2 1 1	4 3 2 1	1 3 1 1 1 1	1 3 1 2 1 4
2 1 3	1 3 4 1 1 1	1 3 4 2	3 1 4 1	1 3 2 4	3 2 1	4 3 2 1	1 3 1 1 1 1	1 3 1 2 1 4

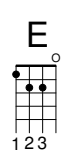
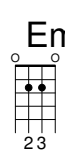
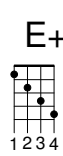
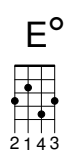
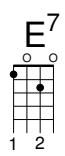
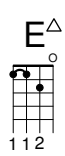
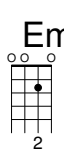
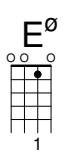
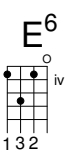
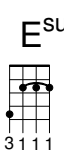
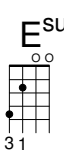
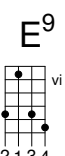
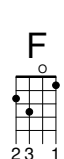
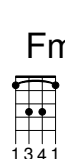
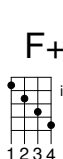
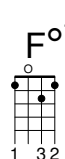
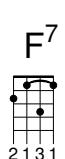
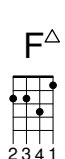
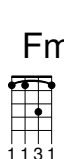
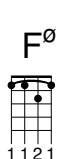
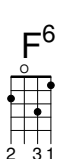
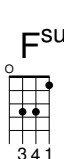
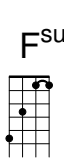
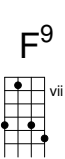
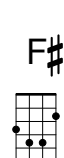


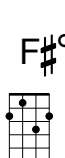
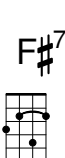
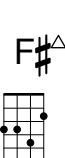

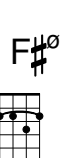

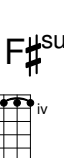

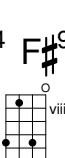
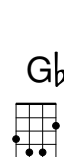

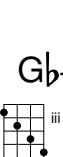
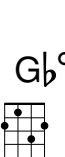
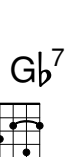
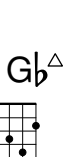

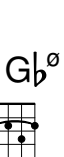
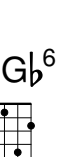

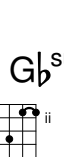
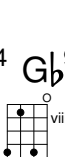


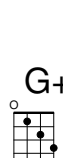
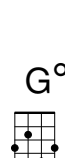
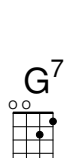
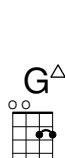

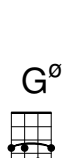

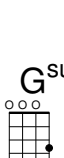
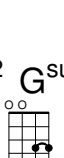
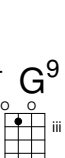
















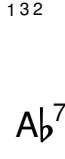




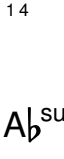
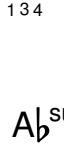
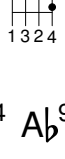











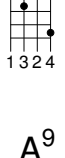
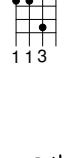
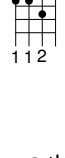

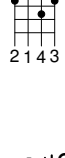
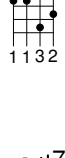
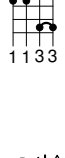

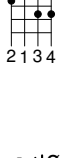
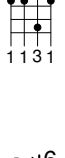
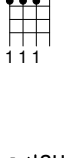

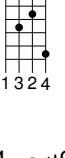


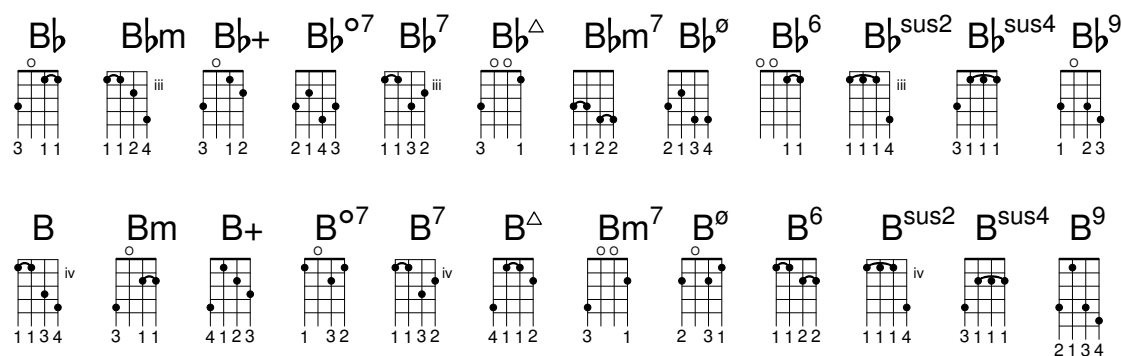
## Diagramas para ukelele



<b>D<sup>#</sup></b>  2 3 1	<b>D<sup>#</sup>m</b>  3 4 2 1	<b>D<sup>#</sup>+</b>  2 3 1	<b>D<sup>#</sup>°</b>  1 3 3	<b>D<sup>#</sup>7</b>  1 1 1 2	<b>D<sup>#</sup>Δ</b>  1 1 1 2	<b>D<sup>#</sup>m<sup>7</sup></b>  2 3 1 4	<b>D<sup>#</sup>6</b>  1 1 1 1	<b>D<sup>#</sup>sus2</b>  2 3 1 1	<b>D<sup>#</sup>sus4</b>  1 3 4 1	<b>D<sup>#</sup>9</b>  1 1 1 1
<b>E<sup>b</sup></b>  2 3 1	<b>E<sup>b</sup>m</b>  3 4 2 1	<b>E<sup>b</sup>+</b>  2 3 1	<b>E<sup>b</sup>°</b>  1 3 3	<b>E<sup>b</sup>7</b>  1 1 1 2	<b>E<sup>b</sup>Δ</b>  1 1 1 2	<b>E<sup>b</sup>m<sup>7</sup></b>  2 3 1 4	<b>E<sup>b</sup>6</b>  1 1 1 1	<b>E<sup>b</sup>sus2</b>  2 3 1 1	<b>E<sup>b</sup>sus4</b>  1 3 4 1	<b>E<sup>b</sup>9</b>  1 1 1 1
<b>E</b>  1 4 2	<b>Em</b>  3 1	<b>E+</b>  1 3	<b>E°</b>  4 1	<b>E7</b>  1 2 3	<b>EΔ</b>  1 3 2	<b>Em<sup>7</sup></b>  1 2	<b>E6</b>  1 2 3	<b>E<sup>sus</sup>2</b>  2 3 1 1	<b>E<sup>sus</sup>4</b>  2 4 1	<b>E9</b>  1 2 3 4
<b>F</b>  2 1	<b>Fm</b>  1 2 4	<b>F+</b>  3 1 2	<b>F°</b>  1 4 1 2	<b>F7</b>  2 3 1 4	<b>FΔ</b>  2 4 1 3	<b>Fm<sup>7</sup></b>  1 3 2 4	<b>F6</b>  2 3 1 4	<b>F<sup>sus</sup>2</b>  1 3	<b>F<sup>sus</sup>4</b>  3 1 1	<b>F9</b>  1 2 3 4
<b>F<sup>#</sup></b>  3 1 2 1	<b>F<sup>#</sup>m</b>  2 1 3	<b>F<sup>#</sup>+</b>  4 3 2 1	<b>F<sup>#</sup>°</b>  1 2	<b>F<sup>#</sup>7</b>  2 3 1 4	<b>F<sup>#</sup>Δ</b>  2 4 1 3	<b>F<sup>#</sup>m<sup>7</sup></b>  1 3 2 4	<b>F<sup>#</sup>6</b>  2 3 1 4	<b>F<sup>#</sup>sus2</b>  1 1 2 4	<b>F<sup>#</sup>sus4</b>  4 1 2 3	<b>F<sup>#</sup>9</b>  1 2 3 3
<b>G<sup>b</sup></b>  3 1 2 1	<b>G<sup>b</sup>m</b>  2 1 3	<b>G<sup>b</sup>+</b>  4 3 2 1	<b>G<sup>b</sup>°</b>  1 2	<b>G<sup>b</sup>7</b>  2 3 1 4	<b>G<sup>b</sup>Δ</b>  2 4 1 3	<b>G<sup>b</sup>m<sup>7</sup></b>  1 3 2 4	<b>G<sup>b</sup>6</b>  2 3 1 4	<b>G<sup>b</sup>sus2</b>  1 1 2 4	<b>G<sup>b</sup>sus4</b>  4 1 2 3	<b>G<sup>b</sup>9</b>  1 2 3 3
<b>G</b>  1 3 2	<b>Gm</b>  2 3 1	<b>G+</b>  2 3 1	<b>G°</b>  1 4 2	<b>G7</b>  2 1 3	<b>GΔ</b>  1 2 3	<b>Gm<sup>7</sup></b>  2 1 1	<b>G6</b>  1 2	<b>G<sup>sus</sup>2</b>  1 2	<b>G<sup>sus</sup>4</b>  1 2 3	<b>G9</b>  2 3 1 4
<b>G<sup>#</sup></b>  4 1 3 2	<b>G<sup>#</sup>m</b>  1 3 4 2	<b>G<sup>#</sup>+</b>  1 3	<b>G<sup>#</sup>°</b>  1 2 4 3	<b>G<sup>#</sup>7</b>  1 3 2 4	<b>G<sup>#</sup>Δ</b>  1 2 3 4	<b>G<sup>#</sup>m<sup>7</sup></b>  1 4 2 3	<b>G<sup>#</sup>6</b>  1 3 2 4	<b>G<sup>#</sup>sus2</b>  1 3 4 1	<b>G<sup>#</sup>sus4</b>  1 3 4 4	<b>G<sup>#</sup>9</b>  2 3 1 4
<b>A<sup>b</sup></b>  4 1 3 2	<b>A<sup>b</sup>m</b>  1 3 4 2	<b>A<sup>b</sup>+</b>  1 3	<b>A<sup>b</sup>°</b>  1 2 4 3	<b>A<sup>b</sup>7</b>  1 3 2 4	<b>A<sup>b</sup>Δ</b>  1 2 3 4	<b>A<sup>b</sup>m<sup>7</sup></b>  1 4 2 3	<b>A<sup>b</sup>6</b>  1 3 2 4	<b>A<sup>b</sup>sus2</b>  1 3 4 1	<b>A<sup>b</sup>sus4</b>  1 3 4 4	<b>A<sup>b</sup>9</b>  2 3 1 4
<b>A</b>  2 1	<b>Am</b>  2	<b>A+</b>  3 1 2	<b>A°</b>  1 2 4 4	<b>A7</b>  1	<b>AΔ</b>  1 2	<b>Am<sup>7</sup></b>  1 3 2 4	<b>A6</b>  1 3 2 4	<b>A<sup>sus</sup>2</b>  1 3 2	<b>A<sup>sus</sup>4</b>  1 2	<b>A9</b>  1 2



 E 1 2 3	 Em 2 3	 E+	 E <sup>°7</sup> 2 1 4 3	 E <sup>7</sup> 1 2	 E <sup>Δ</sup> 1 1 2	 Em <sup>7</sup> 2	 E <sup>∅</sup> 1	 E <sup>6</sup> 1 3 2	 E <sup>sus2</sup> 3 1 1 1	 E <sup>sus4</sup> 3 1	 E <sup>9</sup> 2 1 3 4
 F 2 3 1	 Fm 1 3 4 1	 F+	 F <sup>°7</sup> 1 3 2	 F <sup>7</sup> 2 1 3 1	 F <sup>Δ</sup> 2 3 4 1	 Fm <sup>7</sup> 1 1 3 1	 F <sup>∅</sup> 1 1 2 1	 F <sup>6</sup> 2 3 1	 F <sup>sus2</sup> 3 4 1	 F <sup>sus4</sup> 4 2 1 1	 F <sup>9</sup> 2 1 3 4
 F <sup>#</sup> 2 3 4 1	 F <sup>#m</sup> 1 3 4 1	 F <sup>#+</sup> 1 2 3 4	 F <sup>#°7</sup> 2 1 4 3	 F <sup>#7</sup> 2 1 3 1	 F <sup>#Δ</sup> 2 3 4 1	 F <sup>#m7</sup> 1 1 3 1	 F <sup>#∅</sup> 1 1 2 1	 F <sup>#6</sup> 3 1 4 2	 F <sup>#sus2</sup> 3 1 1 1	 F <sup>#sus4</sup> 4 2 1 1	 F <sup>#9</sup> 2 1 3
 G <sup>b</sup> 2 3 4 1	 G <sup>b</sup> m 1 3 4 1	 G <sup>b+</sup> 1 2 3 4	 G <sup>b°7</sup> 2 1 4 3	 G <sup>b7</sup> 2 1 3 1	 G <sup>bΔ</sup> 2 3 4 1	 G <sup>b</sup> m <sup>7</sup> 1 1 3 1	 G <sup>b∅</sup> 1 1 2 1	 G <sup>b6</sup> 3 1 4 2	 G <sup>b</sup> sus2 3 1 1 1	 G <sup>b</sup> sus4 4 2 1 1	 G <sup>b9</sup> 2 1 3
 G 1 2	 Gm 1 3	 G+	 G <sup>°7</sup> 2 1 4 3	 G <sup>7</sup> 2 1	 G <sup>Δ</sup> 1 1	 Gm <sup>7</sup> 1 1	 G <sup>∅</sup> 1 1 2 1	 G <sup>6</sup> 2	 G <sup>sus2</sup> 3	 G <sup>sus4</sup> 1 1	 G <sup>9</sup> 1 4
 G <sup>#</sup> 1 1 3 4	 G <sup>#m</sup> 1 1 2 4	 G <sup>#+</sup> 1 2 3 4	 G <sup>#°7</sup> 1 3 2	 G <sup>#7</sup> 1 1 3 2	 G <sup>#Δ</sup> 1 1 3 3	 G <sup>#m7</sup> 1 1 2 2	 G <sup>#∅</sup> 1 2 2	 G <sup>#6</sup> 1 1 3 1	 G <sup>#sus2</sup> 1 1 1 4	 G <sup>#sus4</sup> 1 1 3 4	 G <sup>#9</sup> 1 3 2 4
 A <sup>b</sup> 1 1 3 4	 A <sup>b</sup> m 1 1 2 4	 A <sup>b+</sup> 1 2 3 4	 A <sup>b°7</sup> 1 3 2	 A <sup>b7</sup> 1 1 3 2	 A <sup>bΔ</sup> 1 1 3 3	 A <sup>b</sup> m <sup>7</sup> 1 1 2 2	 A <sup>b∅</sup> 1 2 2	 A <sup>b6</sup> 1 1 3 1	 A <sup>b</sup> sus2 1 1 1 4	 A <sup>b</sup> sus4 1 1 3 4	 A <sup>b9</sup> 1 3 2 4
 A 1 1 3	 Am 1 1 2	 A+	 A <sup>°7</sup> 2 1 4 3	 A <sup>7</sup> 1 1 3 2	 A <sup>Δ</sup> 1 1 3 3	 Am <sup>7</sup> 1 1 2 2	 A <sup>∅</sup> 2 1 3 4	 A <sup>6</sup> 1 1 3 1	 A <sup>sus2</sup> 1 1 1	 A <sup>sus4</sup> 1	 A <sup>9</sup> 1 3 2 4
 A <sup>#</sup> 3 1 1	 A <sup>#m</sup> 1 1 2 4	 A <sup>#+</sup> 3 1 2	 A <sup>#°7</sup> 2 1 4 3	 A <sup>#7</sup> 1 1 3 2	 A <sup>#Δ</sup> 3 1 1	 A <sup>#m7</sup> 1 1 2 2	 A <sup>#∅</sup> 2 1 3 4	 A <sup>#6</sup> 1 1	 A <sup>#sus2</sup> 1 1 1 4	 A <sup>#sus4</sup> 3 1 1 1	 A <sup>#9</sup> 1 2 3



## A.5 Tamaños de página predefinidos

Los tamaños de página se definen en el archivo `scm/paper.scm`

### La serie A de “ISO 216”

"a10"	(26 x 37 mm)
"a9"	(37 x 52 mm)
"a8"	(52 x 74 mm)
"a7"	(74 x 105 mm)
"a6"	(105 x 148 mm)
"a5"	(148 x 210 mm)
"a4"	(210 x 297 mm)
"a3"	(297 x 420 mm)
"a2"	(420 x 594 mm)
"a1"	(594 x 841 mm)
"a0"	(841 x 1189 mm)

### La serie B de “ISO 216”

"b10"	(31 x 44 mm)
"b9"	(44 x 62 mm)
"b8"	(62 x 88 mm)
"b7"	(88 x 125 mm)
"b6"	(125 x 176 mm)
"b5"	(176 x 250 mm)
"b4"	(250 x 353 mm)
"b3"	(353 x 500 mm)
"b2"	(500 x 707 mm)
"b1"	(707 x 1000 mm)
"b0"	(1000 x 1414 mm)

### Dos tamaños extendidos según se definen en “DIN 476”

"4a0"	(1682 x 2378 mm)
"2a0"	(1189 x 1682 mm)

### Serie C estándar de “ISO 269”

"c10"	(28 x 40 mm)
-------	--------------

"c9"	(40 x 57 mm)
"c8"	(57 x 81 mm)
"c7"	(81 x 114 mm)
"c6"	(114 x 162 mm)
"c5"	(162 x 229 mm)
"c4"	(229 x 324 mm)
"c3"	(324 x 458 mm)
"c2"	(458 x 648 mm)
"c1"	(648 x 917 mm)
"c0"	(917 x 1297 mm)

#### **Tamaños de página norteamericanos**

"junior-legal"	(8.0 x 5.0 in)
"legal"	(8.5 x 14.0 in)
"ledger"	(17.0 x 11.0 in)
"letter"	(8.5 x 11.0 in)
"tabloid"	(11.0 x 17.0 in)
"11x17"	(11.0 x 17.0 in)
"17x11"	(17.0 x 11.0 in)

#### **Tamaño carta gubernamental del IEEE Printer Working Group, para escritura de niños**

"government-letter"	(8 x 10.5 in)
"government-legal"	(8.5 x 13.0 in)
"philippine-legal"	(8.5 x 13.0 in)

#### **Tamaños de ANSI**

"ansi a"	(8.5 x 11.0 in)
"ansi b"	(17.0 x 11.0 in)
"ansi c"	(17.0 x 22.0 in)
"ansi d"	(22.0 x 34.0 in)
"ansi e"	(34.0 x 44.0 in)
"engineering f"	(28.0 x 40.0 in)

#### **Tamaños arquitectónicos norteamericanos**

"arch a"	(9.0 x 12.0 in)
"arch b"	(12.0 x 18.0 in)



"arch c" (18.0 x 24.0 in)

"arch d" (24.0 x 36.0 in)

"arch e" (36.0 x 48.0 in)

"arch e1" (30.0 x 42.0 in)

**Tamaños antiguos que aún se utilizan en el Reino Unido**

"statement"  
(5.5 x 8.5 in)

"half letter"  
(5.5 x 8.5 in)

"quarto" (8.0 x 10.0 in)

"octavo" (6.75 x 10.5 in)

"executive"  
(7.25 x 10.5 in)

"monarch"  
(7.25 x 10.5 in)

"foolscap"  
(8.27 x 13.0 in)

"folio" (8.27 x 13.0 in)

"super-b"  
(13.0 x 19.0 in)

"post" (15.5 x 19.5 in)

"crown" (15.0 x 20.0 in)

"large post"  
(16.5 x 21.0 in)

"demy" (17.5 x 22.5 in)

"medium" (18.0 x 23.0 in)

"broadsheet"  
(18.0 x 24.0 in)

"royal" (20.0 x 25.0 in)

"elephant"  
(23.0 x 28.0 in)

"double demy"  
(22.5 x 35.0 in)

"quad demy"  
(35.0 x 45.0 in)

"atlas" (26.0 x 34.0 in)

"imperial"  
(22.0 x 30.0 in)

"antiquarian"  
(31.0 x 53.0 in)

### Tamaños basados en PA4

"pa0"	(840 x 1120 mm)
"pa1"	(560 x 840 mm)
"pa2"	(420 x 560 mm)
"pa3"	(280 x 420 mm)
"pa4"	(210 x 280 mm)
"pa5"	(140 x 210 mm)
"pa6"	(105 x 140 mm)
"pa7"	(70 x 105 mm)
"pa8"	(52 x 70 mm)
"pa9"	(35 x 52 mm)
"pa10"	(26 x 35 mm)

### Usado en el Asia suroriental y en Australia

"f4"	(210 x 330 mm)
------	----------------

Usado para los diminutos ejemplos de @lilypond de la documentación; basado en a8 apaisado.

"a8landscape"
(74 x 52 mm)

## A.6 Instrumentos MIDI

A continuación figura una lista con los nombres que se pueden utilizar para la propiedad `midiInstrument` (instrumento MIDI). El orden de los instrumentos que aparece aquí, de arriba a abajo empezando por la columna de la izquierda, corresponde a los 128 números de programa del estándar General MIDI.

acoustic grand	contrabass	lead 7 (fifths)
bright acoustic	tremolo strings	lead 8 (bass+lead)
electric grand	pizzicato strings	pad 1 (new age)
honky-tonk	orchestral harp	pad 2 (warm)
electric piano 1	timpani	pad 3 (polysynth)
electric piano 2	string ensemble 1	pad 4 (choir)
harpsichord	string ensemble 2	pad 5 (bowed)
clav	synthstrings 1	pad 6 (metallic)
celesta	synthstrings 2	pad 7 (halo)
glockenspiel	choir aahs	pad 8 (sweep)
music box	voice oohs	fx 1 (rain)
vibraphone	synth voice	fx 2 (soundtrack)
marimba	orchestra hit	fx 3 (crystal)
xylophone	trumpet	fx 4 (atmosphere)
tubular bells	trombone	fx 5 (brightness)
dulcimer	tuba	fx 6 (goblins)
drawbar organ	muted trumpet	fx 7 (echoes)
percussive organ	french horn	fx 8 (sci-fi)
rock organ	brass section	sitar
church organ	synthbrass 1	banjo
reed organ	synthbrass 2	shamisen

accordion	soprano sax	koto
harmonica	alto sax	kalimba
concertina	tenor sax	bagpipe
acoustic guitar (nylon)	baritone sax	fiddle
acoustic guitar (steel)	oboe	shanai
electric guitar (jazz)	english horn	tinkle bell
electric guitar (clean)	bassoon	agogo
electric guitar (muted)	clarinet	steel drums
overdriven guitar	piccolo	woodblock
distorted guitar	flute	taiko drum
guitar harmonics	recorder	melodic tom
acoustic bass	pan flute	synth drum
electric bass (finger)	blown bottle	reverse cymbal
electric bass (pick)	shakuhachi	guitar fret noise
fretless bass	whistle	breath noise
slap bass 1	ocarina	seashore
slap bass 2	lead 1 (square)	bird tweet
synth bass 1	lead 2 (sawtooth)	telephone ring
synth bass 2	lead 3 (calliope)	helicopter
violin	lead 4 (chiff)	applause
viola	lead 5 (charang)	gunshot
cello	lead 6 (voice)	

## A.7 Lista de colores

### Colores normales

La sintaxis para el uso de colores se encuentra detallada en [Colorear objetos], página 243.

black	white	red	green
blue	cyan	magenta	yellow
grey	darkred	darkgreen	darkblue
darkcyan	darkmagenta	darkyellow	

### Nombre de los colores de CSS

Los nombres de colores de CSS se pueden usar tal cual dentro de argumentos de cadena de caracteres.

aliceblue	darkturquoise	lightsalmon	papayawhip
antiquewhite	darkviolet	lightseagreen	peachpuff
aqua	deeppink	lightskyblue	peru
aquamarine	deepskyblue	lightslategray	pink
azure	dimgray	lightslategrey	plum
beige	dimgrey	lightsteelblue	powderblue
bisque	dodgerblue	lightyellow	purple
black	firebrick	lime	rebeccapurple
blanchedalmond	floralwhite	limegreen	red
blue	forestgreen	linen	rosybrown
blueviolet	fuchsia	magenta	royalblue
brown	gainsboro	maroon	saddlebrown
burlywood	ghostwhite	mediumaquamarine	salmon
cadetblue	gold	mediumblue	sandybrown
chartreuse	goldenrod	mediumorchid	seagreen
chocolate	gray	mediumpurple	seashell

coral	green	mediumseagreen	sienna
cornflowerblue	greenyellow	mediumslateblue	silver
cornsilk	grey	mediumspringgreen	skyblue
crimson	honeydew	mediumturquoise	slateblue
cyan	hotpink	mediumvioletred	slategray
darkblue	indianred	midnightblue	slategrey
darkcyan	indigo	mintcream	snow
darkgoldenrod	ivory	mistyrose	springgreen
darkgray	khaki	moccasin	steelblue
darkgreen	lavender	navajowhite	tan
darkgrey	lavenderblush	navy	teal
darkkhaki	lawngreen	oldlace	thistle
darkmagenta	lemonchiffon	olive	tomato
darkolivegreen	lightblue	olivedrab	turquoise
darkorange	lightcoral	orange	violet
darkorchid	lightcyan	orangered	wheat
darkred	lightgoldenrodyellow	orchid	white
darksalmon	lightgray	palegoldenrod	whitesmoke
darkseagreen	lightgreen	palegreen	yellow
darkslateblue	lightgrey	paleturquoise	yellowgreen
darkslategray	lightpink	palevioletred	
darkslategrey			

Las definiciones de colores de CSS difieren de los nombres de colores de X por los colores siguientes: green, grey, maroon, purple.

## Nombres de los colores de X

Los nombres de los colores de X11 ([https://en.wikipedia.org/wiki/X11\\_color\\_names](https://en.wikipedia.org/wiki/X11_color_names)) ofrecen un abanico de opciones más amplio que los nombres de CSS. Admiten distintas variantes:

- Un nombre que se escribe como una palabra única con mayúsculas intercaladas (p.ej. ‘LightSlateBlue’) se puede escribir también como palabras separadas por espacios, con o sin mayúsculas (p.ej. ‘light slate blue’).
- La palabra ‘grey’ siempre se puede escribir como ‘gray’ (p.ej. ‘DarkSlateGray’), sin ninguna diferencia en el resultado.
- Algunos nombres admiten un sufijo numérico (p.ej. ‘LightSalmon4’).

Las tablas siguientes presentan todos los nombres de colores que se pueden usar sin un sufijo numérico y después el subconjunto de ellos que se pueden usar con un sufijo.

## Nombres de los colores sin sufijo numérico

AliceBlue	LawnGreen	OrangeRed	firebrick
AntiqueWhite	LemonChiffon	PaleGoldenrod	gainsboro
BlanchedAlmond	LightBlue	PaleGreen	gold
BlueViolet	LightCoral	PaleTurquoise	goldenrod
CadetBlue	LightCyan	PaleVioletRed	green
CornflowerBlue	LightGoldenrod	PapayaWhip	grey
DarkBlue	LightGoldenrodYellow	PeachPuff	honeydew
DarkCyan	LightGreen	PowderBlue	ivory
DarkGoldenrod	LightGrey	RosyBrown	khaki
DarkGreen	LightPink	RoyalBlue	lavender
DarkGrey	LightSalmon	SaddleBrown	linen
DarkKhaki	LightSeaGreen	SandyBrown	magenta

DarkMagenta	LightSkyBlue	SeaGreen	maroon
DarkOliveGreen	LightSlateBlue	SkyBlue	moccasin
DarkOrange	LightSlateGrey	SlateBlue	navy
DarkOrchid	LightSteelBlue	SlateGrey	orange
DarkRed	LightYellow	SpringGreen	orchid
DarkSalmon	LimeGreen	SteelBlue	peru
DarkSeaGreen	MediumAquamarine	VioletRed	pink
DarkSlateBlue	MediumBlue	WhiteSmoke	plum
DarkSlateGrey	MediumOrchid	YellowGreen	purple
DarkTurquoise	MediumPurple	aquamarine	red
DarkViolet	MediumSeaGreen	azure	salmon
DeepPink	MediumSlateBlue	beige	seashell
DeepSkyBlue	MediumSpringGreen	bisque	sienna
DimGrey	MediumTurquoise	black	snow
DodgerBlue	MediumVioletRed	blue	tan
FloralWhite	MidnightBlue	brown	thistle
ForestGreen	MintCream	burlywood	tomato
GhostWhite	MistyRose	chartreuse	turquoise
GreenYellow	NavajoWhite	chocolate	violet
HotPink	NavyBlue	coral	wheat
IndianRed	OldLace	cornsilk	white
LavenderBlush	OliveDrab	cyan	yellow

## Nombres de los colores con sufijo numérico

En los nombres siguientes, el sufijo N debe ser un número entero entre 1 y 4, de más claro a más oscuro:

AntiqueWhiteN	LightSkyBlueN	SteelBlueN	khakiN
CadetBlueN	LightSteelBlueN	VioletRedN	magentaN
DarkGoldenrodN	LightYellowN	aquamarineN	maroonN
DarkOliveGreenN	MediumOrchidN	azureN	orangeN
DarkOrangeN	MediumPurpleN	bisqueN	orchidN
DarkOrchidN	MistyRoseN	blueN	pinkN
DarkSeaGreenN	NavajoWhiteN	brownN	plumN
DeepPinkN	OliveDrabN	burlywoodN	purpleN
DeepSkyBlueN	OrangeRedN	chartreuseN	redN
DodgerBlueN	PaleGreenN	chocolateN	salmonN
HotPinkN	PaleTurquoiseN	coralN	seashellN
IndianRedN	PaleVioletRedN	cornsilkN	siennaN
LavenderBlushN	PeachPuffN	cyanN	snowN
LemonChiffonN	RosyBrownN	firebrickN	tanN
LightBlueN	RoyalBlueN	goldN	thistleN
LightCyanN	SeaGreenN	goldenrodN	tomatoN
LightGoldenrodN	SkyBlueN	greenN	turquoiseN
LightPinkN	SlateBlueN	honeydewN	wheatN
LightSalmonN	SpringGreenN	ivoryN	yellowN

## Escala de grises

Se puede obtener una escala de grises utilizando:

`greyN`

donde N es un número entre 0 y 100.

## A.8 La tipografía Emmentaler

La fuente tipográfica Emmentaler consiste en dos *subconjuntos* de glifos. “Feta”, utilizado para la notación clásica, y “Parmesan”, usado para la notación de la música antigua.

Cualquier glifo que esté comprendido dentro de la tipografía Emmentaler se puede acceder directamente usando elementos de marcado de texto junto con el nombre del glifo (según se muestra en las tablas, más adelante). Por ejemplo:

```
g^\markup {\musicglyph "scripts.segno" }
```

or

```
\markup {\musicglyph "five"}
```

Para ver más información, consulte Sección 1.8.2 [Formatear el texto], página 264.

### Glifos de clave

<code>clefs.C</code>		<code>clefs.C_change</code>	
<code>clefs.varC</code>		<code>clefs.varC_change</code>	
<code>clefs.F</code>		<code>clefs.F_change</code>	
<code>clefs.G</code>		<code>clefs.G_change</code>	
<code>clefs.GG</code>		<code>clefs.GG_change</code>	
<code>clefs.tenorG</code>		<code>clefs.tenorG_change</code>	
<code>clefs.percussion</code>		<code>clefs.percussion_change</code>	
<code>clefs.varpercussion</code>		<code>clefs .varpercussion_change</code>	
<code>clefs.tab</code>		<code>clefs.tab_change</code>	

### Glifos de indicación de compás

<code>timesig.C44</code>		<code>timesig.C22</code>	
--------------------------	---	--------------------------	---

## Glifos de cifras







plus	+	comma	,
hyphen	-	period	.
zero	<b>0</b>	one	<b>1</b>
two	<b>2</b>	three	<b>3</b>
four	<b>4</b>	five	<b>5</b>
six	<b>6</b>	seven	<b>7</b>
eight	<b>8</b>	nine	<b>9</b>

## Glifos de alteraciones

accidentals.sharp	#	accidentals .sharp.arrowup	
accidentals .sharp.arrowdown		accidentals .sharp.arrowboth	
accidentals.sharp .slashslash.stem	‡	accidentals.sharp .slashslashslash.stemstem	‡‡
accidentals.sharp .slashslashslash.stem	‡	accidentals .sharp.slash.stem	‡
accidentals.sharp .slashslash.stemstemstem	‡‡	accidentals.doublesharp	‡‡
accidentals.natural	♮	accidentals .natural.arrowup	
accidentals .natural.arrowdown		accidentals .natural.arrowboth	

<code>accidentals.flat</code>		<code>accidentals.flat.arrowup</code>	
<code>accidentals.flat.arrowdown</code>		<code>accidentals.flat.arrowboth</code>	
<code>accidentals.flat.slash</code>		<code>accidentals.flat.slashslash</code>	
<code>accidentals.mirroredflat.flat</code>		<code>accidentals.mirroredflat</code>	
<code>accidentals.mirroredflat.backslash</code>		<code>accidentals.flatflat</code>	
<code>accidentals.flatflat.slash</code>		<code>accidentals.rightparen</code>	
<code>accidentals.leftparen</code>			





## Glifos de las cabezas de nota predeterminadas

<code>noteheads.uM2</code>		<code>noteheads.dM2</code>	
<code>noteheads.sM1</code>		<code>noteheads.s0</code>	
<code>noteheads.s1</code>		<code>noteheads.s2</code>	






















## Glifos de las cabezas de nota especiales































<code>noteheads.sM1double</code>		<code>noteheads.s0diamond</code>	
<code>noteheads.s1diamond</code>		<code>noteheads.s2diamond</code>	
<code>noteheads.s0triangle</code>		<code>noteheads.s1triangle</code>	
<code>noteheads.s2triangle</code>		<code>noteheads.s0slash</code>	



<code>noteheads.s1slash</code>		<code>noteheads.s2slash</code>	
<code>noteheads.s0cross</code>		<code>noteheads.s1cross</code>	
<code>noteheads.s2cross</code>		<code>noteheads.s2xcircle</code>	
<code>noteheads.s0harmonic</code>		<code>noteheads.s2harmonic</code>	

## Glifos de las cabezas de nota con formas




















<code>noteheads.s0do</code>		<code>noteheads.s1do</code>	
<code>noteheads.s2do</code>		<code>noteheads.s0doThin</code>	
<code>noteheads.s1doThin</code>		<code>noteheads.s2doThin</code>	
<code>noteheads.s0re</code>		<code>noteheads.s1re</code>	
<code>noteheads.s2re</code>		<code>noteheads.s0reThin</code>	
<code>noteheads.s1reThin</code>		<code>noteheads.s2reThin</code>	
<code>noteheads.s0mi</code>		<code>noteheads.s1mi</code>	
<code>noteheads.s2mi</code>		<code>noteheads.s0miMirror</code>	
<code>noteheads.s1miMirror</code>		<code>noteheads.s2miMirror</code>	
<code>noteheads.s0miThin</code>		<code>noteheads.s1miThin</code>	
<code>noteheads.s2miThin</code>		<code>noteheads.u0fa</code>	

<code>noteheads.d0fa</code>		<code>noteheads.u1fa</code>	
<code>noteheads.d1fa</code>		<code>noteheads.u2fa</code>	
<code>noteheads.d2fa</code>		<code>noteheads.u0faThin</code>	
<code>noteheads.d0faThin</code>		<code>noteheads.u1faThin</code>	
<code>noteheads.d1faThin</code>		<code>noteheads.u2faThin</code>	
<code>noteheads.d2faThin</code>		<code>noteheads.s0sol</code>	
<code>noteheads.s1sol</code>		<code>noteheads.s2sol</code>	
<code>noteheads.s0la</code>		<code>noteheads.s1la</code>	
<code>noteheads.s2la</code>		<code>noteheads.s0laThin</code>	
<code>noteheads.s1laThin</code>		<code>noteheads.s2laThin</code>	
<code>noteheads.s0ti</code>		<code>noteheads.s1ti</code>	
<code>noteheads.s2ti</code>		<code>noteheads.s0tiThin</code>	
<code>noteheads.s1tiThin</code>		<code>noteheads.s2tiThin</code>	
<code>noteheads.u0doFunk</code>		<code>noteheads.d0doFunk</code>	
<code>noteheads.u1doFunk</code>		<code>noteheads.d1doFunk</code>	







<code>noteheads.u2doFunk</code>	◀	<code>noteheads.d2doFunk</code>	◀
<code>noteheads.u0reFunk</code>	▶	<code>noteheads.d0reFunk</code>	◀
<code>noteheads.u1reFunk</code>	▶	<code>noteheads.d1reFunk</code>	◀
<code>noteheads.u2reFunk</code>	▶	<code>noteheads.d2reFunk</code>	◀
<code>noteheads.u0miFunk</code>	◊	<code>noteheads.d0miFunk</code>	◊
<code>noteheads.u1miFunk</code>	◊	<code>noteheads.d1miFunk</code>	◊
<code>noteheads.s2miFunk</code>	◆	<code>noteheads.u0faFunk</code>	◀
<code>noteheads.d0faFunk</code>	▶	<code>noteheads.u1faFunk</code>	◀
<code>noteheads.d1faFunk</code>	▶	<code>noteheads.u2faFunk</code>	◀
<code>noteheads.d2faFunk</code>	▶	<code>noteheads.s0solFunk</code>	○
<code>noteheads.s1solFunk</code>	○	<code>noteheads.s2solFunk</code>	●
<code>noteheads.s0laFunk</code>	□	<code>noteheads.s1laFunk</code>	□
<code>noteheads.s2laFunk</code>	■	<code>noteheads.u0tiFunk</code>	▶
<code>noteheads.d0tiFunk</code>	◊	<code>noteheads.ultiFunk</code>	▶
<code>noteheads.d1tiFunk</code>	◊	<code>noteheads.u2tiFunk</code>	▶

<code>noteheads.d2tiFunk</code>	◀	<code>noteheads.s0doWalker</code>	▷
<code>noteheads.u1doWalker</code>	▽	<code>noteheads.d1doWalker</code>	▷
<code>noteheads.u2doWalker</code>	▼	<code>noteheads.d2doWalker</code>	▲
<code>noteheads.s0reWalker</code>	◁	<code>noteheads.u1reWalker</code>	▷
<code>noteheads.d1reWalker</code>	◁	<code>noteheads.u2reWalker</code>	►
<code>noteheads.d2reWalker</code>	◀	<code>noteheads.s0miWalker</code>	◇
<code>noteheads.s1miWalker</code>	◇	<code>noteheads.s2miWalker</code>	◆
<code>noteheads.s0faWalker</code>	▷	<code>noteheads.u1faWalker</code>	▽
<code>noteheads.d1faWalker</code>	▷	<code>noteheads.u2faWalker</code>	▼
<code>noteheads.d2faWalker</code>	▲	<code>noteheads.s0laWalker</code>	◻
<code>noteheads.s1laWalker</code>	◻	<code>noteheads.s2laWalker</code>	■
<code>noteheads.s0tiWalker</code>	◁	<code>noteheads.ultiWalker</code>	▷
<code>noteheads.d1tiWalker</code>	◁	<code>noteheads.u2tiWalker</code>	►
<code>noteheads.d2tiWalker</code>	◀		

## Glifos de silencios

<code>rests.0</code>		<code>rests.1</code>	
<code>rests.0o</code>		<code>rests.1o</code>	
<code>rests.M3</code>		<code>rests.M2</code>	
<code>rests.M1</code>		<code>rests.M1o</code>	
<code>rests.2</code>		<code>rests.2classical</code>	
<code>rests.2z</code>		<code>rests.3</code>	
<code>rests.4</code>		<code>rests.5</code>	
<code>rests.6</code>		<code>rests.7</code>	
<code>rests.8</code>		<code>rests.9</code>	
<code>rests.10</code>			

## Glifos de corchetes

<code>flags.u3</code>		<code>flags.u4</code>	
<code>flags.u5</code>		<code>flags.u6</code>	
<code>flags.u7</code>		<code>flags.u8</code>	




## Glifos de inscripciones









<code>scripts.ufermata</code>		<code>scripts.dfermata</code>	
<code>scripts</code> <code>.uhenzeshortfermata</code>		<code>scripts</code> <code>.dhenzeshortfermata</code>	
<code>scripts</code> <code>.uhenzelongfermata</code>		<code>scripts</code> <code>.dhenzelongfermata</code>	
<code>scripts.ushortfermata</code>		<code>scripts.dshortfermata</code>	
<code>scripts</code> <code>.uveryshortfermata</code>		<code>scripts</code> <code>.dveryshortfermata</code>	
<code>scripts.ulongfermata</code>		<code>scripts.dlongfermata</code>	
<code>scripts.uverylongfermata</code>		<code>scripts.dverylongfermata</code>	
<code>scripts.thumb</code>		<code>scripts.sforzato</code>	
<code>scripts.espr</code>		<code>scripts.staccato</code>	
<code>scripts.ustaccatissimo</code>		<code>scripts.dstaccatissimo</code>	
<code>scripts.tenuto</code>		<code>scripts.uportato</code>	
<code>scripts.dportato</code>		<code>scripts.umarcato</code>	
<code>scripts.dmarcato</code>		<code>scripts.open</code>	
<code>scripts.halfopen</code>		<code>scripts.halfopenvertical</code>	

scripts.stopped	+	scripts.upbow	V
scripts.downbow	▮	scripts.reverseturn	∞
scripts.turn	∞	scripts.slashturn	∞
scripts.haydnturn	↻	scripts.trill	tr
scripts.upedalheel	U	scripts.dpedalheel	u
scripts.upedaltoe	V	scripts.dpedaltoe	^
scripts.flageolet	○	scripts.segno	♫
scripts.varsegno	∞	scripts.coda	⦿
scripts.varcoda	⦿	scripts.rcomma	,
scripts.lcomma	(	scripts.rvarcomma	/
scripts.lvarcomma	/	scripts.arpeggio	↗
scripts.trill_element	~	scripts.arpeggio .arrow.M1	↘
scripts.arpeggio.arrow.1	↗	scripts.trillelement	↘
scripts.prall	∞	scripts.mordent	∞
scripts.prallprall	∞	scripts.prallmordent	∞



<code>scripts.upprall</code>		<code>scripts.upmordent</code>	
<code>scripts.prallup</code>		<code>scripts.downprall</code>	
<code>scripts.downmordent</code>		<code>scripts.pralldown</code>	
<code>scripts.lineprall</code>		<code>scripts.caesura.curved</code>	
<code>scripts.caesura.straight</code>		<code>scripts.tickmark</code>	
<code>scripts.snappizzicato</code>		<code>scripts.ictus</code>	
<code>scripts.uaccentus</code>		<code>scripts.daccentus</code>	
<code>scripts.usemicirculus</code>		<code>scripts.dsemicirculus</code>	
<code>scripts.circulus</code>		<code>scripts.augmentum</code>	
<code>scripts</code> <code>.usignumcongruentiae</code>		<code>scripts</code> <code>.dsignumcongruentiae</code>	








## Glifos de flechas

<code>arrowheads.open.01</code>		<code>arrowheads.open.0M1</code>	
<code>arrowheads.open.11</code>		<code>arrowheads.open.1M1</code>	
<code>arrowheads.close.01</code>		<code>arrowheads.close.0M1</code>	
<code>arrowheads.close.11</code>		<code>arrowheads.close.1M1</code>	









## Glifos de puntas de corchete

<code>brackettips.up</code>		<code>brackettips.down</code>	
-----------------------------	---	-------------------------------	---



Glifos de pedal

pedal.*		pedal.M	
pedal..		pedal.P	
pedal.d		pedal.e	
pedal.Ped			









Glifos de acordeón

accordion.discant		accordion.dot	
accordion.freebass		accordion.stdbass	
accordion.bayanbass		accordion.oldEE	
accordion.push		accordion.pull	

Glifos de ligadura

ties.lyric.short		ties.lyric.default	
------------------	---	--------------------	---

Glifos de vaticana

clefs.vaticana.do		clefs.vaticana.do_change	
clefs.vaticana.fa		clefs.vaticana.fa_change	
custodes.vaticana.u0		custodes.vaticana.u1	
custodes.vaticana.u2		custodes.vaticana.d0	

custodes.vaticana.d1	┆	custodes.vaticana.d2	┆
accidentals.vaticanaM1	♭	accidentals.vaticana0	♮
dots.dotvaticana	.	noteheads .svaticana.punctum	■
noteheads.svaticana .punctum.cavum	◻	noteheads.svaticana .linea.punctum	■
noteheads.svaticana .linea.punctum.cavum	◻	noteheads.svaticana .inclinatum	◆
noteheads.svaticana.lpes	■	noteheads .svaticana.vlpes	■
noteheads.svaticana.upes	■	noteheads .svaticana.vupes	■
noteheads .svaticana.plica	.	noteheads .svaticana.vplica	,
noteheads .svaticana.epiphonus	■	noteheads.svaticana .vepiphonus	■
noteheads.svaticana .reverse.plica	.	noteheads.svaticana .reverse.vplica	.
noteheads.svaticana .inner.cephalicus	■	noteheads.svaticana .cephalicus	■
noteheads .svaticana.quilisma	■		

## Glifos de medicaea





























clefs.medicaea.do	⦿	clefs.medicaea.do_change	⦿
clefs.medicaea.fa	⦿	clefs.medicaea.fa_change	⦿
custodes.medicaea.u0	┆	custodes.medicaea.u1	┆

custodes.medicaea.u2	┆	custodes.medicaea.d0	┆
custodes.medicaea.d1	┆	custodes.medicaea.d2	┆
accidentals.medicaeaM1	♭	noteheads.smedicaea .inclinatum	◆
noteheads .smedicaea.punctum	■	noteheads .smedicaea.rvirga	┆■
noteheads .smedicaea.virga	┆■		





























## Glifos de Hufnagel













clefs.hufnagel.do	ꞑ	clefs.hufnagel.do_change	ꞑ
clefs.hufnagel.fa	ꝑ	clefs.hufnagel.fa_change	ꝑ
clefs.hufnagel.do.fa	ꝑꝑ	clefs.hufnagel .do.fa_change	ꝑꝑ
custodes.hufnagel.u0	✓	custodes.hufnagel.u1	✓
custodes.hufnagel.u2	✓	custodes.hufnagel.d0	↘
custodes.hufnagel.d1	↘	custodes.hufnagel.d2	↘
accidentals.hufnagelM1	♭	noteheads .shufnagel.punctum	◆
noteheads .shufnagel.virga	↑	noteheads.shufnagel.lpes	▮

## Glifos de mensural



















rests.M3mensural		rests.M2mensural	
rests.M1mensural		rests.0mensural	
rests.1mensural		rests.2mensural	
rests.3mensural		rests.4mensural	
clefs.mensural.c		clefs.mensural.c_change	
clefs.blackmensural.c		clefs.blackmensural.c_change	
clefs.mensural.f		clefs.mensural.f_change	
clefs.mensural.g		clefs.mensural.g_change	
custodes.mensural.u0		custodes.mensural.u1	
custodes.mensural.u2		custodes.mensural.d0	
custodes.mensural.d1		custodes.mensural.d2	
accidentals.mensural1		accidentals.mensuralM1	
flags.mensuralu03		flags.mensuralu13	
flags.mensuralu23		flags.mensurald03	

flags.mensurald13	{	flags.mensurald23	{
flags.mensuralu04	}	flags.mensuralu14	}
flags.mensuralu24	}	flags.mensurald04	{
flags.mensurald14	{	flags.mensurald24	{
flags.mensuralu05	}	flags.mensuralu15	}
flags.mensuralu25	}	flags.mensurald05	{
flags.mensurald15	{	flags.mensurald25	{
flags.mensuralu06	}	flags.mensuralu16	}
flags.mensuralu26	}	flags.mensurald06	{
flags.mensurald16	{	flags.mensurald26	{
timesig.mensural44	C	timesig.mensural22	¢
timesig.mensural32	O	timesig.mensural64	©
timesig.mensural94	⊙	timesig.mensural34	ϕ
timesig.mensural68	¢	timesig.mensural98	ϕ

















timesig.mensural48		timesig.mensural68alt	
timesig.mensural24		noteheads.uM3mensural	
noteheads.dM3mensural		noteheads.sM3ligmensural	
noteheads.uM2mensural		noteheads.dM2mensural	
noteheads.sM2ligmensural		noteheads.sM1mensural	
noteheads.urM3mensural		noteheads.drM3mensural	
noteheads .srM3ligmensural		noteheads.urM2mensural	
noteheads.drM2mensural		noteheads .srM2ligmensural	
noteheads.srM1mensural		noteheads .uM3semimensural	
noteheads .dM3semimensural		noteheads .sM3semiligmensural	
noteheads .uM2semimensural		noteheads .dM2semimensural	
noteheads .sM2semiligmensural		noteheads .sM1semimensural	
noteheads .urM3semimensural		noteheads .drM3semimensural	
noteheads .srM3semiligmensural		noteheads .urM2semimensural	
noteheads .drM2semimensural		noteheads .srM2semiligmensural	

noteheads .srM1semimensural		noteheads .uM3blackmensural	
noteheads .dM3blackmensural		noteheads .sM3blackligmensural	
noteheads .uM2blackmensural		noteheads .dM2blackmensural	
noteheads .sM2blackligmensural		noteheads .sM1blackmensural	
noteheads.s0mensural		noteheads.s1mensural	
noteheads.s2mensural		noteheads .s0blackmensural	













## Glifos de neomensural







rests.M3neomensural		rests.M2neomensural	
rests.M1neomensural		rests.0neomensural	
rests.1neomensural		rests.2neomensural	
rests.3neomensural		rests.4neomensural	
clefs.neomensural.c		clefs.neomensural .c_change	
timesig.neomensural44		timesig.neomensural22	
timesig.neomensural32		timesig.neomensural64	
timesig.neomensural94		timesig.neomensural34	
timesig.neomensural68		timesig.neomensural98	










timesig.neomensural48		timesig.neomensural68alt	
timesig.neomensural24		noteheads.uM3neomensural	
noteheads.dM3neomensural		noteheads.uM2neomensural	
noteheads.dM2neomensural		noteheads.sM1neomensural	
noteheads .urM3neomensural		noteheads .drM3neomensural	
noteheads .urM2neomensural		noteheads .drM2neomensural	
noteheads .srM1neomensural		noteheads.s0neomensural	
noteheads.s1neomensural		noteheads.s2neomensural	

## Glifos de Petrucci
















clefs.petrucci.c1		clefs.petrucci.c1_change	
clefs.petrucci.c2		clefs.petrucci.c2_change	
clefs.petrucci.c3		clefs.petrucci.c3_change	
clefs.petrucci.c4		clefs.petrucci.c4_change	
clefs.petrucci.c5		clefs.petrucci.c5_change	
clefs.petrucci.f		clefs.petrucci.f_change	

<code>clefs.petrucchi.g</code>		<code>clefs.petrucchi.g_change</code>	
<code>noteheads.s0petrucci</code>		<code>noteheads.s1petrucci</code>	
<code>noteheads.s2petrucci</code>		<code>noteheads.s0blackpetrucci</code>	
<code>noteheads.s1blackpetrucci</code>		<code>noteheads.s2blackpetrucci</code>	

## Glifos de Solesmes

<code>noteheads.ssolesmes.incl.parvum</code>		<code>noteheads.ssolesmes.auct.asc</code>	
<code>noteheads.ssolesmes.auct.desc</code>		<code>noteheads.ssolesmes.incl.auctum</code>	
<code>noteheads.ssolesmes.stropha</code>		<code>noteheads.ssolesmes.stropha.aucta</code>	
<code>noteheads.ssolesmes.oriscus</code>			

## Glifos de la notación del canto kievano

<code>clefs.kievan.do</code>		<code>clefs.kievan.do_change</code>	
<code>accidentals.kievan1</code>		<code>accidentals.kievanM1</code>	
<code>scripts.barline.kievan</code>		<code>dots.dotkievan</code>	
<code>noteheads.sM2kievan</code>		<code>noteheads.sM1kievan</code>	
<code>noteheads.s0kievan</code>		<code>noteheads.d2kievan</code>	
<code>noteheads.u2kievan</code>		<code>noteheads.s1kievan</code>	
<code>noteheads.sr1kievan</code>		<code>noteheads.d3kievan</code>	
<code>noteheads.u3kievan</code>			

## A.9 Estilos de cabezas de nota

Se pueden usar los siguientes estilos para las cabezas de las notas.

The image displays 12 musical staves, each showing a different style of note head. The staves are arranged in six pairs, with each pair labeled above. The styles are: **default**, **altdefault**, **baroque**, **neomensural**, **mensural**, **petrucci**, **harmonic**, **harmonic-black**, **harmonic-mixed**, **diamond**, **cross**, **xcircle**, **triangle**, and **slash**. Each staff shows a sequence of notes in a 12/8 time signature, demonstrating how the chosen style affects the appearance of the note heads.

## A.10 Estilos de clave

La tabla siguiente presenta todos los estilos de clave posibles y el lugar del *Do central* en cada una.

## Claves estándar

Ejemplo

Salida

Ejemplo

Salida

`\clef G`



`\clef "G2"`



`\clef treble`



`\clef violin`



`\clef french`



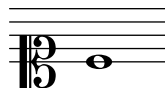
`\clef GG`



`\clef tenorG`



`\clef soprano`



`\clef mezzosoprano`



`\clef C`



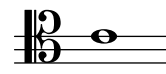
`\clef alto`



`\clef tenor`



`\clef baritone`



`\clef varC`



`\clef altovarC`



`\clef tenorvarC`



`\clef baritonevarC`



`\clef varbaritone`



`\clef baritonevarF`



`\clef F`



`\clef bass`



`\clef subbass`



## Clave de pentagramas de percusión

**Ejemplo**

**Salida**

`\clef percussion`



**Ejemplo**

**Salida**

`\clef varpercussion`



## Claves de pautas de tablatura

**Ejemplo**

**Salida**

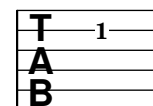
```
\new TabStaff {
  \clef tab
}
```



**Ejemplo**

**Salida**

```
\new TabStaff {
  \clef moderntab
}
```



## Claves de la música antigua

Gregoriano

**Ejemplo**

**Salida**

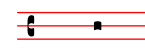
`\clef "vaticana-do1"`



**Ejemplo**

**Salida**

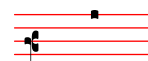
`\clef "vaticana-do2"`



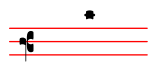
`\clef "vaticana-do3"`



`\clef "vaticana-fa1"`



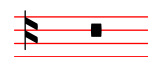
`\clef "vaticana-fa2"`



`\clef "medicaea-do1"`



`\clef "medicaea-do2"`



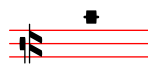
`\clef "medicaea-do3"`



`\clef "medicaea-fa1"`



`\clef "medicaea-fa2"`



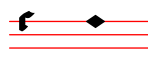
`\clef "hufnagel-do1"`



`\clef "hufnagel-do2"`



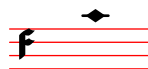
`\clef "hufnagel-do3"`



`\clef "hufnagel-fa1"`



`\clef "hufnagel-fa2"`



`\clef  
"hufnagel-do-fa"`



## Mensural

**Ejemplo**

**Salida**

**Ejemplo**

**Salida**

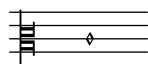
`\clef "mensural-c1"`



`\clef "mensural-c2"`



`\clef "mensural-c3"`



`\clef "mensural-c4"`



`\clef "mensural-c5"`



`\clef "mensural-f"`



`\clef "mensural-g"`



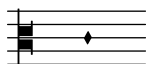
`\clef "blackmensural-c1"`



`\clef  
"blackmensural-c2"`



`\clef "blackmensural-c3"`



`\clef  
"blackmensural-c4"`



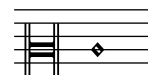
`\clef "blackmensural-c5"`



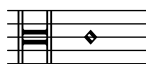
`\clef "neomensural-c1"`



`\clef "neomensural-c2"`



`\clef "neomensural-c3"`



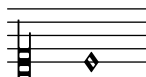
`\clef "neomensural-c4"`



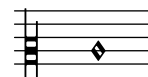
`\clef "neomensural-c5"`



`\clef "petrucci-c1"`



`\clef "petrucci-c2"`



`\clef "petrucci-c3"`



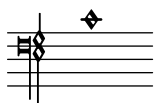
`\clef "petrucci-c4"`



`\clef "petrucci-c5"`



`\clef "petrucci-f"`



`\clef "petrucci-f2"`



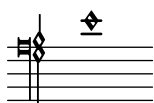
`\clef "petrucci-f3"`



`\clef "petrucci-f4"`



`\clef "petrucci-f5"`



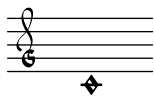
`\clef "petrucci-g1"`



`\clef "petrucci-g2"`



`\clef "petrucci-g"`

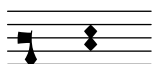


## Kievanas

### Ejemplo

`\clef "kievan-do"`

### Salida



## A.11 Instrucciones de marcado de texto

The following commands can all be used inside `\markup { }`.



### A.11.1 Font

`\abs-fontsize` *size* (number) *arg* (markup)

Use *size* as the absolute font size (in points) to display *arg*. Adjusts `baseline-skip` and `word-space` accordingly.

```
\markup {
  default text font size
  \hspace #2
  \abs-fontsize #16 { text font size 16 }
  \hspace #2
  \abs-fontsize #12 { text font size 12 }
}
```

default text font size    **text font size 16**    text font size 12

Used properties:

- `baseline-skip` (3)
- `word-space` (0.6)

`\bold` *arg* (markup)

Switch to bold font-series.

```
\markup {
  default
  \hspace #2
  \bold
  bold
}
```

default    **bold**

`\box` *arg* (markup)

Draw a box round *arg*. Looks at `thickness`, `box-padding` and `font-size` properties to determine line thickness and padding around the markup.

```
\markup {
  \override #'(box-padding . 0.5)
  \box
  \line { V. S. }
}
```

V. S.

Used properties:

- `box-padding` (0.2)
- `font-size` (0)
- `thickness` (1)

`\caps` *arg* (markup)

Copy of the `\smallCaps` command.

```
\markup {
  default
  \hspace #2
  \caps {
```

```

        Text in small caps
    }
}

default    TEXT IN SMALL CAPS

```

`\dynamic arg` (markup)

Use the dynamic font. This font only contains **s**, **f**, **m**, **z**, **p**, and **r**. When producing phrases, like ‘più **f**’, the normal words (like ‘più’) should be done in a different font. The recommended font for this is bold and italic.

```

\markup {
  \dynamic {
    sfzp
  }
}

```

***sfzp***

`\finger arg` (markup)

Set *arg* as small numbers.

```

\markup {
  \finger {
    1 2 3 4 5
  }
}

```

**1 2 3 4 5**

`\fontCaps arg` (markup)

Set *font-shape* to caps

Note: `\fontCaps` requires the installation and selection of fonts which support the caps font shape.

`\fontsize increment` (number) *arg* (markup)

Add *increment* to the font-size. Adjusts *baseline-skip* accordingly.

```

\markup {
  default
  \hspace #2
  \fontsize #-1.5
  smaller
}

```

default    smaller

Used properties:

- *baseline-skip* (2)
- *word-space* (1)
- *font-size* (0)

`\huge arg` (markup)

Set font size to +2.

```

\markup {
  default

```

```

\hspace #2
\huge
huge
}

default huge

```

`\italic arg` (markup)  
 Use italic font-shape for *arg*.

```

\markup {
  default
  \hspace #2
  \italic
  italic
}

default italic

```

`\large arg` (markup)  
 Set font size to +1.

```

\markup {
  default
  \hspace #2
  \large
  large
}

default large

```

`\larger arg` (markup)  
 Increase the font size relative to the current setting.

```

\markup {
  default
  \hspace #2
  \larger
  larger
}

default larger

```

`\magnify sz (number) arg` (markup)  
 Set the font magnification for its argument. In the following example, the middle A is 10% larger:

```
A \magnify #1.1 { A } A
```

Note: Magnification only works if a font name is explicitly selected. Use `\fontsize` otherwise.

```

\markup {
  default
  \hspace #2
  \magnify #1.5 {
    50% larger
  }
}

```

}

default **50% larger**

`\medium arg` (markup)

Switch to medium font-series (in contrast to bold).

```
\markup {
  \bold {
    some bold text
    \hspace #2
    \medium {
      medium font series
    }
    \hspace #2
    bold again
  }
}
```

**some bold text** medium font series **bold again**

`\normal-size-sub arg` (markup)

Set *arg* in subscript with a normal font size.

```
\markup {
  default
  \normal-size-sub {
    subscript in standard size
  }
}
```

default subscript in standard size

Used properties:

- `font-size (0)`

`\normal-size-super arg` (markup)

Set *arg* in superscript with a normal font size.

```
\markup {
  default
  \normal-size-super {
    superscript in standard size
  }
}
```

default superscript in standard size

Used properties:

- `font-size (0)`

`\normal-text arg` (markup)

Set all font related properties (except the size) to get the default normal text font, no matter what font was used earlier.

```
\markup {
```

```
\huge \bold \sans \caps {
  huge bold sans caps
  \hspace #2
  \normal-text {
    huge normal
  }
  \hspace #2
  as before
}
```

**HUGE BOLD SANS CAPS**   huge normal   **AS BEFORE**

`\normalsize` *arg* (markup)

Set font size to default.

```
\markup {
  \teeny {
    this is very small
    \hspace #2
    \normalsize {
      normal size
    }
    \hspace #2
    teeny again
  }
}
```

this is very small   **normal size**   teeny again

`\number` *arg* (markup)

Set font family to **number**, which yields the font used for time signatures and fingerings. This font contains numbers and some punctuation; it has no letters.

```
\markup {
  \number {
    0 1 2 3 4 5 6 7 8 9 . ,
  }
}
```

**0123456789.,**

`\overtie` *arg* (markup)

Overtie *arg*.

```
\markup \line {
  \overtie "overtied"
  \override #'((offset . 5) (thickness . 1))
  \overtie "overtied"
  \override #'((offset . 1) (thickness . 5))
  \overtie "overtied"
}
```



Used properties:

- `shorten-pair` ((0 . 0))
- `height-limit` (0.7)
- `direction` (1)
- `offset` (2)
- `thickness` (1)

`\replace` *replacements* (list) *arg* (markup)

Used to automatically replace a string by another in the markup *arg*. Each pair of the alist *replacements* specifies what should be replaced. The **key** is the string to be replaced by the **value** string.

```
\markup \replace #'(("thx" . "Thanks!")) thx
```

**Thanks!**

Used properties:

- `replacement-alist`

`\roman` *arg* (markup)

Set font family to roman.

```
\markup {
  \sans \bold {
    sans serif, bold
    \hspace #2
    \roman {
      text in roman font family
    }
    \hspace #2
    return to sans
  }
}
```

**sans serif, bold    text in roman font family    return to sans**

`\sans` *arg* (markup)

Switch to the sans serif font family.

```
\markup {
  default
  \hspace #2
  \sans {
    sans serif
  }
}
```

**default    sans serif**

`\simple` *str* (string)

A simple text string; `\markup { foo }` is equivalent with `\markup { \simple #"foo" }`.

Note: for creating standard text markup or defining new markup commands, the use of `\simple` is unnecessary.

```
\markup {
```

```
\simple #"simple"
\simple #"text"
\simple #"strings"
}
```

**simple text strings**

**\small *arg* (markup)**  
Set font size to -1.

```
\markup {
  default
  \hspace #2
  \small
  small
}
```

**default    small**

**\smallCaps *arg* (markup)**  
Emit *arg* as small caps.  
Note: \smallCaps does not support accented characters.

```
\markup {
  default
  \hspace #2
  \smallCaps {
    Text in small caps
  }
}
```

**default    TEXT IN SMALL CAPS**

**\smaller *arg* (markup)**  
Decrease the font size relative to the current setting.

```
\markup {
  \fontsize #3.5 {
    large text
    \hspace #2
    \smaller { smaller text }
    \hspace #2
    large text
  }
}
```

**large text    smaller text    large text**

**\sub *arg* (markup)**  
Set *arg* in subscript.

```
\markup {
  \concat {
    H
    \sub {
      2
    }
  }
}
```

```

    }
  0
}
}

```

$\text{H}_2\text{O}$

Used properties:

- `font-size` (0)

`\super arg` (markup)

Set *arg* in superscript.

```

\markup {
  E =
  \concat {
    mc
    \super
    2
  }
}

```

$E = mc^2$

Used properties:

- `font-size` (0)

`\teeny arg` (markup)

Set font size to -3.

```

\markup {
  default
  \hspace #2
  \teeny
  teeny
}

```

**default**    teeny

`\text arg` (markup)

Use a text font instead of music symbol or music alphabet font.

```

\markup {
  \number {
    1, 2,
    \text {
      three, four,
    }
  }
  5
}
}

```

**1,2**, three, four, **5**

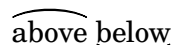
`\tie arg` (markup)

Adds a horizontal bow created with `make-tie-stencil` at bottom or top of *arg*. Looks at `thickness` to determine line thickness, and `offset` to determine y-offset.



The added bow fits the extent of *arg*, **shorten-pair** may be used to modify this. *direction* may be set using an **override** or direction-modifiers or **voiceOne**, etc.

```
\markup {
  \override #'(direction . 1)
  \tie "above"
  \override #'(direction . -1)
  \tie "below"
}
```



Used properties:

- **shorten-pair** ((0 . 0))
- **height-limit** (0.7)
- **direction** (1)
- **offset** (2)
- **thickness** (1)

**\tiny** *arg* (markup)

Set font size to -2.

```
\markup {
  default
  \hspace #2
  \tiny
  tiny
}
```

**default** tiny

**\typewriter** *arg* (markup)

Use font-family typewriter for *arg*.

```
\markup {
  default
  \hspace #2
  \typewriter
  typewriter
}
```

**default** typewriter

**\underline** *arg* (markup)

Underline *arg*. Looks at **thickness** to determine line thickness, **offset** to determine line y-offset from *arg* and **underline-skip** to determine the distance of additional lines from the others. **underline-shift** is used to get subsequent calls correct. Overriding it makes little sense, it would end up adding the provided value to the one of **offset**.

```
\markup \justify-line {
  \underline "underlined"
  \override #'(offset . 5)
  \override #'(thickness . 1)
  \underline "underlined"
```

```

\override #'(offset . 1)
\override #'(thickness . 5)
\underline "underlined"
\override #'(offset . 5)
\override #'(underline-skip . 4)
\underline \underline \underline "multiple underlined"
}

```

underlined    underlined    underlined    multiple underlined

Used properties:

- underline-skip (2)
- underline-shift (0)
- offset (2)
- thickness (1)

```

\undertie arg (markup)
  \markup \line {
    \undertie "undertied"
    \override #'((offset . 5) (thickness . 1))
    \undertie "undertied"
    \override #'((offset . 1) (thickness . 5))
    \undertie "undertied"
  }

```

undertied undertied undertied

Used properties:

- shorten-pair ((0 . 0))
- height-limit (0.7)
- direction (1)
- offset (2)
- thickness (1)

`\upright arg (markup)`

Set font-shape to upright. This is the opposite of *italic*.

```

\markup {
  \italic {
    italic text
    \hspace #2
    \upright {
      upright text
    }
    \hspace #2
    italic again
  }
}

```

*italic text*    upright text    *italic again*

### A.11.2 Align

`\center-align` *arg* (markup)

Align *arg* to its X center.

```
\markup {
  \column {
    one
    \center-align
    two
    three
  }
}

one
two
three
```

`\center-column` *args* (markup list)

Put *args* in a centered column.

```
\markup {
  \center-column {
    one
    two
    three
  }
}

one
two
three
```

Used properties:

- `baseline-skip`

`\column` *args* (markup list)

Stack the markups in *args* vertically. The property `baseline-skip` determines the space between markups in *args*.

```
\markup {
  \column {
    one
    two
    three
  }
}

one
two
three
```

Used properties:

- `baseline-skip`

`\combine` *arg1* (markup) *arg2* (markup)

Print two markups on top of each other.

Note: `\combine` cannot take a list of markups enclosed in curly braces as an argument; for this purpose use `\overlay` instead.

```
\markup {
  \fontsize #5
  \override #'(thickness . 2)
  \combine
  \draw-line #'(0 . 4)
  \arrow-head #Y #DOWN ##f
}
```



`\concat args` (markup list)

Concatenate *args* in a horizontal line, without spaces in between. Strings and simple markups are concatenated on the input level, allowing ligatures. For example, `\concat { "f" \simple #"i" }` is equivalent to "fi".

```
\markup {
  \concat {
    one
    two
    three
  }
}
```

onetwothree

`\dir-column args` (markup list)

Make a column of *args*, going up or down, depending on the setting of the `direction` layout property.

```
\markup {
  \override #`(direction . ,UP)
  \dir-column {
    going up
  }
  \hspace #1
  \dir-column {
    going down
  }
  \hspace #1
  \override #'(direction . 1)
  \dir-column {
    going up
  }
}
```

```
up          up
going going going
          down
```

Used properties:

- `baseline-skip`

- `direction`

`\fill-line` *args* (markup list)

Put *markups* in a horizontal line of width *line-width*. The markups are spaced or flushed to fill the entire line. If there are no arguments, return an empty stencil.

```
\markup {
  \column {
    \fill-line {
      Words evenly spaced across the page
    }
    \null
    \fill-line {
      \line { Text markups }
      \line {
        \italic { evenly spaced }
      }
      \line { across the page }
    }
    \null
    \override #'(line-width . 50)
    \fill-line {
      Width explicitly specified
    }
  }
}
```

Words evenly spaced across the page

Text markups *evenly spaced* across the page

Width explicitly specified

Used properties:

- `line-width` (`#f`)
- `word-space` (0.6)
- `text-direction` (1)

`\fill-with-pattern` *space* (number) *dir* (direction) *pattern* (markup) *left* (markup) *right* (markup)

Put *left* and *right* in a horizontal line of width *line-width* with a line of markups *pattern* in between. Patterns are spaced apart by *space*. Patterns are aligned to the *dir* markup.

```
\markup \column {
  "right-aligned :"
  \fill-with-pattern #1 #RIGHT . first right
  \fill-with-pattern #1 #RIGHT . second right
  \null
  "center-aligned :"
  \fill-with-pattern #1.5 #CENTER - left right
  \null
  "left-aligned :"
  \override #'(line-width . 50)
```

```
\fill-with-pattern #2 #LEFT : left first
\override #'(line-width . 50)
\fill-with-pattern #2 #LEFT : left second
}
```

```
right-aligned :
first .....right
second .....right
```

```
center-aligned :
left - - - - - right
```

```
left-aligned :
left: : : : : : : : : : : : : : : first
left: : : : : : : : : : : : : : : second
```

Used properties:

- line-width
- word-space

`\general-align` *axis* (integer) *dir* (number) *arg* (markup)  
Align *arg* in *axis* direction to the *dir* side.

```
\markup {
  \column {
    one
    \general-align #X #LEFT
    two
    three
    \null
    one
    \general-align #X #CENTER
    two
    three
    \null
    \line {
      one
      \general-align #Y #UP
      two
      three
    }
    \null
    \line {
      one
      \general-align #Y #3.2
      two
      three
    }
  }
}
```

}

one  
two  
three

one  
two  
three

one two three

one three  
two

`\halign` *dir* (number) *arg* (markup)

Set horizontal alignment. If *dir* is -1, then it is left-aligned, while +1 is right. Values in between interpolate alignment accordingly.

```
\markup {
  \column {
    one
    \halign #LEFT
    two
    three
    \null
    one
    \halign #CENTER
    two
    three
    \null
    one
    \halign #RIGHT
    two
    three
    \null
    one
    \halign #-5
    two
    three
  }
}
```

```

}

one
two
three

one
two
three

one
two
three

one
two
three

```

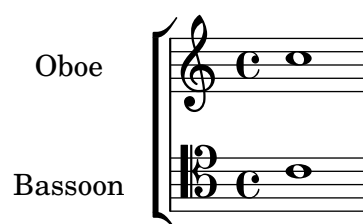
`\hcenter-in length (number) arg (markup)`

Center *arg* horizontally within a box of extending *length*/2 to the left and right.

```

\new StaffGroup <<
  \new Staff {
    \set Staff.instrumentName = \markup {
      \hcenter-in #12
      Oboe
    }
    c''1
  }
  \new Staff {
    \set Staff.instrumentName = \markup {
      \hcenter-in #12
      Bassoon
    }
    \clef tenor
    c'1
  }
>>

```



`\hspace amount (number)`

Create an invisible object taking up horizontal space *amount*.

```

\markup {
  one
  \hspace #2
}

```



```

two
\hspace #8
three
}

one two three

```

`\justify-field` *symbol* (symbol)

Justify the data which has been assigned to *symbol*.

```

\header {
  title = "My title"
  myText = "Lorem ipsum dolor sit amet, consectetur
    adipisicing elit, sed do eiusmod tempor incididunt
    ut labore et dolore magna aliqua. Ut enim ad minim
    veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat."
}

```

```

\paper {
  bookTitleMarkup = \markup {
    \column {
      \fill-line { \fromproperty #'header:title }
      \null
      \justify-field #'header:myText
    }
  }
}

```

```

\markup {
  \null
}

```

My title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

`\justify-line` *args* (markup list)

Put *markups* in a horizontal line of width *line-width*. The markups are spread to fill the entire line and separated by equal space. If there are no arguments, return an empty stencil.

```

\markup {
  \justify-line {
    Constant space between neighboring words
  }
}

```

Constant space between neighboring words

Used properties:

- `line-width` (`#f`)
- `word-space` (0.6)
- `text-direction` (1)

`\justify` *args* (markup list)

Like `\wordwrap`, but with lines stretched to justify the margins. Use `\override #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

```
\markup {
  \justify {
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore
    magna aliqua. Ut enim ad minim veniam, quis nostrud
    exercitation ullamco laboris nisi ut aliquip ex ea
    commodo consequat.
  }
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (`#f`)
- `baseline-skip`

`\justify-string` *arg* (string)

Justify a string. Paragraphs may be separated with double newlines

```
\markup {
  \override #'(line-width . 40)
  \justify-string #"Lorem ipsum dolor sit amet, consectetur
  adipisicing elit, sed do eiusmod tempor incididunt ut
  labore et dolore magna aliqua.
```

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum"

}

Lorem ipsum dolor sit amet, consectetur  
adipiscing elit, sed do eiusmod tempor  
incididunt ut labore et dolore magna  
aliqua.

Ut enim ad minim veniam, quis nostrud  
exercitation ullamco laboris nisi ut  
aliquip ex ea commodo consequat.

Excepteur sint occaecat cupidatat non  
proident, sunt in culpa qui officia  
deserunt mollit anim id est laborum

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width`
- `baseline-skip`

`\left-align` *arg* (markup)

Align *arg* on its left edge.

```
\markup {
  \column {
    one
    \left-align
    two
    three
  }
}
```

one  
two  
three

`\left-column` *args* (markup list)

Put *args* in a left-aligned column.

```
\markup {
  \left-column {
    one
    two
    three
  }
}
```

one  
two  
three

Used properties:

- `baseline-skip`

`\line` *args* (markup list)

Put *args* in a horizontal line. The property `word-space` determines the space between markups in *args*.

```
\markup {
  \line {
    one two three
  }
}
```

one two three

Used properties:

- `text-direction` (1)
- `word-space`

`\lower` *amount* (number) *arg* (markup)

Lower *arg* by the distance *amount*. A negative *amount* indicates raising; see also `\raise`.

```
\markup {
  one
  \lower #3
  two
  three
}
```

one      three  
two

`\overlay` *args* (markup list)

Takes a list of markups combining them.

```
\markup {
  \fontsize #5
  \override #'(thickness . 2)
  \overlay {
    \draw-line #'(0 . 4)
    \arrow-head #Y #DOWN ##f
    \translate #'(0 . 4)\arrow-head #Y #UP ##f
  }
}
```



`\pad-around` *amount* (number) *arg* (markup)

Add padding *amount* all around *arg*.

```
\markup {
  \box {
    default
  }
  \hspace #2
  \box {
    \pad-around #0.5 {
      padded
    }
  }
}
```

```

    }
  }
}



default



padded


```

`\pad-markup` *amount* (number) *arg* (markup)

Add space around a markup object. Identical to `pad-around`.

```

\markup {
  \box {
    default
  }
  \hspace #2
  \box {
    \pad-markup #1 {
      padded
    }
  }
}

```

default

padded

`\pad-to-box` *x-ext* (pair of numbers) *y-ext* (pair of numbers) *arg* (markup)

Make *arg* take at least *x-ext*, *y-ext* space.

```

\markup {
  \box {
    default
  }
  \hspace #4
  \box {
    \pad-to-box #'(0 . 10) #'(0 . 3) {
      padded
    }
  }
}

```

default

padded

`\pad-x` *amount* (number) *arg* (markup)

Add padding *amount* around *arg* in the X direction.

```

\markup {
  \box {
    default
  }
  \hspace #4
  \box {
    \pad-x #2 {
      padded
    }
  }
}

```

}

**default**

**padded**

`\put-adjacent` *axis* (integer) *dir* (direction) *arg1* (markup) *arg2* (markup)

Put *arg2* next to *arg1*, without moving *arg1*.

`\raise` *amount* (number) *arg* (markup)

Raise *arg* by the distance *amount*. A negative *amount* indicates lowering, see also `\lower`.

The argument to `\raise` is the vertical displacement amount, measured in (global) staff spaces. `\raise` and `\super` raise objects in relation to their surrounding markups.

If the text object itself is positioned above or below the staff, then `\raise` cannot be used to move it, since the mechanism that positions it next to the staff cancels any shift made with `\raise`. For vertical positioning, use the `padding` and/or `extra-offset` properties.

```
\markup {
  C
  \small
  \bold
  \raise #1.0
  9/7+
}
```

**C 9/7+**

`\right-align` *arg* (markup)

Align *arg* on its right edge.

```
\markup {
  \column {
    one
    \right-align
    two
    three
  }
}
```

one  
two  
three

`\right-column` *args* (markup list)

Put *args* in a right-aligned column.

```
\markup {
  \right-column {
    one
    two
    three
  }
}
```

```

    }

    one
    two
    three

```

Used properties:

- `baseline-skip`

`\rotate` *ang* (number) *arg* (markup)

Rotate object with *ang* degrees around its center.

```

\markup {
  default
  \hspace #2
  \rotate #45
  \line {
    rotated 45°
  }
}

```

default *rotated 45°*

`\translate` *offset* (pair of numbers) *arg* (markup)

Translate *arg* relative to its surroundings. *offset* is a pair of numbers representing the displacement in the X and Y axis.

```

\markup {
  *
  \translate #'(2 . 3)
  \line { translated two spaces right, three up }
}

```

translated two spaces right, three up  
\*

`\translate-scaled` *offset* (pair of numbers) *arg* (markup)

Translate *arg* by *offset*, scaling the offset by the `font-size`.

```

\markup {
  \fontsize #5 {
    * \translate #'(2 . 3) translate
    \hspace #2
    * \translate-scaled #'(2 . 3) translate-scaled
  }
}

```

\* **translate** \* **translate-scaled**

Used properties:

- `font-size` (0)

`\vcenter` *arg* (markup)

Align *arg* to its Y center.

```
\markup {
  one
  \vcenter
  two
  three
}
```

one two three

`\vspace` *amount* (number)

Create an invisible object taking up vertical space of *amount* multiplied by 3.

```
\markup {
  \center-column {
    one
    \vspace #2
    two
    \vspace #5
    three
  }
}
```

one

two

three

`\wordwrap-field` *symbol* (symbol)

Wordwrap the data which has been assigned to *symbol*.

```
\header {
  title = "My title"
  myText = "Lorem ipsum dolor sit amet, consectetur
  adipisicing elit, sed do eiusmod tempor incididunt ut
  labore et dolore magna aliqua. Ut enim ad minim
  veniam, quis nostrud exercitation ullamco laboris nisi
  ut aliquip ex ea commodo consequat."
}

\paper {
  bookTitleMarkup = \markup {
    \column {
      \fill-line { \fromproperty #'header:title }
      \null
      \wordwrap-field #'header:myText
    }
  }
```



```

    }
  }
}

\markup {
  \null
}

```

My title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

`\wordwrap` *args* (markup list)

Simple wordwrap. Use `\override #'(line-width . X)` to set the line width, where *X* is the number of staff spaces.

```

\markup {
  \wordwrap {
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore
    magna aliqua. Ut enim ad minim veniam, quis nostrud
    exercitation ullamco laboris nisi ut aliquip ex ea
    commodo consequat.
  }
}

```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\wordwrap-string` *arg* (string)

Wordwrap a string. Paragraphs may be separated with double newlines.

```

\markup {
  \override #'(line-width . 40)
  \wordwrap-string #"Lorem ipsum dolor sit amet,
    consectetur adipisicing elit, sed do eiusmod tempor
    incididunt ut labore et dolore magna aliqua.

```

```

    Ut enim ad minim veniam, quis nostrud exercitation
    ullamco laboris nisi ut aliquip ex ea commodo
    consequat.

```

```

    Excepteur sint occaecat cupidatat non proident,
    sunt in culpa qui officia deserunt mollit anim id
    est laborum"
}

```

```

    Lorem ipsum dolor sit amet,
    consectetur adipisicing elit, sed do
    eiusmod tempor incididunt ut labore et
    dolore magna aliqua.
    Ut enim ad minim veniam, quis
    nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo
    consequat.
    Excepteur sint occaecat cupidatat non
    proident, sunt in culpa qui officia
    deserunt mollit anim id est laborum

```

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width`
- `baseline-skip`

### A.11.3 Graphic

`\arrow-head axis (integer) dir (direction) filled (boolean)`

Produce an arrow head in specified direction and axis. Use the filled head if *filled* is specified.

```

\markup {
  \fontsize #5 {
    \general-align #Y #DOWN {
      \arrow-head #Y #UP ##t
      \arrow-head #Y #DOWN ##f
      \hspace #2
      \arrow-head #X #RIGHT ##f
      \arrow-head #X #LEFT ##f
    }
  }
}

```

▲Υ ><

`\beam width (number) slope (number) thickness (number)`

Create a beam with the specified parameters.

```

\markup {
  \beam #5 #1 #2

```

}



`\bracket arg (markup)`

Draw vertical brackets around *arg*.

```
\markup {
  \bracket {
    \note {2.} #UP
  }
}
```



`\circle arg (markup)`

Draw a circle around *arg*. Use **thickness**, **circle-padding** and **font-size** properties to determine line thickness and padding around the markup.

```
\markup {
  \circle {
    Hi
  }
}
```



Used properties:

- **circle-padding** (0.2)
- **font-size** (0)
- **thickness** (1)

`\draw-circle radius (number) thickness (number) filled (boolean)`

A circle of radius *radius* and thickness *thickness*, optionally filled.

```
\markup {
  \draw-circle #2 #0.5 ##f
  \hspace #2
  \draw-circle #2 #0 ##t
}
```



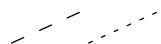
`\draw-dashed-line dest (pair of numbers)`

A dashed line.

If **full-length** is set to **#t** (default) the dashed-line extends to the whole length given by *dest*, without white space at beginning or end. **off** will then be altered to fit. To insist on the given (or default) values of **on**, **off** use **\override #'(full-length . #f)** Manual settings for **on**, **off** and **phase** are possible.

```
\markup {
  \draw-dashed-line #'(5.1 . 2.3)
  \override #'((on . 0.3) (off . 0.5))
}
```

```
\draw-dashed-line #'(5.1 . 2.3)
}
```



Used properties:

- `full-length` (`#t`)
- `phase` (0)
- `off` (1)
- `on` (1)
- `thickness` (1)

`\draw-dotted-line` *dest* (pair of numbers)

A dotted line.

The dotted-line always extends to the whole length given by *dest*, without white space at beginning or end. Manual settings for `off` are possible to get larger or smaller space between the dots. The given (or default) value of `off` will be altered to fit the line-length.

```
\markup {
  \draw-dotted-line #'(5.1 . 2.3)
  \override #'((thickness . 2) (off . 0.2))
  \draw-dotted-line #'(5.1 . 2.3)
}
```



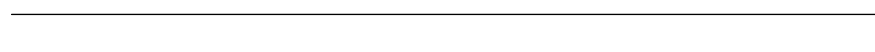
Used properties:

- `phase` (0)
- `off` (1)
- `thickness` (1)

`\draw-hline`

Draws a line across a page, where the property `span-factor` controls what fraction of the page is taken up.

```
\markup {
  \column {
    \draw-hline
    \override #'(span-factor . 1/3)
    \draw-hline
  }
}
```



Used properties:

- `span-factor` (1)
- `line-width`
- `draw-line-markup`

`\draw-line` *dest* (pair of numbers)

A simple line.

```
\markup {
  \draw-line #'(4 . 4)
  \override #'(thickness . 5)
  \draw-line #'(-3 . 0)
}
```



Used properties:

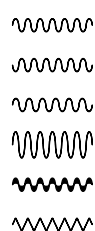
- `thickness` (1)

`\draw-squiggle-line` *sq-length* (number) *dest* (pair of numbers) *eq-end?* (boolean)

A squiggled line.

If `eq-end?` is set to `#t`, it is ensured the squiggled line ends with a bow in same direction as the starting one. `sq-length` is the length of the first bow. `dest` is the end point of the squiggled line. To match `dest` the squiggled line is scaled accordingly. Its appearance may be customized by overrides for `thickness`, `angularity`, `height` and `orientation`.

```
\markup
\column {
  \draw-squiggle-line #0.5 #'(6 . 0) ##t
  \override #'(orientation . -1)
  \draw-squiggle-line #0.5 #'(6 . 0) ##t
  \draw-squiggle-line #0.5 #'(6 . 0) ##f
  \override #'(height . 1)
  \draw-squiggle-line #0.5 #'(6 . 0) ##t
  \override #'(thickness . 5)
  \draw-squiggle-line #0.5 #'(6 . 0) ##t
  \override #'(angularity . 2)
  \draw-squiggle-line #0.5 #'(6 . 0) ##t
}
```



Used properties:

- `orientation` (1)
- `height` (0.5)
- `angularity` (0)
- `thickness` (0.5)

`\ellipse` *arg* (markup)

Draw an ellipse around *arg*. Use `thickness`, `x-padding`, `y-padding` and `font-size` properties to determine line thickness and padding around the markup.

```
\markup {
```

```
\ellipse {
  Hi
}
```



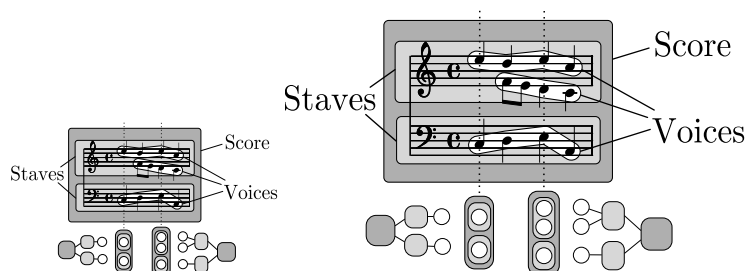
Used properties:

- `y-padding` (0.2)
- `x-padding` (0.2)
- `font-size` (0)
- `thickness` (1)

`\epsfile axis (number) size (number) file-name (string)`

Inline an EPS image. The image is scaled along *axis* to *size*.

```
\markup {
  \general-align #Y #DOWN {
    \epsfile #X #20 #"context-example.eps"
    \epsfile #Y #20 #"context-example.eps"
  }
}
```



`\filled-box xext (pair of numbers) yext (pair of numbers) blot (number)`

Draw a box with rounded corners of dimensions *xext* and *yext*. For example,

```
\filled-box #'(-.3 . 1.8) #'(-.3 . 1.8) #0
```

creates a box extending horizontally from -0.3 to 1.8 and vertically from -0.3 up to 1.8, with corners formed from a circle of diameter 0 (i.e., sharp corners).

```
\markup {
  \filled-box #'(0 . 4) #'(0 . 4) #0
  \filled-box #'(0 . 2) #'(-4 . 2) #0.4
  \combine
  \filled-box #'(1 . 8) #'(0 . 7) #0.2
  \with-color #white
  \filled-box #'(3.6 . 5.6) #'(3.5 . 5.5) #0.7
}
```



`\hbracket arg (markup)`

Draw horizontal brackets around *arg*.

```
\markup {
```

```
\hbracket {
  \line {
    one two three
  }
}
```

one two three

`\oval arg` (markup)

Draw an oval around *arg*. Use `thickness`, `x-padding`, `y-padding` and `font-size` properties to determine line thickness and padding around the markup.

```
\markup {
  \oval {
    Hi
  }
}
```



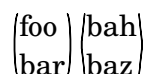
Used properties:

- `y-padding` (0.75)
- `x-padding` (0.75)
- `font-size` (0)
- `thickness` (1)

`\parenthesize arg` (markup)

Draw parentheses around *arg*. This is useful for parenthesizing a column containing several lines of text.

```
\markup {
  \parenthesize
  \column {
    foo
    bar
  }
  \override #'(angularity . 2)
  \parenthesize
  \column {
    bah
    baz
  }
}
```



Used properties:

- `width` (0.25)
- `line-thickness` (0.1)
- `thickness` (1)
- `size` (1)

- `padding`
- `angularity` (0)

`\path` *thickness* (number) *commands* (list)

Draws a path with line *thickness* according to the directions given in *commands*. *commands* is a list of lists where the `car` of each sublist is a drawing command and the `cdr` comprises the associated arguments for each command.

There are seven commands available to use in the list `commands`: `moveto`, `rmoveto`, `lineto`, `rlineto`, `curveto`, `rcurveto`, and `closepath`. Note that the commands that begin with *r* are the relative variants of the other three commands.

The commands `moveto`, `rmoveto`, `lineto`, and `rlineto` take 2 arguments; they are the X and Y coordinates for the destination point.

The commands `curveto` and `rcurveto` create cubic Bézier curves, and take 6 arguments; the first two are the X and Y coordinates for the first control point, the second two are the X and Y coordinates for the second control point, and the last two are the X and Y coordinates for the destination point.

The `closepath` command takes zero arguments and closes the current subpath in the active path.

Note that a sequence of commands *must* begin with a `moveto` or `rmoveto` to work with the SVG output.

Line-cap styles and line-join styles may be customized by overriding the `line-cap-style` and `line-join-style` properties, respectively. Available line-cap styles are `'butt`, `'round`, and `'square`. Available line-join styles are `'miter`, `'round`, and `'bevel`.

The property `filled` specifies whether or not the path is filled with color.

```
samplePath =
  #'((moveto 0 0)
    (lineto -1 1)
    (lineto 1 1)
    (lineto 1 -1)
    (curveto -5 -5 -5 5 -1 0)
    (closepath))

\markup {
  \path #0.25 #samplePath

  \override #'(line-join-style . miter)
  \path #0.25 #samplePath

  \override #'(filled . #t)
  \path #0.25 #samplePath
}
```



Used properties:

- `filled` (`#f`)
- `line-join-style` (`round`)
- `line-cap-style` (`round`)



`\polygon` *points* (list of number pairs)

A polygon delimited by the list of *points*. *extroversion* defines how the shape of the polygon is adapted to its thickness. If it is 0, the polygon is traced as-is. If -1, the outer side of the line is just on the given points. If 1, the line has its inner side on the points. The *thickness* property controls the thickness of the line; for filled polygons, this means the diameter of the blot.

```
regularPentagon =
#'((1 . 0) (0.31 . 0.95) (-0.81 . 0.59) (-0.81 . -0.59) (0.31 . -0.95))

\markup {
  \polygon #'((-1 . -1) (0 . -3) (2 . 2) (1 . 2))
  \override #'(filled . #f)
  \override #'(thickness . 2)
  \combine
    \with-color "blue"
    \polygon #regularPentagon
    \with-color "red"
    \override #'(extroversion . 1)
    \polygon #regularPentagon
}
```



Used properties:

- *thickness* (1)
- *filled* (#t)
- *extroversion* (0)

`\postscript` *str* (string)

This inserts *str* directly into the output as a PostScript command string.

```
ringsps = #"
  0.15 setlinewidth
  0.9 0.6 moveto
  0.4 0.6 0.5 0 361 arc
  stroke
  1.0 0.6 0.5 0 361 arc
  stroke
  "

rings = \markup {
  \with-dimensions #'(-0.2 . 1.6) #'(0 . 1.2)
  \postscript #ringsps
}

\relative c'' {
  c2^\rings
  a2_\rings
}
```

}



`\rounded-box` *arg* (markup)

Draw a box with rounded corners around *arg*. Looks at **thickness**, **box-padding** and **font-size** properties to determine line thickness and padding around the markup; the **corner-radius** property makes it possible to define another shape for the corners (default is 1).

```
c4~\markup {
  \rounded-box {
    Overtura
  }
}
c,8. c16 c4 r
```



Used properties:

- **box-padding** (0.5)
- **font-size** (0)
- **corner-radius** (1)
- **thickness** (1)

`\scale` *factor-pair* (pair of numbers) *arg* (markup)

Scale *arg*. *factor-pair* is a pair of numbers representing the scaling-factor in the X and Y axes. Negative values may be used to produce mirror images.

```
\markup {
  \line {
    \scale #'(2 . 1)
    stretched
    \scale #'(1 . -1)
    mirrored
  }
}
```

**stretched** **mirrored**

`\triangle` *filled* (boolean)

A triangle, either filled or empty.

```
\markup {
  \triangle ##t
  \hspace #2
  \triangle ##f
}
```



Used properties:

- `thickness` (1)
- `font-size` (0)
- `extroversion` (0)

`\with-url` *url* (string) *arg* (markup)

Add a link to URL *url* around *arg*. This only works in the PDF backend.

```
\markup {
  \with-url #"https://lilypond.org/" {
    LilyPond ... \italic {
      music notation for everyone
    }
  }
}
```

LilyPond ... *music notation for everyone*

#### A.11.4 Music

`\accidental` *alteration* (an exact rational number)

Select an accidental glyph from an alteration, given as rational number.

```
\markup \accidental #1/2
```

#

Used properties:

- `alteration-glyph-name-alist`

`\compound-meter` *time-sig* (number or pair)

Draw a numeric time signature.

```
\markup {
  \column {
    \line { Single number:
      \compound-meter #3 }
    \line { Conventional:
      \compound-meter #'(4 . 4) or
      \compound-meter #'(4 4) }
    \line { Compound:
      \compound-meter #'(2 3 8) }
    \line { Single-number compound:
      \compound-meter #'((2) (3)) }
    \line { Complex compound:
      \compound-meter #'((2 3 8) (3 4)) }
  }
}
```

Single number: **3**

Conventional:  $\frac{4}{4}$  or  $\frac{4}{4}$

Compound:  $2 + \frac{3}{8}$

Single-number compound:  $2 + 3$

Complex compound:  $2 + \frac{3}{8} + \frac{3}{4}$

`\customTabClef num-strings (integer) staff-space (number)`  
 Draw a tab clef sans-serif style.

`\doubleflat`  
 Draw a double flat symbol.

```
\markup {
  \doubleflat
}
```



`\doublesharp`  
 Draw a double sharp symbol.

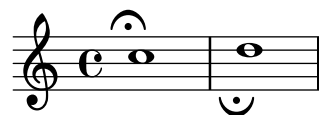
```
\markup {
  \doublesharp
}
```



`\fermata` Create a fermata glyph. When *direction* is `DOWN`, use an inverted glyph. Note that within music, one would usually use the `\fermata` articulation instead of a markup.

```
{ c''1^\markup \fermata d''1_\markup \fermata }
```

```
\markup { \fermata \override #`(direction . ,DOWN) \fermata }
```



Used properties:

- `direction` (1)

`\flat` Draw a flat symbol.

```
\markup {
  \flat
}
```



`\multi-measure-rest-by-number duration-scale (non-negative integer)`

Returns a multi-measure rest symbol.

If the number of measures is greater than the number given by `expand-limit` a horizontal line is printed. For every multi-measure rest lasting more than one measure a number is printed on top.

```
\markup {
  Multi-measure rests may look like
  \multi-measure-rest-by-number #12
  or
  \multi-measure-rest-by-number #7
  (church rests)
```

}

Multi-measure rests may look like  or  (church rests)

Used properties:

- `multi-measure-rest-number` (#t)
- `width` (8)
- `expand-limit` (10)
- `hair-thickness` (2.0)
- `thick-thickness` (6.6)
- `word-space`
- `style` (())
- `font-size` (0)

`\musicglyph` *glyph-name* (string)

*glyph-name* is converted to a musical symbol; for example, `\musicglyph #"accidentals.natural"` selects the natural sign from the music font. See Sección “The Emmentaler font” en *Referencia de la Notación* for a complete listing of the possible glyphs.

```
\markup {
  \musicglyph #"f"
  \musicglyph #"rests.2"
  \musicglyph #"clefs.G_change"
}
```



`\natural` Draw a natural symbol.

```
\markup {
  \natural
}
```



`\note-by-number` *log* (number) *dot-count* (number) *dir* (number)

Construct a note symbol, with stem and flag. By using fractional values for *dir*, longer or shorter stems can be obtained. Supports all note-head-styles. Ancient note-head-styles will get mensural-style-flags. `flag-style` may be overridden independently. Supported flag-styles are `default`, `old-straight-flag`, `modern-straight-flag`, `flat-flag`, `mensural` and `neomensural`. The latter two flag-styles will both result in mensural-flags. Both are supplied for convenience.

```
\markup {
  \note-by-number #3 #0 #DOWN
  \hspace #2
  \note-by-number #1 #2 #0.8
}
```



Used properties:

- `style (())`
- `flag-style (())`
- `font-size (0)`

`\note duration (duration) dir (number)`

This produces a note with a stem pointing in *dir* direction, with the *duration* for the note head type and augmentation dots. For example, `\note {4.} #-0.75` creates a dotted quarter note, with a shortened down stem.

```
\markup {
  \override #'(style . cross)
  \note {4..} #UP
  \hspace #2
  \note {\breve} #0
}
```



Used properties:

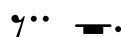
- `style (())`
- `flag-style (())`
- `font-size (0)`

`\rest-by-number log (integer) dot-count (integer)`

A rest symbol.

For duration logs specified with property `ledgers`, rest symbols with ledger lines are selected.

```
\markup {
  \rest-by-number #3 #2
  \hspace #2
  \rest-by-number #0 #1
}
```



Used properties:

- `style (())`
- `ledgers ((-1 0 1))`
- `font-size (0)`

`\rest duration (duration)`

Returns a rest symbol.




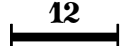
If `multi-measure-rest` is set to true, a multi-measure rest symbol may be returned. In this case the duration needs to be entered as `{ 1*2 }` to get a multi-measure rest for two bars. Actually, it's only the scaling factor that determines the length, the basic duration is disregarded.

```
\markup {
  Rests:
  \hspace #2
  \rest { 4.. }
  \hspace #2
}
```

```

\rest { \breve }
\hspace #2
Multi-measure rests:
\override #'(multi-measure-rest . #t)
{
\hspace #2
\override #'(multi-measure-rest-number . #f)
\rest { 1*7 }
\hspace #2
\rest { 1*12 }
}
}

```

Rests:   Multi-measure rests:  

Used properties:

- multi-measure-rest-number (#t)
- width (8)
- expand-limit (10)
- hair-thickness (2.0)
- thick-thickness (6.6)
- word-space
- style (())
- font-size (0)
- style (())
- ledgers ((-1 0 1))
- font-size (0)

`\score score (score)`

Inline an image of music. The reference point (usually the middle staff line) of the lowest staff in the top system is placed on the baseline.

```

\markup {
\score {
\new PianoStaff <<
\new Staff \relative c' {
\key f \major
\time 3/4
\mark \markup { Allegro }
f2\p( a4)
c2( a4)
bes2( g'4)
f8( e) e4 r
}
\new Staff \relative c {
\clef bass
\key f \major
\time 3/4
f8( a c a c a
f c' es c es c)
f,( bes d bes d bes)
}
}

```

```

        f( g bes g bes g)
    }
>>
\layout {
  indent = 0.0\cm
  \context {
    \Score
    \override RehearsalMark.break-align-symbols =
      #'(time-signature key-signature)
    \override RehearsalMark.self-alignment-X = #LEFT
  }
  \context {
    \Staff
    \override TimeSignature
      .break-align-anchor-alignment = #LEFT
  }
}
}
}

```



Used properties:

- `baseline-skip`

`\semiflat`

Draw a semiflat symbol.

```

\markup {
  \semiflat
}

```



`\semisharp`

Draw a semisharp symbol.

```

\markup {
  \semisharp
}

```



`\sesquiflat`

Draw a 3/2 flat symbol.

```

\markup {
  \sesquiflat
}

```





### `\sesquisharp`

Draw a 3/2 sharp symbol.

```
\markup {
  \sesquisharp
}
```

##

### `\sharp`

Draw a sharp symbol.

```
\markup {
  \sharp
}
```

#

### `\tied-lyric` *str* (string)

Like simple-markup, but use tie characters for ‘~’ tilde symbols.

```
\markup \column {
  \tied-lyric
    #"Siam navi~all'onde~argenti Lasciate~in abbandono"
  \tied-lyric
    #"Impetuosi venti I nostri~affetti sono"
  \tied-lyric
    #"Ogni diletto~e scoglio Tutta la vita~e~un mar."
}
```

Siam navi~all'onde~argenti Lasciate~in abbandono  
 Impetuosi venti I nostri~affetti sono  
 Ogni diletto~e scoglio Tutta la vita~e~un mar.

Used properties:

- `word-space`

## A.11.5 Instrument Specific Markup

### `\fret-diagram` *definition-string* (string)

Make a (guitar) fret diagram. For example, say

```
\markup \fret-diagram #"s:0.75;6-x;5-x;4-o;3-2;2-3;1-2;"
```

for fret spacing 3/4 of staff space, D chord diagram

Syntax rules for *definition-string*:

- Diagram items are separated by semicolons.
- Possible items:
  - `s: number` – Set the fret spacing of the diagram (in staff spaces). Default: 1.
  - `t: number` – Set the line thickness (relative to normal line thickness). Default: 0.5.
  - `h: number` – Set the height of the diagram in frets. Default: 4.
  - `w: number` – Set the width of the diagram in strings. Default: 6.
  - `f: number` – Set fingering label type (0 = none, 1 = in circle on string, 2 = below string). Default: 0.
  - `d: number` – Set radius of dot, in terms of fret spacing. Default: 0.25.

- `p:number` – Set the position of the dot in the fret space. 0.5 is centered; 1 is on lower fret bar, 0 is on upper fret bar. Default: 0.6.
  - `c:string1-string2-fret` – Include a barre mark from *string1* to *string2* on *fret*.
  - `string-fret` – Place a dot on *string* at *fret*. If *fret* is ‘o’, *string* is identified as open. If *fret* is ‘x’, *string* is identified as muted.
  - `string-fret-fingering` – Place a dot on *string* at *fret*, and label with *fingering* as defined by the `f:` code.
- Note: There is no limit to the number of fret indications per string.

Used properties:

- `thickness` (0.5)
- `fret-diagram-details`
- `size` (1.0)
- `align-dir` (-0.4)

`\fret-diagram-terse` *definition-string* (string)

Make a fret diagram markup using terse string-based syntax.

Here is an example

```
\markup \fret-diagram-terse #"x;x;o;2;3;2;"
```

for a D chord diagram.

Syntax rules for *definition-string*:

- Strings are terminated by semicolons; the number of semicolons is the number of strings in the diagram.
- Mute strings are indicated by ‘x’.
- Open strings are indicated by ‘o’.
- A number indicates a fret indication at that fret.
- If there are multiple fret indicators desired on a string, they should be separated by spaces.
- Fingerings are given by following the fret number with a -, followed by the finger indicator, e.g. ‘3-2’ for playing the third fret with the second finger.
- Where a barre indicator is desired, follow the fret (or fingering) symbol with -( to start a barre and -) to end the barre.

Used properties:

- `thickness` (0.5)
- `fret-diagram-details`
- `size` (1.0)
- `align-dir` (-0.4)

`\fret-diagram-verbose` *marking-list* (pair)

Make a fret diagram containing the symbols indicated in *marking-list*.

For example,

```
\markup \fret-diagram-verbose
  #'((mute 6) (mute 5) (open 4)
    (place-fret 3 2) (place-fret 2 3) (place-fret 1 2))
```

produces a standard D chord diagram without fingering indications.

Possible elements in *marking-list*:

(mute *string-number*)

Place a small ‘x’ at the top of string *string-number*.

(open *string-number*)

Place a small ‘o’ at the top of string *string-number*.

(barre *start-string end-string fret-number*)

Place a barre indicator (much like a tie) from string *start-string* to string *end-string* at fret *fret-number*.

(capo *fret-number*)

Place a capo indicator (a large solid bar) across the entire fretboard at fret location *fret-number*. Also, set fret *fret-number* to be the lowest fret on the fret diagram.

(place-fret *string-number fret-number [finger-value] [color-modifier] [color] ['parenthesized ['default-paren-color]]*)

Place a fret playing indication on string *string-number* at fret *fret-number* with an optional fingering label *finger-value*, an optional color modifier *color-modifier*, an optional color *color*, an optional parenthesis *'parenthesized* and an optional parenthesis color *'default-paren-color*. By default, the fret playing indicator is a solid dot. This can be globally changed by setting the value of the variable *dot-color* or for a single dot by setting the value of *color*. The dot can be parenthesized by adding *'parenthesized*. By default the color for the parenthesis is taken from the dot. Adding *'default-paren-color* will take the parenthesis-color from the global *dot-color*, as a fall-back black will be used. Setting *color-modifier* to *inverted* inverts the dot color for a specific fingering. The values for *string-number*, *fret-number*, and the optional *finger* should be entered first in that order. The order of the other optional arguments does not matter. If the *finger* part of the *place-fret* element is present, *finger-value* will be displayed according to the setting of the variable *finger-code*. There is no limit to the number of fret indications per string.

Used properties:

- *thickness* (0.5)
- *fret-diagram-details*
- *size* (1.0)
- *align-dir* (-0.4)

`\harp-pedal` *definition-string* (string)

Make a harp pedal diagram.

Possible elements in *definition-string*:

- |   |   |
|---|---|
| ^ | pedal is up   |
| - | pedal is neutral  |
| v | pedal is down   |
|   | vertical divider line                                       |
| o | the following pedal should be circled (indicating a change) |

The function also checks if the string has the typical form of three pedals, then the divider and then the remaining four pedals. If not it prints out a warning. However, in any case, it will also print each symbol in the order as given. This means you can place the divider (even multiple dividers) anywhere you want, but you'll have to live with the warnings.

The appearance of the diagram can be tweaked inter alia using the size property of the TextScript grob (`\override Voice.TextScript.size = #0.3`) for the overall, the thickness property (`\override Voice.TextScript.thickness = #3`) for the line thickness of the horizontal line and the divider. The remaining configuration (box sizes, offsets and spaces) is done by the harp-pedal-details list of properties (`\override Voice.TextScript.harp-pedal-details.box-width = #1`). It contains the following settings: `box-offset` (vertical shift of the box center for up/down pedals), `box-width`, `box-height`, `space-before-divider` (the spacing between two boxes before the divider) and `space-after-divider` (box spacing after the divider).

```
\markup \harp-pedal #"^-v|--ov^"
```



Used properties:

- `thickness` (0.5)
- `harp-pedal-details` (())
- `size` (1.2)

`\woodwind-diagram` *instrument* (symbol) *user-draw-commands* (list)

Make a woodwind-instrument diagram. For example, say

```
\markup \woodwind-diagram
      #'oboe #'((lh . (d ees)) (cc . (five3qT1q)) (rh . (gis)))
```

for an oboe with the left-hand d key, left-hand ees key, and right-hand gis key depressed while the five-hole of the central column effectuates a trill between 1/4 and 3/4 closed.

The following instruments are supported:

- piccolo
- flute
- oboe
- clarinet
- bass-clarinet
- saxophone
- bassoon
- contrabassoon

To see all of the callable keys for a given instrument, include the function (`print-keys 'instrument`) in your .ly file, where *instrument* is the instrument whose keys you want to print.

Certain keys allow for special configurations. The entire gamut of configurations possible is as follows:

- 1q (1/4 covered)
- 1h (1/2 covered)

- 3q (3/4 covered)
- R (ring depressed)
- F (fully covered; the default if no state put)

Additionally, these configurations can be used in trills. So, for example, `three3qTR` effectuates a trill between 3/4 full and ring depressed on the three hole. As another example, `threeRT` effectuates a trill between R and open, whereas `threeTR` effectuates a trill between open and shut. To see all of the possibilities for all of the keys of a given instrument, invoke `(print-keys-verbose 'instrument)`.

Lastly, substituting an empty list for the pressed-key alist will result in a diagram with all of the keys drawn but none filled, for example:

```
\markup \woodwind-diagram #'oboe #'()
```

Used properties:

- `graphical` (`#t`)
- `thickness` (`0.1`)
- `size` (`1`)

## A.11.6 Accordion Registers

`\discant` *name* (string)

`\discant` *name* generates a discant accordion register symbol.

To make it available,





```
 #(use-modules (lily accreg))
```

is required near the top of your input file.

The register names in the default `\discant` register set have modeled after numeric Swiss notation like depicted in [http://de.wikipedia.org/wiki/Register\\_%28Akkordeon%29](http://de.wikipedia.org/wiki/Register_%28Akkordeon%29), omitting the slashes and dropping leading zeros.

The string *name* is basically a three-digit number with the lowest digit specifying the number of 16' reeds, the tens the number of 8' reeds, and the hundreds specifying the number of 4' reeds. Without modification, the specified number of reeds in 8' is centered in the symbol. Newer instruments may have registrations where 8' can be used either within or without a tone chamber, 'cassotto'. Notationally, the central dot then indicates use of cassotto. One can suffix the tens' digits '1' and '2' with '+' or '-' to indicate clustering the dots at the right or left respectively rather than centered.

Some examples are

	
<code>\discant "1"</code>	<code>\discant "1+0"</code>
	
<code>\discant "120"</code>	<code>\discant "131"</code>

Used properties:

- `font-size` (`0`)

`\freeBass` *name* (string)




`\freeBass` *name* generates a free bass/converter accordion register symbol for the usual two-reed layout.

To make it available,

```
#(use-modules (lily accreg))
```

is required near the top of your input file.

Available registrations are


  
`\freeBass "1"`     `\freeBass "11"`  
  
  
`\freeBass "10"`

Used properties:

- `font-size` (0)

`\stdBass` *name* (string)

`\stdBass` *name* generates a standard bass accordion register symbol.

To make it available,

```
#(use-modules (lily accreg))
```

is required near the top of your input file.

The default bass register definitions have been modeled after the article <http://www.accordions.com/index/art/stradella.shtml> originally appearing in Accord Magazine.

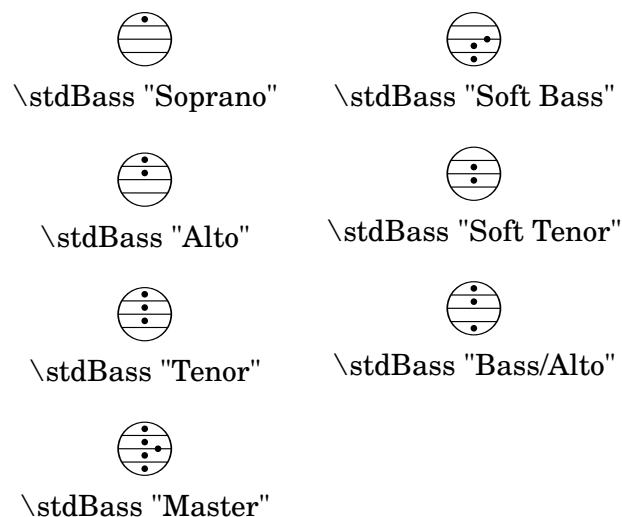
The underlying register model is



This kind of overlapping arrangement is common for Italian instruments though the exact location of the octave breaks differ.

When not composing for a particular target instrument, using the five reed definitions makes more sense than using a four reed layout: in that manner, the ‘**Master**’ register is unambiguous. This is rather the rule in literature bothering about bass registrations at all.

Available registrations are



Used properties:

- `font-size` (0)

`\stdBassIV` *name* (string)

`\stdBassIV` *name* generates a standard bass accordion register symbol.

To make it available,

`\#(use-modules (lily accreg))`

is required near the top of your input file.









The main use is for four-reed standard bass instruments with reedbank layout



Notable instruments are Morino models with MIII (the others are five-reed instead) and the Atlantic IV. Most of those models have three register switches. Some newer Morinos with MIII might have five or even seven.

The prevalent three-register layout uses the middle three switches ‘**Tenor**’, ‘**Master**’, ‘**Soft Bass**’. Note that the sound is quite darker than the same registrations of ‘**c**,’-based instruments.

Available registrations are

	
<code>\stdBassIV "Soprano"</code>	<code>\stdBassIV "Soft Bass"</code>
	
<code>\stdBassIV "Alto"</code>	<code>\stdBassIV "Bass/Alto"</code>
	
<code>\stdBassIV "Tenor"</code>	<code>\stdBassIV "Soft Bass/Alto"</code>
	
<code>\stdBassIV "Master"</code>	<code>\stdBassIV "Soft Tenor"</code>

Used properties:

- `font-size (0)`

`\stdBassV` *name* (string)

`\stdBassV` *name* generates a standard bass accordion register symbol.

To make it available,

`#(use-modules (lily accreg))`

is required near the top of your input file.

The main use is for five-reed standard bass instruments with reedbank layout













This tends to be the bass layout for Hohner's Morino series without convertor or MIII manual.

With the exception of the rather new 7-register layout, the highest two chord reeds are usually sounded together. The Older instruments offer 5 or 3 bass registers. The Tango VM offers an additional 'Solo Bass' setting that mutes the chord reeds. The symbol on the register buttons of the Tango VM would actually match the physical five-octave layout reflected here, but it is not used in literature.

Composers should likely prefer the five-reed versions of these symbols. The mismatch of a four-reed instrument with five-reed symbols is easier to resolve for the player than the other way round.



Available registrations are

	
<code>\stdBassV "Bass/Alto"</code>	<code>\stdBassV "Soft Bass"</code>
	
<code>\stdBassV "Soft Bass/Alto"</code>	<code>\stdBassV "Soft Tenor"</code>
	
<code>\stdBassV "Alto"</code>	<code>\stdBassV "Soprano"</code>
	
<code>\stdBassV "Tenor"</code>	<code>\stdBassV "Sopranos"</code>
	
<code>\stdBassV "Master"</code>	<code>\stdBassV "Solo Bass"</code>

Used properties:

- `font-size (0)`

`\stdBassVI` *name* (string)

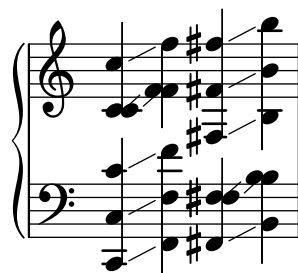
`\stdBassVI` *name* generates a standard bass accordion register symbol for six reed basses.

To make it available,

```
#(use-modules (lily accreg))
```








is required near the top of your input file.

This is primarily the register layout for the Hohner “Gola” model. The layout is



The registers are effectively quite similar to that of `\stdBass`. An additional bass reed at alto pitch is omitted for esthetical reasons from the ‘Master’ setting, so the symbols are almost the same except for the ‘Alto/Soprano’ register with bass notes at Alto pitch and chords at Soprano pitch.

Available registrations are

	
<code>\stdBassVI "Soprano"</code>	<code>\stdBassVI "Alto/Soprano"</code>
	
<code>\stdBassVI "Alto"</code>	<code>\stdBassVI "Bass/Alto"</code>
	
<code>\stdBassVI "Soft Tenor"</code>	<code>\stdBassVI "Soft Bass"</code>
	
<code>\stdBassVI "Master"</code>	

Used properties:

- `font-size` (0)

### A.11.7 Other

`\auto-footnote` *mkup* (markup) *note* (markup)

Have footnote *note* act as an annotation to the markup *mkup*.

```
\markup {
  \auto-footnote a b
  \override #'(padding . 0.2)
  \auto-footnote c d
}
```

a c

The footnote will be annotated automatically.

Used properties:

- `padding` (0.0)
- `raise` (0.5)

`\backslashed-digit` *num* (integer)

A feta number, with backslash. This is for use in the context of figured bass notation.

```
\markup {
  \backslashed-digit #5
  \hspace #2
  \override #'(thickness . 3)
  \backslashed-digit #7
}
```

**5 7**

Used properties:

- `thickness` (1.6)
- `font-size` (0)

`\char num` (integer)

Produce a single character. Characters encoded in hexadecimal format require the prefix `#x`.

```
\markup {
  \char #65 \char ##x00a9
}
```

A ©

`\eyeglasses`

Prints out eyeglasses, indicating strongly to look at the conductor.

```
\markup { \eyeglasses }
```



`\first-visible args` (markup list)

Use the first markup in *args* that yields a non-empty stencil and ignore the rest.

```
\markup {
  \first-visible {
    \fromproperty #'header:composer
    \italic Unknown
  }
}
```

*Unknown*

`\footnote mkup` (markup) *note* (markup)

Have footnote *note* act as an annotation to the markup *mkup*.

```
\markup {
  \auto-footnote a b
  \override #'(padding . 0.2)
  \auto-footnote c d
}
```

a c

The footnote will not be annotated automatically.

`\fraction arg1` (markup) *arg2* (markup)

Make a fraction of two markups.

```
\markup {
   $\pi \approx$ 
  \fraction 355 113
}
```

$\pi \approx \frac{355}{113}$

Used properties:

- `font-size` (0)

`\fromproperty symbol` (symbol)

Read the *symbol* from property settings, and produce a stencil from the markup contained within. If *symbol* is not defined, it returns an empty markup.

```
\header {
```

```

myTitle = "myTitle"
title = \markup {
  from
  \italic
  \fromproperty #'header:myTitle
}
}
\markup {
  \null
}

```

**from *myTitle***

`\left-brace` *size* (number)

A feta brace in point size *size*.

```

\markup {
  \left-brace #35
  \hspace #2
  \left-brace #45
}

```

{ }

`\lookup` *glyph-name* (string)

Lookup a glyph by name.

```

\markup {
  \override #'(font-encoding . fetaBraces) {
    \lookup #"brace200"
    \hspace #2
    \rotate #180
    \lookup #"brace180"
  }
}

```

{ }

`\markalphabet` *num* (integer)

Make a markup letter for *num*. The letters start with A to Z and continue with double letters.

```

\markup {
  \markalphabet #8
  \hspace #2
}

```

```
\markalphabet #26
}
```

H Z

`\markletter` *num* (integer)

Make a markup letter for *num*. The letters start with A to Z (skipping letter I), and continue with double letters.

```
\markup {
  \markletter #8
  \hspace #2
  \markletter #26
}
```

H AA

`\null`

An empty markup with extents of a single point.

```
\markup {
  \null
}
```

`\on-the-fly` *procedure* (procedure) *arg* (markup)

Apply the *procedure* markup command to *arg*. *procedure* takes the same arguments as `interpret-markup` and returns a stencil.

`\override` *new-prop* (pair) *arg* (markup)

Add the argument *new-prop* to the property list. Properties may be any property supported by Sección “font-interface” en *Referencia de Funcionamiento Interno*, Sección “text-interface” en *Referencia de Funcionamiento Interno* and Sección “instrument-specific-markup-interface” en *Referencia de Funcionamiento Interno*.

*new-prop* may be either a single alist pair, or non-empty alist of its own.

```
\markup {
  \undertie "undertied"
  \override #'(offset . 15)
  \undertie "offset undertied"
  \override #'((offset . 15)(thickness . 3))
  \undertie "offset thick undertied"
}
```

undertied offset undertied offset thick undertied

`\page-link` *page-number* (number) *arg* (markup)

Add a link to the page *page-number* around *arg*. This only works in the PDF backend.

```
\markup {
  \page-link #2 { \italic { This links to page 2... } }
}
```

*This links to page 2...*

`\page-ref` *label* (symbol) *gauge* (markup) *default* (markup)

Reference to a page number. *label* is the label set on the referenced page (using `\label` or `\tocItem`), *gauge* a markup used to estimate the maximum width of the page number, and *default* the value to display when *label* is not found.

(If the current book or bookpart is set to use roman numerals for page numbers, the reference will be formatted accordingly – in which case the *gauge*'s width may require additional tweaking.)

`\pattern` *count* (non-negative integer) *axis* (non-negative integer) *space* (number) *pattern* (markup)

Prints *count* times a *pattern* markup. Patterns are spaced apart by *space* (defined as for `\hspace` or `\vspace`, respectively). Patterns are distributed on *axis*.

```
\markup \column {
  "Horizontally repeated : "
  \pattern #7 #X #2 \flat
  \null
  "Vertically repeated : "
  \pattern #3 #Y #0.5 \flat
}
```

Horizontally repeated :

b b b b b b b

Vertically repeated :

b

b

b

`\property-recursive` *symbol* (symbol)

Print out a warning when a header field markup contains some recursive markup definition.

`\right-brace` *size* (number)

A feta brace in point size *size*, rotated 180 degrees.

```
\markup {
  \right-brace #45
  \hspace #2
  \right-brace #35
}
```

} }

`\slashed-digit` *num* (integer)

A feta number, with slash. This is for use in the context of figured bass notation.

```
\markup {
  \slashed-digit #5
  \hspace #2
  \override #'(thickness . 3)
  \slashed-digit #7
}
```

}

**5 7**

Used properties:

- `thickness` (1.6)
- `font-size` (0)

`\stencil stil (stencil)`

Use a stencil as markup.

```
\markup {
  \stencil #(make-circle-stencil 2 0 #t)
}
```



`\strut`

Create a box of the same height as the space in the current font.

`\transparent arg (markup)`

Make *arg* transparent.

```
\markup {
  \transparent {
    invisible text
  }
}
```

`\verbatim-file name (string)`

Read the contents of file *name*, and include it verbatim.

```
\markup {
  \verbatim-file #"en/included/simple.ly"
}

%% A simple piece in LilyPond, a scale.
\relative {
  c' d e f g a b c
}

%% Optional helper for automatic updating
%% by convert-ly. May be omitted.
\version "2.19.21"
```

`\whiteout arg (markup)`

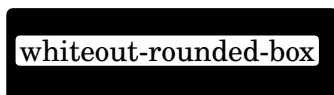
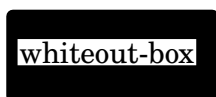
Provide a white background for *arg*. The shape of the white background is determined by *style*. The default is `box` which produces a rectangle. `rounded-box` produces a rounded rectangle. `outline` approximates the outline of the markup.

```
\markup {
  \combine
    \filled-box #'(-1 . 15) #'(-3 . 4) #1
    \override #'(thickness . 1.5)
  \whiteout whiteout-box
}
```

```

}
\markup {
  \combine
    \filled-box #'(-1 . 24) #'(-3 . 4) #1
    \override #'((style . rounded-box) (thickness . 3))
    \whiteout whiteout-rounded-box
}
\markup {
  \combine
    \filled-box #'(-1 . 18) #'(-3 . 4) #1
    \override #'((style . outline) (thickness . 3))
    \whiteout whiteout-outline
}

```



Used properties:

- `thickness (())`
- `style (box)`

`\with-color color (color) arg (markup)`

Draw *arg* in color specified by *color*.

```

\markup {
  \with-color #red
  red
  \hspace #2
  \with-color #green
  green
  \hspace #2
  \with-color "#0000ff"
  blue
}

```

**red green blue**

`\with-dimensions-from arg1 (markup) arg2 (markup)`

Print *arg2* with the horizontal and vertical dimensions of *arg1*.

`\with-dimensions x (pair of numbers) y (pair of numbers) arg (markup)`

Set the horizontal and vertical dimensions of *arg* to *x* and *y*.

`\with-link label (symbol) arg (markup)`

Add a link to the page holding label *label* around *arg*. This only works in the PDF backend.

```

\markup {

```



```
\with-link #'label {
  \italic { This links to the page
            containing the label... }
}
```

*This links to the page containing the label...*

`\with-outline` *outline* (markup) *arg* (markup)

Print *arg* with the outline and dimensions of *outline*. The outline is used by skylines to resolve collisions (not for whiteout).

## A.12 Instrucciones de listas de marcado de texto

Se puede usar cualquiera de las siguientes instrucciones con `\markuplist`:

`\column-lines` *args* (markup list)

Like `\column`, but return a list of lines instead of a single markup. `baseline-skip` determines the space between each markup in *args*.

Used properties:

- `baseline-skip`

`\justified-lines` *args* (markup list)

Like `\justify`, but return a list of lines instead of a single markup. Use `\override-lines #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\map-markup-commands` *compose* (procedure) *args* (markup list)

This applies the function *compose* to every markup in *args* (including elements of markup list command calls) in order to produce a new markup list. Since the return value from a markup list command call is not a markup list but rather a list of stencils, this requires passing those stencils off as the results of individual markup calls. That way, the results should work out as long as no markups rely on side effects.

`\override-lines` *new-prop* (pair) *args* (markup list)

Like `\override`, for markup lists.

`\score-lines` *score* (score)

This is the same as the `\score` markup but delivers its systems as a list of lines. Its *score* argument is entered in braces like it would be for `\score`.

`\string-lines` *strg* (string)

Takes the string *strg* and splits it at the character provided by the property `split-char`, defaulting to `#\newline`. Surrounding whitespace is removed from every resulting string. The returned list of markups is ready to be formatted by other markup or markup list commands like `\column`, `\line`, etc.

```
\markup {
  \column
```

```

\string-lines
"foo, foo,
bar, bar,
buzz, buzz!"
}

foo, foo,
bar, bar,
buzz, buzz!

```

Used properties:

- `split-char` (`#\newline`)

`\table` *column-align* (number list) *lst* (markup list)

Returns a table.

*column-align* specifies how each column is aligned, possible values are -1, 0, 1. The number of elements in *column-align* determines how many columns will be printed. The entries to print are given by *lst*, a markup-list. If needed, the last row is filled up with `point-stencils`. Overriding `padding` may be used to increase columns horizontal distance. Overriding `baseline-skip` to increase rows vertical distance.

```

\markuplist {
  \override #'(padding . 2)
  \table
    #'(0 1 0 -1)
    {
      \underline { center-aligned right-aligned
                    center-aligned left-aligned }
      one      \number 1 thousandth \number 0.001
      eleven   \number 11 hundredth  \number 0.01
      twenty   \number 20 tenth      \number 0.1
      thousand \number 1000 one       \number 1.0
    }
}

```

center-aligned right-aligned center-aligned left-aligned

one	<b>1</b>	thousandth	<b>0.001</b>
eleven	<b>11</b>	hundredth	<b>0.01</b>
twenty	<b>20</b>	tenth	<b>0.1</b>
thousand	<b>1000</b>	one	<b>1.0</b>

Used properties:

- `baseline-skip`
- `padding` (0)

`\table-of-contents`

Used properties:

- `baseline-skip`

`\wordwrap-internal justify` (boolean) *args* (markup list)

Internal markup list command used to define `\justify` and `\wordwrap`.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)

`\wordwrap-lines args` (markup list)

Like `\wordwrap`, but return a list of lines instead of a single markup. Use `\override-lines #'(line-width . X)` to set the line width, where *X* is the number of staff spaces.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\wordwrap-string-internal justify` (boolean) *arg* (string)

Internal markup list command that is used to define `\justify-string` and `\wordwrap-string`.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width`

## A.13 Lista de caracteres especiales

Se pueden usar las siguientes referencias a caracteres especiales; para ver más detalles, consulte [Alias de ASCII], página 547.

Se usa la sintaxis del HTML y casi todas estas referencias son las mismas que en el HTML. El resto están inspiradas en  $\text{\LaTeX}$ .

Los caracteres están enmarcados en un rectángulo de forma que pueda verse el tamaño que tienen. Se ha añadido un pequeño relleno de separación entre el carácter y el rectángulo para mejorar la legibilidad.

<code>&amp;iexcl;</code>	<code>&amp;iquest;</code>	<code>&amp;solidus;</code>	<code>&amp;flq;</code>	<code>&amp;frq;</code>
<code>&amp;frq;</code>	<code>&amp;flqq;</code>	<code>&amp;frqq;</code>	<code>&amp;glq;</code>	<code>&amp;glqq;</code>
<code>&amp;grq;</code>	<code>&amp;glqq;</code>	<code>&amp;grqq;</code>	<code>&amp;elq;</code>	<code>&amp;elqq;</code>
<code>&amp;erq;</code>	<code>&amp;elqq;</code>	<code>&amp;erqq;</code>	<code>&amp;ensp;</code>	<code>&amp;thinsp;</code>
<code>&amp;emsp;</code>	<code>&amp;thinsp;</code>	<code>&amp;nbsp;</code>	<code>&amp;nnbsp;</code>	<code>&amp;zwj;</code>
<code>&amp;zwj;</code>	<code>&amp;zwnj;</code>	<code>&amp;middot;</code>	<code>&amp;bull;</code>	<code>&amp;bullet;</code>

&copyright;	©	&registered;	®	&trademark;	™	&dagger;	†
&Dagger;	‡	&numero;	№	&ordf;	ª	&ordm;	º
&para;	¶	&sect;	§	&deg;	◻	&numero;	№
&permil;	‰	&brvbar;	ı	&acute;	◌́	&acutedbl;	◌̀
&grave;	◌̀	&breve;	◌̆	&caron;	◌̈́	&cedilla;	◌̈́
&circumflex;	◌̂	&diaeresis;	◌̈	&macron;	◌̄	&aa;	◌̄
&AA;	Å	&ae;	æ	&AE;	Æ	&auml;	ä
&Auml;	Ä	&dh;	ð	&DH;	Ð	&dj;	đ
&DJ;	Đ	&l;	ł	&L;	Ł	&ng;	ŋ
&NG;	Ŋ	&o;	ø	&O;	Ø	&oe;	œ
&OE;	Œ	&ouml;	ö	&Ouml;	Ö	&s;	ſ
&ss;	ß	&th;	þ	&TH;	Þ	&uuml;	ü
&Uuml;	Ü	&plus;	+	&minus;	=	&times;	×
&div;	÷	&sup1;	¹	&sup2;	²	&sup3;	³
&sqrt;	√	&increment;	Δ	&infty;	∞	&sum;	Σ
&pm;	±	&bullettop;	◦	&partial;	∂	&neg;	¬
&currency;	¤	&dollar;	\$	&euro;	€	&pounds;	£
&yen;	¥	&cent;	¢				

## A.14 Lista de articulaciones

En la lógica interna de LilyPond, una ‘articulación’ es cualquier objeto (que no sea una indicación de matiz dinámico) que puede ser adosado directamente después de un evento rítmico (o con una duración): notas, acordes; incluso silencios y saltos, o el artificio de acorde vacío <> (véase Sección “Estructura de un elemento de nota” en *Manual de Aprendizaje*). Incluso las ligaduras de expresión, las digitaciones y las inscripciones textuales son técnicamente articulaciones, aunque estas no se muestran aquí.

Por tanto, las listas que aparecen a continuación incluyen no solo las marcas de articulación, sino también todos los demás elementos textuales dentro de la tipografía Emmentaler que pueden ser adosados a notas (la forma en que se inserta un acento como ‘c’\accent’ o como ‘c’->’). Cada ejemplo muestra la inscripción en sus dos posibles posiciones verticales: respectivamente, *superior* e *inferior*, así como su posición por defecto (*neutra*). Consulte también [Glifos de inscripciones], página 722, para ver una lista más completa de glifos, para su utilización con la instrucción de marcado \musicglyph tal y como se explica en [Notación musical dentro de elementos de marcado], página 276.

### Indicaciones de articulación

\accent o ->



\espressivo



\marcato o -^



\portato o -\_



\staccatissimo o -!



\staccato o -.



\tenuto o --



### Indicaciones de adornos

\prall



\prallup



\pralldown



\upprall



\downprall



\prallprall



\lineprall



\prallmordent



\mordent



\upmordent



\downmordent



\trill



`\turn`



`\reverseturn`



`\slashturn`



`\haydnturn`



## Indicaciones de calderón

`\veryshortfermata`



`\shortfermata`



`\fermata`



`\longfermata`



`\verylongfermata`



`\henzeshortfermata`



`\henzelongfermata`



## Indicaciones específicas de ciertos instrumentos

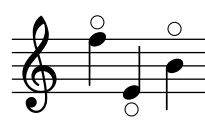
`\upbow`



`\downbow`



`\flageolet`



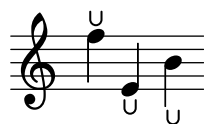
`\open`



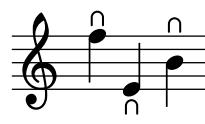
`\halfopen`



`\lheel`



`\rheel`



`\ltoe`



`\rtoe`



`\snappizzicato`



`\stopped or -+`



`\thumb`



## Indicaciones de repetición

`\segno`



`\coda`

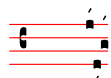


`\varcoda`

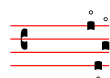


## Indicaciones antiguas

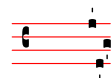
`\accentus`



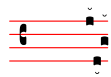
`\circculus`



`\ictus`



`\semicirculus`



`\signumcongruentiae`



## A.15 Notas de percusión

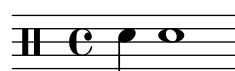
bassdrum  
bd



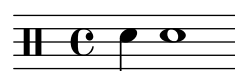
acousticbassdrum  
bda



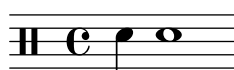
snare  
sn



acousticsnare  
sna



electricsnare  
sne



lowfloortom  
tomfl



highfloortom  
tomfh



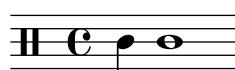
lowtom  
toml



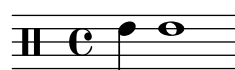
hightom  
tomh



lowmidtom  
tomml



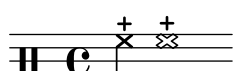
himidtom  
tommh



highhat  
hh



closedhihat  
hhc



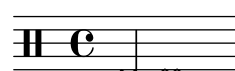
openhighhat  
hhho



halfopenhihat  
hhho



pedalhihat  
hhp



crashcymbal  
cymc



crashcymbala  
cymca



crashcymbalb  
cymcb



ridecymbal  
cymr



ridecymbala  
cymra



ridecymbalb  
cymrb



chineseecymbal  
cymch



splashcymbal  
cymss



ridebell  
rb



cowbell  
cb



hibongo  
boh



openhongo  
boho



mutehibongo  
bohmm



lobongo  
bol



openlobongo  
bolo



mutelobongo  
bolmm



hiconga  
cgh



openhiconga  
cgho



mutehiconga  
cghmm



loonga  
cgl



openloonga  
cglo



muteloonga  
cgllmm



hitimbale  
timh



lotimbale  
timl



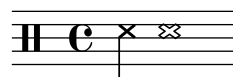
hiagogo  
agh



loagogo  
agll



sidestick  
ss



hisidestick  
ssh





losidestick  
ssl



guiro  
gui



shortguiro  
guis



longguiro  
guil



cabasa  
cab



maracas  
mar



shortwhistle  
whs



longwhistle  
whl



handclap  
hc



tambourine  
tamb



vibraslap  
vibs



tamtam  
tt



claves  
cl



hiwoodblock  
wbh



lowoodblock  
wbl



opencuica  
cuio



mutecuica  
cuim



triangle  
tri



opentriangle  
trio



mutetriangle  
trim



oneup  
ua



twoup  
ub



threeup  
uc



fourup  
ud



fiveup  
ue



onedown  
da



twodown  
db



threedown  
dc



`fourdown`  
`dd`



`fivedown`  
`de`



## A.16 Glosario técnico

Glosario de los términos técnicos y conceptos que se utilizan internamente en LilyPond. Estos términos pueden aparecer en los manuales, en las listas de distribución de correo o en el código fuente.

### **alist (lista-A)**

Una lista asociativa o abreviadamente una **lista-A** (alist en inglés) es una pareja de Scheme que asocia un valor con una clave: (`clave . valor`). Por ejemplo, en `scm/lily.scm`, la lista-A “type-p-name-alist” asocia ciertos predicados de tipo (p.ej. `ly:music?`) con nombres (p.ej. “music”) de forma que se pueda informar de los fallos de comprobación de tipo con un mensaje de consola que incluye el nombre del predicado de tipo esperado.

### **callback**

Una **callback** es una rutina, función o método cuya referencia se pasa como argumento en una llamada a otra rutina, permitiendo así que la rutina llamada invoque a aquélla. La técnica permite que una capa de software de nivel más bajo llame a una función definida en una capa de nivel más alto. Las funciones de callback se usan ampliamente en LilyPond para permitir al código de Scheme del nivel de usuario definir cuántas acciones de bajo nivel se llevan a cabo.

### **closure (cerradura)**

En Scheme, se crea una **cerradura** cuando una función, por lo general una expresión lambda, se pasa como variable. La cerradura contiene el código de la función y referencias a las ligaduras léxicas de las variables libres de la función (es decir, las variables que se usan en la expresión pero se definen fuera de ella). Cuando más tarde se aplica esta función a diferentes argumentos, las ligaduras de variables libres que se capturaron dentro de la cerradura se utilizan para obtener los valores de las variables libres que se usarán en el cálculo. Una propiedad útil de las cerraduras es la retención de los valores internos de las variables de una invocación a otra, permitiendo así que se pueda mantener un estado.

### **glifo**

Un **glifo** es una representación gráfica particular de un carácter tipográfico, o una combinación de dos caracteres que forman una ligadura. Un conjunto de glifos con un estilo y forma uniformes forman una fuente tipográfica, y un conjunto de fuentes tipográficas que abarcan varios estilos forman un tipo.

### **Véase también**

Referencia de la notación: Sección 1.8.3 [Tipografías], página 279, Sección 3.3.3 [Caracteres especiales], página 546.

### **grob (objeto gráfico)**

Los objetos de LilyPond que representan elementos de la notación en la salida impresa tales como la cabeza y la plica de las notas, ligaduras de unión y de expresión, digitaciones, claves,

et. se denominan ‘objetos de presentación’, a menudo conocidos como ‘Objetos GRáficos’, o abreviadamente **grobs**. Se representan mediante instancias de la clase **Grob**.

## Véase también

Manual de aprendizaje: Sección “Objetos e interfaces” en *Manual de Aprendizaje*, Sección “Convenciones de nombres de objetos y propiedades” en *Manual de Aprendizaje*, Sección “Propiedades de los objetos de presentación” en *Manual de Aprendizaje*.

Referencia de funcionamiento interno: Sección “grob-interface” en *Referencia de Funcionamiento Interno*, Sección “All layout objects” en *Referencia de Funcionamiento Interno*.

## immutable

Un objeto **immutable** es aquel cuyo estado no se puede modificar después de su creación, en contraste con los objetos mutables, que se pueden modificar después de su creación.

En LilyPond, las propiedades inmutables o compartidas definen el estilo y comportamiento predeterminados de los grobs. Se comparten por parte de muchos objetos. En aparente contradicción con su nombre, se pueden cambiar utilizando `\override` y `\revert`.

## Véase también

Referencia de la notación: [mutable], página 806.

## interfaz

Las acciones y propiedades comunes a un conjunto de grobs se agrupan en un objeto denominado **interfaz de grob** (**grob-interface**), o abreviadamente ‘interfaz’.

## Véase también

Manual de aprendizaje: Sección “Objetos e interfaces” en *Manual de Aprendizaje*, Sección “Convenciones de nombres de objetos y propiedades” en *Manual de Aprendizaje*, Sección “Propiedades de los interfaces” en *Manual de Aprendizaje*.

Referencia de la notación: Sección 5.2.2 [Interfaces de la presentación], página 643.

Referencia de funcionamiento interno: Sección “Graphical Object Interfaces” en *Referencia de Funcionamiento Interno*.

## lexer (analizador léxico)

Un **lexer** o analizador léxico es un programa que convierte una secuencia de caracteres en una secuencia de elementos o tokens, en un proceso que se llama análisis léxico. El analizador léxico de LilyPond convierte el flujo obtenido a partir de un archivo de entrada `.ly` en un flujo descompuesto en tokens más apto para la siguiente fase del procesado: el análisis sintáctico, véase [parser (analizador sintáctico)], página 807. El analizador léxico de LilyPond `lexer` está construido con la herramienta Flex a partir del archivo de lexer `lily/lexer.ll` que contiene las reglas léxicas. Este archivo es parte del código fuente y no se incluye dentro de la instalación binaria de LilyPond.

## mutable

Un objeto **mutable** es aquél cuyo estado se puede modificar después de su creación, en contraste con un objeto immutable, cuyo estado se fija en el momento de la creación.

En LilyPond, las propiedades mutables contienen valores específicos de un grob. Por lo general, las listas de otros objetos o los resultados de los cálculos se almacenan en propiedades mutables.

## Véase también

Referencia de la notación: [immutable], página 806.

## output-def (definición de salida)

Una instancia de la clase `Output-def` contiene los métodos y estructuras de datos asociados con un bloque de salida. Se crean instancias para los bloques `midi`, `layout` y `paper`.

## parser (analizador sintáctico)

Un **parser** o analizador sintáctico analiza la secuencia de tokens o elementos léxicos producida por un analizador léxico para determinar su estructura gramatical, agrupando los elementos léxicos en conjuntos mayores según las reglas de la gramática. Si la secuencia de elementos léxicos es válida, el producto final es un árbol de tokens cuya raíz es el símbolo inicial de la gramática. Si no se puede conseguir esto, el archivo es inválido y se produce un mensaje de error adecuado. Las agrupaciones sintácticas y las reglas para construir estas agrupaciones a partir de sus elementos constituyentes para la sintaxis de LilyPond están definidas en `lily/parser.yy` y se muestran en la forma normal de Backus (BNF) dentro de Sección “LilyPond grammar” en *Guía del colaborador*. Este archivo se usa para construir el analizador sintáctico durante la compilación del programa por parte del generador de analizadores sintácticos, Bison. Es parte del código fuente y no se incluye dentro de la instalación binaria de LilyPond.

## variable del analizador sintáctico

Son variables definidas directamente dentro de Scheme. Su uso directo por parte de los usuarios está fuertemente desaconsejado, porque su semántica de ámbito puede ser confusa.

Cuando el valor de una de estas variables se modifica dentro de un archivo `.ly`, el cambio es global, y a no ser que se revierta explícitamente, el nuevo valor persistirá hasta el final del archivo, afectando a todos los bloques `\score` así como a los archivos externos añadidos con la instrucción `\include`. Esto puede conducir a consecuencias imprevistas y en proyectos de composición tipográfica complejos puede ser difícil de rastrear.

LilyPond utiliza las siguientes variables del analizador sintáctico:

- `afterGraceFraction`
- `musicQuotes`
- `mode`
- `output-count`
- `output-suffix`
- `partCombineListener`
- `pitchnames`
- `toplevel-bookparts`
- `toplevel-scores`
- `showLastLength`
- `showFirstLength`

## prob (objeto de propiedades)

Los **Objetos de Propiedades**, o abreviadamente **probs**, son instancias de la clase `Prob`, que es una sencilla clase básica que tiene listas-A de propiedades mutables e inmutables y los métodos para manipularlas. Las clases `Music` y `Stream_event` derivan de `Prob`. También se crean instancias de la clase `Prob` para almacenar el contenido formateado de los grobs del sistema y los bloques de títulos durante el proceso de disposición de la página.

## smob (objeto de Scheme)

Los **Smobs**, u **Objetos de Scheme**, forman parte del mecanismo utilizado por Guile para exportar objetos de C y de C++ al código de Scheme. En LilyPond, se crean smobs a partir de objetos

de C++ por medio de macros. Hay dos tipos de objetos smob: los smobs simples, orientados a objetos inmutables simples como números, y los smobs complejos, usados para objetos con identidades. Si tiene acceso a las fuentes de LilyPond sources, encontrará más información en `lily/includes/smob.hh`.

## stencil (sello)

Las instancias de la clase **stencil** contienen la información necesaria para imprimir un objeto tipográfico. Es un smob simple que contiene una caja de confinamiento, que a su vez define las dimensiones vertical y horizontal del objeto, y una expresión de Scheme que imprime el objeto cuando se evalúa. Los stencils o sellos se pueden combinar para formar sellos más complejos definidos por un árbol de expresiones de Scheme formado a partir de las expresiones de Scheme de los sellos que lo componen.

La propiedad **stencil**, que conecta a un grob con su sello, se define dentro del interfaz `grob-interface`.

## Véase también

Referencia de funcionamiento interno: Sección “`grob-interface`” en *Referencia de Funcionamiento Interno*.

## A.17 Todas las propiedades de contexto

**accidentalGrouping** (symbol)

If set to 'voice, accidentals on the same note in different octaves may be horizontally staggered if in different voices.

**additionalBassStrings** (list)

The additional tablature bass-strings, which will not get a seprate line in TabStaff. It is a list of the pitches of each string (starting with the lowest numbered one).

**additionalPitchPrefix** (string)

Text with which to prefix additional pitches within a chord name.

**aDueText** (markup)

Text to print at a unisono passage.

**alignAboveContext** (string)

Where to insert newly created context in vertical alignment.

**alignBelowContext** (string)

Where to insert newly created context in vertical alignment.

**alterationGlyphs** (list)

Alist mapping alterations to accidental glyphs. Alterations are given as exact numbers, e.g., -1/2 for flat. This applies to all grobs that can print accidentals.

**alternativeNumber** (integer)

When set, the index of the current `\alternative` element, starting from one. Not set outside of alternatives. Note the distinction from volta number: an alternative may pertain to multiple volte.

**alternativeNumberingStyle** (symbol)

The scheme and style for numbering bars in repeat alternatives. If not set (the default), bar numbers continue through alternatives. Can be set to **numbers** to reset the bar number at each alternative, or set to **numbers-with-letters** to reset and also include letter suffixes.

**alternativeRestores** (symbol list)

Timing variables that are restored to their value at the start of the first alternative in subsequent alternatives.

**associatedVoice** (string)

Name of the context (see **associatedVoiceType** for its type, usually **Voice**) that has the melody for this **Lyrics** line.

**associatedVoiceType** (symbol)

Type of the context that has the melody for this **Lyrics** line.

**autoAccidentals** (list)

List of different ways to typeset an accidental.

For determining when to print an accidental, several different rules are tried. The rule that gives the highest number of accidentals is used.

Each entry in the list is either a symbol or a procedure.

*symbol*      The symbol is the name of the context in which the following rules are to be applied. For example, if *context* is Sección “Score” en *Referencia de Funcionamiento Interno* then all staves share accidentals, and if *context* is Sección “Staff” en *Referencia de Funcionamiento Interno* then all voices in the same staff share accidentals, but staves do not.

*procedure*   The procedure represents an accidental rule to be applied to the previously specified context.

The procedure takes the following arguments:

**context**      The current context to which the rule should be applied.

**pitch**        The pitch of the note to be evaluated.

**barnum**       The current bar number.

**measurepos**

The current measure position.

The procedure returns a pair of booleans. The first states whether an extra natural should be added. The second states whether an accidental should be printed. (**#t** . **#f**) does not make sense.

**autoBeamCheck** (procedure)

A procedure taking three arguments, *context*, *dir* [start/stop (-1 or 1)], and *test* [shortest note in the beam]. A non-**#f** return value starts or stops the auto beam.

**autoBeaming** (boolean)

If set to true then beams are generated automatically.

**autoCautionaries** (list)

List similar to **autoAccidentals**, but it controls cautionary accidentals rather than normal ones. Both lists are tried, and the one giving the most accidentals wins. In case of draw, a normal accidental is typeset.

**automaticBars** (boolean)

If set to false then bar lines will not be printed automatically; they must be explicitly created with a **\bar** command. Unlike the **\cadenzaOn** keyword, measures are still counted. Bar line generation will resume according to that count if this property is unset.

**barAlways** (boolean)

If set to true a bar line is drawn after each note.

**barCheckSynchronize** (boolean)

If true then reset **measurePosition** when finding a bar check.

**barNumberFormatter** (procedure)

A procedure that takes a bar number, measure position, and alternative number and returns a markup of the bar number to print.

**barNumberVisibility** (procedure)

A procedure that takes a bar number and a measure position and returns whether the corresponding bar number should be printed. Note that the actual print-out of bar numbers is controlled with the **break-visibility** property.

The following procedures are predefined:

**all-bar-numbers-visible**

Enable bar numbers for all bars, including the first one and broken bars (which get bar numbers in parentheses).

**first-bar-number-invisible**

Enable bar numbers for all bars (including broken bars) except the first one. If the first bar is broken, it doesn't get a bar number either.

**first-bar-number-invisible-save-broken-bars**

Enable bar numbers for all bars (including broken bars) except the first one. A broken first bar gets a bar number.

**first-bar-number-invisible-and-no-parenthesized-bar-numbers**

Enable bar numbers for all bars except the first bar and broken bars. This is the default.

**(every-nth-bar-number-visible *n*)**

Assuming *n* is value 2, for example, this enables bar numbers for bars 2, 4, 6, etc.

**(modulo-bar-number-visible *n m*)**

If bar numbers 1, 4, 7, etc., should be enabled, *n* (the modulo) must be set to 3 and *m* (the division remainder) to 1.

**baseMoment** (moment)

Smallest unit of time that will stand on its own as a subdivided section.

**beamExceptions** (list)

An alist of exceptions to autobeam rules that normally end on beats.

**beamHalfMeasure** (boolean)

Whether to allow a beam to begin halfway through the measure in triple time, which could look like 6/8.

**beatStructure** (list)

List of **baseMoments** that are combined to make beats.

**centerBarNumbers** (boolean)

Whether to center bar numbers in their measure instead of aligning them on the bar line.

**chordChanges** (boolean)

Only show changes in chords scheme?

**chordNameExceptions** (list)

An alist of chord exceptions. Contains (*chord . markup*) entries.

`chordNameFunction` (procedure)

The function that converts lists of pitches to chord names.

`chordNameLowercaseMinor` (boolean)

Downcase roots of minor chords?

`chordNameSeparator` (markup)

The markup object used to separate parts of a chord name.

`chordNoteNamer` (procedure)

A function that converts from a pitch object to a text markup. Used for single pitches.

`chordPrefixSpacer` (number)

The space added between the root symbol and the prefix of a chord name.

`chordRootNamer` (procedure)

A function that converts from a pitch object to a text markup. Used for chords.

`clefGlyph` (string)

Name of the symbol within the music font.

`clefPosition` (number)

Where should the center of the clef symbol go, measured in half staff spaces from the center of the staff.

`clefTransposition` (integer)

Add this much extra transposition. Values of 7 and -7 are common.

`clefTranspositionFormatter` (procedure)

A procedure that takes the Transposition number as a string and the style as a symbol and returns a markup.

`clefTranspositionStyle` (symbol)

Determines the way the ClefModifier grob is displayed. Possible values are ‘default’, ‘parenthesized’ and ‘bracketed’.

`completionBusy` (boolean)

Whether a completion-note head is playing.

`completionFactor` (an exact rational or procedure)

When `Completion_heads_engraver` and `Completion_rest_engraver` need to split a note or rest with a scaled duration, such as `c2*3`, this specifies the scale factor to use for the newly-split notes and rests created by the engraver.

If `#f`, the completion engraver uses the scale-factor of each duration being split.

If set to a callback procedure, that procedure is called with the context of the completion engraver, and the duration to be split.

`completionUnit` (moment)

Sub-bar unit of completion.

`connectArpeggios` (boolean)

If set, connect arpeggios across piano staff.

`countPercentRepeats` (boolean)

If set, produce counters for percent repeats.

`createKeyOnClefChange` (boolean)

Print a key signature whenever the clef is changed.

`createSpacing` (boolean)

Create `StaffSpacing` objects? Should be set for staves.



**crescendoSpanner** (symbol)

The type of spanner to be used for crescendo. Available values are ‘hairpin’ and ‘text’. If unset, a hairpin crescendo is used.

**crescendoText** (markup)

The text to print at start of non-hairpin crescendo, i.e., ‘cresc.’.

**cueClefGlyph** (string)

Name of the symbol within the music font.

**cueClefPosition** (number)

Where should the center of the clef symbol go, measured in half staff spaces from the center of the staff.

**cueClefTransposition** (integer)

Add this much extra transposition. Values of 7 and -7 are common.

**cueClefTranspositionFormatter** (procedure)

A procedure that takes the Transposition number as a string and the style as a symbol and returns a markup.

**cueClefTranspositionStyle** (symbol)

Determines the way the ClefModifier grob is displayed. Possible values are ‘default’, ‘parenthesized’ and ‘bracketed’.

**currentBarNumber** (integer)

Contains the current barnumber. This property is incremented at every bar line.

**decrescendoSpanner** (symbol)

The type of spanner to be used for decrescendi. Available values are ‘hairpin’ and ‘text’. If unset, a hairpin decrescendo is used.

**decrescendoText** (markup)

The text to print at start of non-hairpin decrescendo, i.e., ‘dim.’.

**defaultBarType** (string)

Set the default type of bar line. See **whichBar** for information on available bar types. This variable is read by Sección “Timing\_translator” en *Referencia de Funcionamiento Interno* at Sección “Score” en *Referencia de Funcionamiento Interno* level.

**defaultStrings** (list)

A list of strings to use in calculating frets for tablatures and fretboards if no strings are provided in the notes for the current moment.

**doubleRepeatSegnoType** (string)

Set the default bar line for the combinations double repeat with segno. Default is ‘:|.S.|:’.

**doubleRepeatType** (string)

Set the default bar line for double repeats.

**doubleSlurs** (boolean)

If set, two slurs are created for every slurred note, one above and one below the chord.

**drumPitchTable** (hash table)

A table mapping percussion instruments (symbols) to pitches.

**drumStyleTable** (hash table)

A hash table which maps drums to layout settings. Predefined values: ‘drums-style’, ‘agostini-drums-style’, ‘timbales-style’, ‘congas-style’, ‘bongos-style’, and ‘percussion-style’.

The layout style is a hash table, containing the drum-pitches (e.g., the symbol ‘hihat’) as keys, and a list (*notehead-style script vertical-position*) as values.

**endAtSkip** (boolean)

End *DurationLine* grob on *skip-event*

**endRepeatSegnoType** (string)

Set the default bar line for the combinations ending of repeat with segno. Default is ‘|.S’.

**endRepeatType** (string)

Set the default bar line for the ending of repeats.

**explicitClefVisibility** (vector)

‘break-visibility’ function for clef changes.

**explicitCueClefVisibility** (vector)

‘break-visibility’ function for cue clef changes.

**explicitKeySignatureVisibility** (vector)

‘break-visibility’ function for explicit key changes. ‘\override’ of the *break-visibility* property will set the visibility for normal (i.e., at the start of the line) key signatures.

**extendersOverRests** (boolean)

Whether to continue extenders as they cross a rest.

**extraNatural** (boolean)

Whether to typeset an extra natural sign before accidentals that reduce the effect of a previous alteration.

**figuredBassAlterationDirection** (direction)

Where to put alterations relative to the main figure.

**figuredBassCenterContinuations** (boolean)

Whether to vertically center pairs of extender lines. This does not work with three or more lines.

**figuredBassFormatter** (procedure)

A routine generating a markup for a bass figure.

**figuredBassPlusDirection** (direction)

Where to put plus signs relative to the main figure.

**fineBarType** (string)

The bar line for *\fine*. See *whichBar* for information on available bar types.

**fineSegnoType** (string)

Set the default bar line for a requested segno with fine. Default is ‘|.S’.

**fineStartRepeatSegnoType** (string)

Set the default bar line for the combinations beginning of repeat with segno and fine. Default is ‘|.S.|:’.

**fineText** (markup)

The text to print at *\fine*.

**fingeringOrientations** (list)

A list of symbols, containing ‘left’, ‘right’, ‘up’ and/or ‘down’. This list determines where fingerings are put relative to the chord being fingered.

- firstClef** (boolean)  
If true, create a new clef when starting a staff.
- followVoice** (boolean)  
If set, note heads are tracked across staff switches by a thin line.
- fontSize** (number)  
The relative size of all grobs in a context.
- forbidBreak** (boolean)  
If set to **#t**, prevent a line break at this point.
- forceClef** (boolean)  
Show clef symbol, even if it has not changed. Only active for the first clef after the property is set, not for the full staff.
- fretLabels** (list)  
A list of strings or Scheme-formatted markups containing, in the correct order, the labels to be used for lettered frets in tablature.
- glissandoMap** (list)  
A map in the form of `'((source1 . target1) (source2 . target2) (sourcen . targetn))` showing the glissandi to be drawn for note columns. The value `'()` will default to `'((0 . 0) (1 . 1) (n . n))`, where `n` is the minimal number of note-heads in the two note columns between which the glissandi occur.
- gridInterval** (moment)  
Interval for which to generate **GridPoints**.
- handleNegativeFrets** (symbol)  
How the automatic fret calculator should handle calculated negative frets. Values include **'ignore**, to leave them out of the diagram completely, **'include**, to include them as calculated, and **'recalculate**, to ignore the specified string and find a string where they will fit with a positive fret number.
- harmonicAccidentals** (boolean)  
If set, harmonic notes in chords get accidentals.
- harmonicDots** (boolean)  
If set, harmonic notes in dotted chords get dots.
- highStringOne** (boolean)  
Whether the first string is the string with highest pitch on the instrument. This used by the automatic string selector for tablature notation.
- ignoreBarChecks** (boolean)  
Ignore bar checks.
- ignoreBarNumberChecks** (boolean)  
Ignore bar number checks.
- ignoreFiguredBassRest** (boolean)  
Don't swallow rest events.
- ignoreMelismata** (boolean)  
Ignore melismata for this Sección “Lyrics” en *Referencia de Funcionamiento Interno* line.
- implicitBassFigures** (list)  
A list of bass figures that are not printed as numbers, but only as extender lines.

- includeGraceNotes** (boolean)  
Do not ignore grace notes for Sección “Lyrics” en *Referencia de Funcionamiento Interno*.
- initialTimeSignatureVisibility** (vector)  
break visibility for the initial time signature.
- instrumentCueName** (markup)  
The name to print if another instrument is to be taken.
- instrumentEqualizer** (procedure)  
A function taking a string (instrument name), and returning a (*min* . *max*) pair of numbers for the loudness range of the instrument.
- instrumentName** (markup)  
The name to print left of a staff. The **instrumentName** property labels the staff in the first system, and the **shortInstrumentName** property labels following lines.
- instrumentTransposition** (pitch)  
Define the transposition of the instrument. Its value is the pitch that sounds when the instrument plays written middle C. This is used to transpose the MIDI output, and \quotes.
- internalBarNumber** (integer)  
Contains the current barnumber. This property is used for internal timekeeping, among others by the **Accidental\_engraver**.
- keepAliveInterfaces** (list)  
A list of symbols, signifying grob interfaces that are worth keeping a staff with **remove-empty** set around for.
- keyAlterationOrder** (list)  
A list of pairs that defines in what order alterations should be printed. The format of an entry is (*step* . *alter*), where *step* is a number from 0 to 6 and *alter* from -1 (double flat) to 1 (double sharp), with exact rationals for alterations in between, e.g., 1/2 for sharp.
- keyAlterations** (list)  
The current key signature. This is an alist containing (*step* . *alter*) or ((*octave* . *step*) . *alter*), where *step* is a number in the range 0 to 6 and *alter* a fraction, denoting alteration. For alterations, use symbols, e.g. **keyAlterations** = #`((6 . ,FLAT)).
- lyricMelismaAlignment** (number)  
Alignment to use for a melisma syllable.
- magnifyStaffValue** (positive number)  
The most recent value set with \magnifyStaff.
- majorSevenSymbol** (markup)  
How should the major 7th be formatted in a chord name?
- markFormatter** (procedure)  
A procedure taking as arguments the context and the rehearsal mark. It should return the formatted mark as a markup object.
- maximumFretStretch** (number)  
Don’t allocate frets further than this from specified frets.
- measureLength** (moment)  
Length of one measure in the current time signature.

**measurePosition** (moment)

How much of the current measure have we had. This can be set manually to create incomplete measures.

**measureStartNow** (boolean)

True at the beginning of a measure.

**melismaBusyProperties** (list)

A list of properties (symbols) to determine whether a melisma is playing. Setting this property will influence how lyrics are aligned to notes. For example, if set to '(melismaBusy beamMelismaBusy)', only manual melismata and manual beams are considered. Possible values include **melismaBusy**, **slurMelismaBusy**, **tieMelismaBusy**, and **beamMelismaBusy**.

**metronomeMarkFormatter** (procedure)

How to produce a metronome markup. Called with two arguments: a **TempoChangeEvent** and context.

**middleCClefPosition** (number)

The position of the middle C, as determined only by the clef. This can be calculated by looking at **clefPosition** and **clefGlyph**.

**middleCCuePosition** (number)

The position of the middle C, as determined only by the clef of the cue notes. This can be calculated by looking at **cueClefPosition** and **cueClefGlyph**.

**middleCOffset** (number)

The offset of middle C from the position given by **middleCClefPosition**. This is used for ottava brackets.

**middleCPosition** (number)

The place of the middle C, measured in half staff-spaces. Usually determined by looking at **middleCClefPosition** and **middleCOffset**.

**midiBalance** (number)

Stereo balance for the MIDI channel associated with the current context. Ranges from -1 to 1, where the values -1 (**#LEFT**), 0 (**#CENTER**) and 1 (**#RIGHT**) correspond to leftmost emphasis, center balance, and rightmost emphasis, respectively.

**midiChannelMapping** (symbol)

How to map MIDI channels: per **staff** (default), **instrument** or **voice**.

**midiChorusLevel** (number)

Chorus effect level for the MIDI channel associated with the current context. Ranges from 0 to 1 (0=off, 1=full effect).

**midiExpression** (number)

Expression control for the MIDI channel associated with the current context. Ranges from 0 to 1 (0=off, 1=full effect).

**midiInstrument** (string)

Name of the MIDI instrument to use.

**midiMaximumVolume** (number)

Analogous to **midiMinimumVolume**.

**midiMergeUnisons** (boolean)

If true, output only one MIDI note-on event when notes with the same pitch, in the same MIDI-file track, overlap.

- midiMinimumVolume** (number)  
Set the minimum loudness for MIDI. Ranges from 0 to 1.
- midiPanPosition** (number)  
Pan position for the MIDI channel associated with the current context. Ranges from -1 to 1, where the values -1 (**#LEFT**), 0 (**#CENTER**) and 1 (**#RIGHT**) correspond to hard left, center, and hard right, respectively.
- midiReverbLevel** (number)  
Reverb effect level for the MIDI channel associated with the current context. Ranges from 0 to 1 (0=off, 1=full effect).
- minimumFret** (number)  
The tablature auto string-selecting mechanism selects the highest string with a fret at least **minimumFret**.
- minimumPageTurnLength** (moment)  
Minimum length of a rest for a page turn to be allowed.
- minimumRepeatLengthForPageTurn** (moment)  
Minimum length of a repeated section for a page turn to be allowed within that section.
- minorChordModifier** (markup)  
Markup displayed following the root for a minor chord
- noChordSymbol** (markup)  
Markup to be displayed for rests in a ChordNames context.
- noteNameFunction** (procedure)  
Function used to convert pitches into strings and markups.
- noteNameSeparator** (string)  
String used to separate simultaneous NoteName objects.
- noteToFretFunction** (procedure)  
Convert list of notes and list of defined strings to full list of strings and fret numbers. Parameters: The context, a list of note events, a list of tabstring events, and the fretboard grob if a fretboard is desired.
- nullAccidentals** (boolean)  
The **Accidental\_engraver** generates no accidentals for notes in contexts where this is set. In addition to suppressing the printed accidental, this option removes any effect the note would have had on accidentals in other voices.
- ottavaStartNow** (boolean)  
Is an ottava starting in this time step?
- ottavation** (markup)  
If set, the text for an ottava spanner. Changing this creates a new text spanner.
- ottavationMarkups** (list)  
An alist defining the markups used for ottava brackets. It contains entries of the form (*number of octaves* . *markup*).
- output** (music output)  
The output produced by a score-level translator during music interpretation.
- partCombineForced** (symbol)  
Override for the partCombine decision. Can be **apart**, **chords**, **unisono**, **solo1**, or **solo2**.

- partCombineTextsOnNote** (boolean)  
Print part-combine texts only on the next note rather than immediately on rests or skips.
- pedalSostenutoStrings** (list)  
See **pedalSustainStrings**.
- pedalSostenutoStyle** (symbol)  
See **pedalSustainStyle**.
- pedalSustainStrings** (list)  
A list of strings to print for sustain-pedal. Format is (*up updown down*), where each of the three is the string to print when this is done with the pedal.
- pedalSustainStyle** (symbol)  
A symbol that indicates how to print sustain pedals: **text**, **bracket** or **mixed** (both).
- pedalUnaCordaStrings** (list)  
See **pedalSustainStrings**.
- pedalUnaCordaStyle** (symbol)  
See **pedalSustainStyle**.
- predefinedDiagramTable** (hash table)  
The hash table of predefined fret diagrams to use in **FretBoards**.
- printAccidentalNames** (boolean or symbol)  
Print accidentals in the **NoteNames** context.
- printKeyCancellation** (boolean)  
Print restoration alterations before a key signature change.
- printNotesLanguage** (string)  
Use a specific language in the **NoteNames** context.
- printOctaveNames** (boolean or symbol)  
Print octave marks in the **NoteNames** context.
- printPartCombineTexts** (boolean)  
Set ‘Solo’ and ‘A due’ texts in the part combiner?
- proportionalNotationDuration** (moment)  
Global override for shortest-playing duration. This is used for switching on proportional notation.
- rehearsalMark** (integer)  
The last rehearsal mark printed.
- repeatCommands** (list)  
This property is a list of commands of the form (**list** ‘volta *x*’), where *x* is a string or #f. ‘end-repeat’ is also accepted as a command.
- repeatCountVisibility** (procedure)  
A procedure taking as arguments an integer and context, returning whether the corresponding percent repeat number should be printed when **countPercentRepeats** is set.
- restCompletionBusy** (boolean)  
Signal whether a completion-rest is active.
- restNumberThreshold** (number)  
If a multimeasure rest has more measures than this, a number is printed.

**restrainOpenStrings** (boolean)

Exclude open strings from the automatic fret calculator.

**searchForVoice** (boolean)

Signal whether a search should be made of all contexts in the context hierarchy for a voice to provide rhythms for the lyrics.

**sectionBarType** (string)

The bar line for `\section`. See **whichBar** for information on available bar types.

**segnoType** (string)

Set the default bar line for a requested segno. Default is ‘S’.

**shapeNoteStyles** (vector)

Vector of symbols, listing style for each note head relative to the tonic (qv.) of the scale.

**shortInstrumentName** (markup)

See **instrumentName**.

**shortVocalName** (markup)

Name of a vocal line, short version.

**skipBars** (boolean)

If set to true, then skip the empty bars that are produced by multimeasure notes and rests. These bars will not appear on the printed output. If not set (the default), multimeasure notes and rests expand into their full length, printing the appropriate number of empty bars so that synchronization with other voices is preserved.

```
{
  r1 r1*3 R1*3
  \set Score.skipBars= ##t
  r1*3 R1*3
}
```

**skipTypesetting** (boolean)

If true, no typesetting is done, speeding up the interpretation phase. Useful for debugging large scores.

**slashChordSeparator** (markup)

The markup object used to separate a chord name from its root note in case of inversions or slash chords.

**soloIIIText** (markup)

The text for the start of a solo for voice ‘two’ when part-combining.

**soloText** (markup)

The text for the start of a solo when part-combining.

**squashedPosition** (integer)

Vertical position of squashing for Sección “Pitch\_squash\_engraver” en *Referencia de Funcionamiento Interno*.

**staffLineLayoutFunction** (procedure)

Layout of staff lines, **traditional**, or **semitone**.

**stanza** (markup)

Stanza ‘number’ to print before the start of a verse. Use in **Lyrics** context.

**startAtNoteColumn** (boolean)

Start **DurationLine** grob at entire **NoteColumn**.



- startAtSkip** (boolean)  
Start **DurationLine** grob at **skip-event**.
- startRepeatSegnoType** (string)  
Set the default bar line for the combinations beginning of repeat with segno. Default is 'S.|:'.  
See **startRepeatType**.
- startRepeatType** (string)  
Set the default bar line for the beginning of repeats.
- stemLeftBeamCount** (integer)  
Specify the number of beams to draw on the left side of the next note. Overrides automatic beaming. The value is only used once, and then it is erased.
- stemRightBeamCount** (integer)  
See **stemLeftBeamCount**.
- strictBeatBeaming** (boolean)  
Should partial beams reflect the beat structure even if it causes flags to hang out?
- stringNumberOrientations** (list)  
See **fingeringOrientations**.
- stringOneTopmost** (boolean)  
Whether the first string is printed on the top line of the tablature.
- stringTunings** (list)  
The tablature strings tuning. It is a list of the pitches of each string (starting with the lowest numbered one).
- strokeFingerOrientations** (list)  
See **fingeringOrientations**.
- subdivideBeams** (boolean)  
If set, multiple beams will be subdivided at **baseMoment** positions by only drawing one beam over the beat.
- suggestAccidentals** (boolean or symbol)  
If set to **#t**, accidentals are typeset as suggestions above the note. Setting it to 'cautionary' only applies that to cautionary accidentals.
- supportNonIntegerFret** (boolean)  
If set in **Score** the **TabStaff** will print micro-tones as  $2\frac{1}{2}$ .
- suspendMelodyDecisions** (boolean)  
When using the **Melody\_engraver**, stop changing orientation of stems based on the melody when this is set to true.
- suspendRestMerging** (boolean)  
When using the **Merge\_rest\_engraver** do not merge rests when this is set to true.
- systemStartDelimiter** (symbol)  
Which grob to make for the start of the system/staff? Set to **SystemStartBrace**, **SystemStartBracket** or **SystemStartBar**.
- systemStartDelimiterHierarchy** (pair)  
A nested list, indicating the nesting of a start delimiters.
- tablatureFormat** (procedure)  
A function formatting a tablature note head. Called with three arguments: context, string number and, fret number. It returns the text as a markup.

**tabStaffLineLayoutFunction** (procedure)

A function determining the staff position of a tablature note head. Called with two arguments: the context and the string.

**tempoHideNote** (boolean)

Hide the note = count in tempo marks.

**tempoWholesPerMinute** (moment)

The tempo in whole notes per minute.

**tieWaitForNote** (boolean)

If true, tied notes do not have to follow each other directly. This can be used for writing out arpeggios.

**timeSignatureFraction** (fraction, as pair)

A pair of numbers, signifying the time signature. For example, '(4 . 4)' is a 4/4 time signature.

**timeSignatureSettings** (list)

A nested alist of settings for time signatures. Contains elements for various time signatures. The element for each time signature contains entries for **baseMoment**, **beatStructure**, and **beamExceptions**.

**timing** (boolean)

Keep administration of measure length, position, bar number, etc.? Switch off for cadenzas.

**tonic** (pitch)

The tonic of the current scale.

**topLevelAlignment** (boolean)

If true, the *Vertical\_align\_engraver* will create a *VerticalAlignment*; otherwise, it will create a *StaffGrouper*

**tupletFullLength** (boolean)

If set, the tuplet is printed up to the start of the next note.

**tupletFullLengthNote** (boolean)

If set, end at the next note, otherwise end on the matter (time signatures, etc.) before the note.

**tupletSpannerDuration** (moment)

Normally, a tuplet bracket is as wide as the `\times` expression that gave rise to it. By setting this property, you can make brackets last shorter.

```
{
  \set tupletSpannerDuration = #(ly:make-moment 1 4)
  \times 2/3 { c8 c c c c c }
}
```

**underlyingRepeatType** (string)

Set the bar line to use at points of repetition or departure where no bar line would normally appear, for example at the end of a system broken in mid measure where the next system begins with a segno.

**useBassFigureExtenders** (boolean)

Whether to use extender lines for repeated bass figures.

**vocalName** (markup)

Name of a vocal line.

`voltaSpannerDuration` (moment)

This specifies the maximum duration to use for the brackets printed for `\alternative`. This can be used to shrink the length of brackets in the situation where one alternative is very large.

`whichBar` (string)

This property is read to determine what type of bar line to create.

Example:

```
\set Staff.whichBar = ".|:"
```

This will create a start-repeat bar in this staff only. Valid values are described in `scm/bar-line.scm`.

## A.18 Propiedades de disposición

`add-stem-support` (boolean)

If set, the `Stem` object is included in this script's support.

`after-line-breaking` (boolean)

Dummy property, used to trigger callback for `after-line-breaking`.

`align-dir` (direction)

Which side to align? -1: left side, 0: around center of width, 1: right side.

`allow-loose-spacing` (boolean)

If set, column can be detached from main spacing.

`allow-span-bar` (boolean)

If false, no inter-staff bar line will be created below this bar line.

`alteration` (number)

Alteration numbers for accidental.

`alteration-alist` (list)

List of (`pitch` . `accidental`) pairs for key signature.

`alteration-glyph-name-alist` (list)

An alist of key-string pairs.

`annotation-balloon` (boolean)

Print the balloon around an annotation.

`annotation-line` (boolean)

Print the line from an annotation to the grob that it annotates.

`arpeggio-direction` (direction)

If set, put an arrow on the arpeggio squiggly line.

`arrow-length` (number)

Arrow length.

`arrow-width` (number)

Arrow width.

`auto-knee-gap` (dimension, in staff space)

If a gap is found between note heads where a horizontal beam fits and it is larger than this number, make a kneed beam.

`automatically-numbered` (boolean)

If set, footnotes are automatically numbered.

**average-spacing-wishes** (boolean)

If set, the spacing wishes are averaged over staves.

**avoid-note-head** (boolean)

If set, the stem of a chord does not pass through all note heads, but starts at the last note head.

**avoid-scripts** (boolean)

If set, a tuplet bracket avoids the scripts associated with the note heads it encompasses.

**avoid-slur** (symbol)

Method of handling slur collisions. Choices are **inside**, **outside**, **around**, and **ignore**. **inside** adjusts the slur if needed to keep the grob inside the slur. **outside** moves the grob vertically to the outside of the slur. **around** moves the grob vertically to the outside of the slur only if there is a collision. **ignore** does not move either. In grobs whose notational significance depends on vertical position (such as accidentals, clefs, etc.), **outside** and **around** behave like **ignore**.

**axes** (list) List of axis numbers. In the case of alignment grobs, this should contain only one number.

**bar-extent** (pair of numbers)

The Y-extent of the actual bar line. This may differ from **Y-extent** because it does not include the dots in a repeat bar line.

**base-shortest-duration** (moment)

Spacing is based on the shortest notes in a piece. Normally, pieces are spaced as if notes at least as short as this are present.

**baseline-skip** (dimension, in staff space)

Distance between base lines of multiple lines of text.

**beam-thickness** (dimension, in staff space)

Beam thickness, measured in **staff-space** units.

**beam-width** (dimension, in staff space)

Width of the tremolo sign.

**beamed-stem-shorten** (list)

How much to shorten beamed stems, when their direction is forced. It is a list, since the value is different depending on the number of flags and beams.

**beaming** (pair)

Pair of number lists. Each number list specifies which beams to make. 0 is the central beam, 1 is the next beam toward the note, etc. This information is used to determine how to connect the beaming patterns from stem to stem inside a beam.

**beamlet-default-length** (pair)

A pair of numbers. The first number specifies the default length of a beamlet that sticks out of the left hand side of this stem; the second number specifies the default length of the beamlet to the right. The actual length of a beamlet is determined by taking either the default length or the length specified by **beamlet-max-length-proportion**, whichever is smaller.

**beamlet-max-length-proportion** (pair)

The maximum length of a beamlet, as a proportion of the distance between two adjacent stems.

**before-line-breaking** (boolean)

Dummy property, used to trigger a callback function.

**bend-me** (boolean)

Decide whether this grob is bent.

**between-cols** (pair)

Where to attach a loose column to.

**bound-details** (list)

An alist of properties for determining attachments of spanners to edges.

**bound-padding** (number)

The amount of padding to insert around spanner bounds.

**bracket-flare** (pair of numbers)

A pair of numbers specifying how much edges of brackets should slant outward. Value 0.0 means straight edges.

**bracket-visibility** (boolean or symbol)

This controls the visibility of the tuplet bracket. Setting it to false prevents printing of the bracket. Setting the property to **if-no-beam** makes it print only if there is no beam associated with this tuplet bracket.

**break-align-anchor** (number)

Grobs aligned to this breakable item will have their X-offsets shifted by this number. In bar lines, for example, this is used to position grobs relative to the (visual) center of the bar line.

**break-align-anchor-alignment** (number)

Read by `ly:break-aligned-interface::calc-extent-aligned-anchor` for aligning an anchor to a grob's extent.

**break-align-orders** (vector)

This is a vector of 3 lists: `#{end-of-line unbroken start-of-line}`. Each list contains *break-align symbols* that specify an order of breakable items (see Sección “break-alignment-interface” en *Referencia de Funcionamiento Interno*).

For example, this places time signatures before clefs:

```
\override Score.BreakAlignment.break-align-orders =
  #(make-vector 3 '(left-edge
                    cue-end-clef
                    ambitus
                    breathing-sign
                    time-signature
                    clef
                    cue-clef
                    staff-bar
                    key-cancellation
                    key-signature
                    custos))
```

**break-align-symbol** (symbol)

This key is used for aligning, ordering, and spacing breakable items. See Sección “break-alignment-interface” en *Referencia de Funcionamiento Interno*.

**break-align-symbols** (list)

A list of *break-align symbols* that determines which breakable items to align this to. If the grob selected by the first symbol in the list is invisible due to **break-visibility**,

we will align to the next grob (and so on). Choices are listed in Sección “break-alignment-interface” en *Referencia de Funcionamiento Interno*.

**break-overshoot** (pair of numbers)

How much does a broken spanner stick out of its bounds?

**break-visibility** (vector)

A vector of 3 booleans, *#(end-of-line unbroken begin-of-line)*. *#t* means visible, *#f* means killed.

**breakable** (boolean)

Allow breaks here.

**broken-bound-padding** (number)

The amount of padding to insert when a spanner is broken at a line break.

**chord-dots-limit** (integer)

Limits the column of dots on each chord to the height of the chord plus **chord-dots-limit** staff-positions.

**circled-tip** (boolean)

Put a circle at start/end of hairpins (al/del niente).

**clef-alignments** (list)

An alist of parent-alignments that should be used for clef modifiers with various clefs

**clip-edges** (boolean)

Allow outward pointing beamlets at the edges of beams?

**collapse-height** (dimension, in staff space)

Minimum height of system start delimiter. If equal or smaller, the bracket/brace/line is removed.

**collision-interfaces** (list)

A list of interfaces for which automatic beam-collision resolution is run.

**collision-voice-only** (boolean)

Does automatic beam collision apply only to the voice in which the beam was created?

**color** (color)

The color of this grob.

**common-shortest-duration** (moment)

The most common shortest note length. This is used in spacing. Enlarging this sets the score tighter.

**concaveness** (number)

A beam is concave if its inner stems are closer to the beam than the two outside stems. This number is a measure of the closeness of the inner stems. It is used for damping the slope of the beam.

**connect-to-neighbor** (pair)

Pair of booleans, indicating whether this grob looks as a continued break.

**control-points** (list of number pairs)

List of offsets (number pairs) that form control points for the tie, slur, or bracket shape. For Béziers, this should list the control points of a third-order Bézier curve.

**count-from** (integer)

The first measure in a measure count receives this number. The following measures are numbered in increments from this initial value.

- damping** (number)  
Amount of beam slope damping.
- dash-definition** (pair)  
List of **dash-elements** defining the dash structure. Each **dash-element** has a starting t value, an ending t-value, a **dash-fraction**, and a **dash-period**.
- dash-fraction** (number)  
Size of the dashes, relative to **dash-period**. Should be between 0.1 and 1.0 (continuous line). If set to 0.0, a dotted line is produced
- dash-period** (number)  
The length of one dash together with whitespace. If negative, no line is drawn at all.
- dashed-edge** (boolean)  
If set, the bracket edges are dashed like the rest of the bracket.
- default-direction** (direction)  
Direction determined by note head positions.
- default-staff-staff-spacing** (list)  
The settings to use for **staff-staff-spacing** when it is unset, for ungrouped staves and for grouped staves that do not have the relevant **StaffGrouper** property set (**staff-staff-spacing** or **staffgroup-staff-spacing**).
- details** (list)  
Alist of parameters for detailed grob behavior. More information on the allowed parameters for a grob can be found by looking at the top of the Internals Reference page for each interface having a **details** property.
- digit-names** (vector)  
Names for string finger digits.
- direction** (direction)  
If **side-axis** is 0 (or X), then this property determines whether the object is placed LEFT, CENTER or RIGHT with respect to the other object. Otherwise, it determines whether the object is placed UP, CENTER or DOWN. Numerical values may also be used: UP=1, DOWN=-1, LEFT=-1, RIGHT=1, CENTER=0.
- dot-count** (integer)  
The number of dots.
- dot-negative-kern** (number)  
The space to remove between a dot and a slash in percent repeat glyphs. Larger values bring the two elements closer together.
- dot-placement-list** (list)  
List consisting of (*description string-number fret-number finger-number*) entries used to define fret diagrams.
- double-stem-separation** (number)  
The distance between the two stems of a half note in tablature when using **\tabFullNotation**, not counting the width of the stems themselves, expressed as a multiple of the default height of a staff-space in the traditional five-line staff.
- duration-log** (integer)  
The 2-log of the note head duration, i.e., 0 = whole note, 1 = half note, etc.
- eccentricity** (number)  
How asymmetrical to make a slur. Positive means move the center to the right.

**edge-height** (pair)

A pair of numbers specifying the heights of the vertical edges: (*left-height* . *right-height*).

**edge-text** (pair)

A pair specifying the texts to be set at the edges: (*left-text* . *right-text*).

**endpoint-alignments** (pair of numbers)

A pair of numbers representing the alignments of an object's endpoints. E.g., the ends of a hairpin relative to `NoteColumn` grobs.

**expand-limit** (integer)

Maximum number of measures expanded in church rests.

**extra-dy** (number)

Slope glissandi this much extra.

**extra-offset** (pair of numbers)

A pair representing an offset. This offset is added just before outputting the symbol, so the typesetting engine is completely oblivious to it. The values are measured in **staff-space** units of the staff's `StaffSymbol`.

**extra-spacing-height** (pair of numbers)

In the horizontal spacing problem, we increase the height of each item by this amount (by adding the 'car' to the bottom of the item and adding the 'cdr' to the top of the item). In order to make a grob infinitely high (to prevent the horizontal spacing problem from placing any other grobs above or below this grob), set this to (`-inf.0` . `+inf.0`).

**extra-spacing-width** (pair of numbers)

In the horizontal spacing problem, we pad each item by this amount (by adding the 'car' on the left side of the item and adding the 'cdr' on the right side of the item). In order to make a grob take up no horizontal space at all, set this to (`+inf.0` . `-inf.0`).

**extroversion** (number)

For polygons, how the thickness of the line is spread on each side of the exact polygon with ideal zero thickness. If this is 0, the middle of line is on the polygon. If 1, the line sticks out of the polygon. If -1, the outer side of the line is exactly on the polygon. Other numeric values are interpolated.

**filled** (boolean)

Whether an object is filled with ink.

**flag-count** (number)

The number of tremolo beams.

**flag-style** (symbol)

The style of the flag to be used with `MetronomeMark`. Available are 'modern-straight-flag', 'old-straight-flag', 'flat-flag', 'mensural' and 'default'.

**flat-positions** (list)

Flats in key signatures are placed within the specified ranges of staff-positions. The general form is a list of pairs, with one pair for each type of clef, in order of the staff-position at which each clef places C: (`alto` `treble` `tenor` `soprano` `baritone` `mezzosoprano` `bass`). If the list contains a single element it applies for all clefs. A single number in place of a pair sets accidentals within the octave ending at that staff-position.



**font-encoding** (symbol)

The font encoding is the broadest category for selecting a font. Currently, only LilyPond's system fonts (Emmentaler) are using this property. Available values are `fetaMusic` (Emmentaler), `fetaBraces`, `fetaText` (Emmentaler).

**font-family** (symbol)

The font family is the broadest category for selecting text fonts. Options include: `sans`, `roman`.

**font-features** (list)

Opentype features.

**font-name** (string)

Specifies a file name (without extension) of the font to load. This setting overrides selection using `font-family`, `font-series` and `font-shape`.

**font-series** (symbol)

Select the series of a font. Choices include `medium`, `bold`, `bold-narrow`, etc.

**font-shape** (symbol)

Select the shape of a font. Choices include `upright`, `italic`, `caps`.

**font-size** (number)

The font size, compared to the 'normal' size. 0 is style-sheet's normal size, -1 is smaller, +1 is bigger. Each step of 1 is approximately 12% larger; 6 steps are exactly a factor 2 larger. If the context property `fontSize` is set, its value is added to this before the glyph is printed. Fractional values are allowed.

**footnote** (boolean)

Should this be a footnote or in-note?

**footnote-music** (music)

Music creating a footnote.

**footnote-text** (markup)

A footnote for the grob.

**force-hshift** (number)

This specifies a manual shift for notes in collisions. The unit is the note head width of the first voice note. This is used by Sección "note-collision-interface" en *Referencia de Funcionamiento Interno*.

**forced-spacing** (number)

Spacing forced between grobs, used in various ligature engravers.

**fraction** (fraction, as pair)

Numerator and denominator of a time signature object.

**french-beaming** (boolean)

Use French beaming style for this stem. The stem stops at the innermost beams.

**fret-diagram-details** (list)

An alist of detailed grob properties for fret diagrams. Each alist entry consists of a (`property . value`) pair. The properties which can be included in `fret-diagram-details` include the following:

- **barre-type** – Type of barre indication used. Choices include `curved`, `straight`, and `none`. Default `curved`.
- **capo-thickness** – Thickness of capo indicator, in multiples of fret-space. Default value 0.5.

- **dot-color** – Color of dots. Options include **black** and **white**. Default **black**.
- **dot-label-font-mag** – Magnification for font used to label fret dots. Default value 1.
- **dot-position** – Location of dot in fret space. Default 0.6 for dots without labels, 0.95-dot-radius for dots with labels.
- **dot-radius** – Radius of dots, in terms of fret spaces. Default value 0.425 for labeled dots, 0.25 for unlabeled dots.
- **finger-code** – Code for the type of fingering indication used. Options include **none**, **in-dot**, and **below-string**. Default **none** for markup fret diagrams, **below-string** for FretBoards fret diagrams.
- **fret-count** – The number of frets. Default 4.
- **fret-distance** – Multiplier to adjust the distance between frets. Default 1.0.
- **fret-label-custom-format** – The format string to be used label the lowest fret number, when **number-type** equals to **custom**. Default **"~a"**.
- **fret-label-font-mag** – The magnification of the font used to label the lowest fret number. Default 0.5.
- **fret-label-vertical-offset** – The offset of the fret label from the center of the fret in direction parallel to strings. Default 0.
- **fret-label-horizontal-offset** – The offset of the fret label from the center of the fret in direction orthogonal to strings. Default 0.
- **handedness** – Print the fret-diagram left- or right-handed. -1, **LEFT** for left ; 1, **RIGHT** for right. Default **RIGHT**.
- **paren-padding** – The padding for the parenthesis. Default 0.05.
- **label-dir** – Side to which the fret label is attached. -1, **LEFT**, or **DOWN** for left or down; 1, **RIGHT**, or **UP** for right or up. Default **RIGHT**.
- **mute-string** – Character string to be used to indicate muted string. Default **"x"**.
- **number-type** – Type of numbers to use in fret label. Choices include **roman-lower**, **roman-upper**, **arabic** and **custom**. In the later case, the format string is supplied by the **fret-label-custom-format** property. Default **roman-lower**.
- **open-string** – Character string to be used to indicate open string. Default **"o"**.
- **orientation** – Orientation of fret-diagram. Options include **normal**, **landscape**, and **opposing-landscape**. Default **normal**.
- **string-count** – The number of strings. Default 6.
- **string-distance** – Multiplier to adjust the distance between strings. Default 1.0.
- **string-label-font-mag** – The magnification of the font used to label fingerings at the string, rather than in the dot. Default value 0.6 for **normal** orientation, 0.5 for **landscape** and **opposing-landscape**.
- **string-thickness-factor** – Factor for changing thickness of each string in the fret diagram. Thickness of string  $k$  is given by **thickness** \*  $(1 + \text{string-thickness-factor})^{(k-1)}$ . Default 0.
- **top-fret-thickness** – The thickness of the top fret line, as a multiple of the standard thickness. Default value 3.

- **xo-font-magnification** – Magnification used for mute and open string indicators. Default value 0.5.
- **xo-padding** – Padding for open and mute indicators from top fret. Default value 0.25.

**full-length-padding** (number)

How much padding to use at the right side of a full-length tuplet bracket.

**full-length-to-extent** (boolean)

Run to the extent of the column for a full-length tuplet bracket.

**full-measure-extra-space** (number)

Extra space that is allocated at the beginning of a measure with only one note. This property is read from the `NonMusicalPaperColumn` that begins the measure.

**full-size-change** (boolean)

Don't make a change clef smaller.

**gap** (dimension, in staff space)

Size of a gap in a variable symbol.

**gap-count** (integer)

Number of gapped beams for tremolo.

**glissando-skip** (boolean)

Should this `NoteHead` be skipped by glissandi?

**glyph** (string)

A string determining what 'style' of glyph is typeset. Valid choices depend on the function that is reading this property.

In combination with (span) bar lines, it is a string resembling the bar line appearance in ASCII form.

**glyph-name** (string)

The glyph name within the font.

In the context of (span) bar lines, *glyph-name* represents a processed form of **glyph**, where decisions about line breaking etc. are already taken.

**graphical** (boolean)

Display in graphical (vs. text) form.

**grow-direction** (direction)

Crescendo or decrescendo?

**hair-thickness** (number)

Thickness of the thin line in a bar line, expressed as a multiple of the default staff-line thickness (i.e. the visual output is *not* influenced by changes to `Staff.StaffSymbol.thickness`).

**harp-pedal-details** (list)

An alist of detailed grob properties for harp pedal diagrams. Each alist entry consists of a (*property* . *value*) pair. The properties which can be included in harp-pedal-details include the following:

- **box-offset** – Vertical shift of the center of flat/sharp pedal boxes above/below the horizontal line. Default value 0.8.
- **box-width** – Width of each pedal box. Default value 0.4.
- **box-height** – Height of each pedal box. Default value 1.0.

- **space-before-divider** – Space between boxes before the first divider (so that the diagram can be made symmetric). Default value 0.8.
- **space-after-divider** – Space between boxes after the first divider. Default value 0.8.
- **circle-thickness** – Thickness (in unit of the line-thickness) of the ellipse around circled pedals. Default value 0.5.
- **circle-x-padding** – Padding in X direction of the ellipse around circled pedals. Default value 0.15.
- **circle-y-padding** – Padding in Y direction of the ellipse around circled pedals. Default value 0.2.

**head-direction** (direction)

Are the note heads left or right in a semitie?

**height** (dimension, in staff space)

Height of an object in **staff-space** units.

**height-limit** (dimension, in staff space)

Maximum slur height: The longer the slur, the closer it is to this height.

**hide-tied-accidental-after-break** (boolean)

If set, an accidental that appears on a tied note after a line break will not be displayed.

**horizon-padding** (number)

The amount to pad the axis along which a **Skyline** is built for the **side-position-interface**.

**horizontal-shift** (integer)

An integer that identifies ranking of **NoteColumns** for horizontal shifting. This is used by Sección “note-collision-interface” en *Referencia de Funcionamiento Interno*.

**horizontal-skylines** (pair of skylines)

Two skylines, one to the left and one to the right of this grob.

**id** (string)

An id string for the grob.

**ignore-ambitus** (boolean)

If set, don’t consider this notehead for ambitus calculation.

**ignore-collision** (boolean)

If set, don’t do note collision resolution on this **NoteColumn**.

**implicit** (boolean)

Is this an implicit bass figure?

**inspect-quants** (pair of numbers)

If debugging is set, set beam and slur position to a (quantized) position that is as close as possible to this value, and print the demerits for the inspected position in the output.

**keep-inside-line** (boolean)

If set, this column cannot have objects sticking into the margin.

**kern** (dimension, in staff space)

The space between individual elements in any compound bar line, expressed as a multiple of the default staff-line thickness (i.e. the visual output is *not* influenced by changes to **Staff.StaffSymbol.thickness**).

**knee** (boolean)

Is this beam kneed?

**knee-spacing-correction** (number)

Factor for the optical correction amount for kneed beams. Set between 0 for no correction and 1 for full correction.

**knee-to-beam** (boolean)

Determines whether a tuplet number will be positioned next to a kneed beam.

**labels** (list)

List of labels (symbols) placed on a column.

**layer** (integer)

An integer which determines the order of printing objects. Objects with the lowest value of layer are drawn first, then objects with progressively higher values are drawn, so objects with higher values overwrite objects with lower values. By default most objects are assigned a layer value of 1.

**ledger-extra** (dimension, in staff space)

Extra distance from staff line to draw ledger lines for.

**ledger-line-thickness** (pair of numbers)

The thickness of ledger lines. It is the sum of 2 numbers: The first is the factor for line thickness, and the second for staff space. Both contributions are added.

**ledger-positions** (list)

Vertical positions of ledger lines. When set on a **StaffSymbol** grob it defines a repeating pattern of ledger lines and any parenthesized groups will always be shown together.

**ledger-positions-function** (any type)

A quoted Scheme procedure that takes a **StaffSymbol** grob and the vertical position of a note head as arguments and returns a list of ledger line positions.

**left-bound-info** (list)

An alist of properties for determining attachments of spanners to edges.

**left-number-text** (markup)

For a measure counter, this is the formatted measure count. When the measure counter extends over several measures (like with compressed multi-measure rests), it is the text on the left side of the dash.

**left-padding** (dimension, in staff space)

The amount of space that is put left to an object (e.g., a lyric extender).

**length** (dimension, in staff space)

User override for the stem length of unbeamed stems (each unit represents half a **staff-space**).

**length-fraction** (number)

Multiplier for lengths. Used for determining ledger lines and stem lengths.

**line-break-penalty** (number)

Penalty for a line break at this column. This affects the choices of the line breaker; it avoids a line break at a column with a positive penalty and prefers a line break at a column with a negative penalty.

**line-break-permission** (symbol)

Instructs the line breaker on whether to put a line break at this column. Can be **force** or **allow**.

`line-break-system-details` (list)

An alist of properties to use if this column is the start of a system.

`line-count` (integer)

The number of staff lines.

`line-positions` (list)

Vertical positions of staff lines.

`line-thickness` (number)

For slurs and ties, this is the diameter of the virtual “pen” that draws the two arcs of the curve’s outline, which intersect at the endpoints. This property is expressed as a multiple of the current staff-line thickness (i.e. the visual output is influenced by changes to `Staff.StaffSymbol.thickness`).

`long-text` (markup)

Text markup. See Sección “Formatting text” en *Referencia de la Notación*.

`max-beam-connect` (integer)

Maximum number of beams to connect to beams from this stem. Further beams are typeset as beamlets.

`max-symbol-separation` (number)

The maximum distance between symbols making up a church rest.

`maximum-gap` (number)

Maximum value allowed for `gap` property.

`measure-count` (integer)

The number of measures for a multi-measure rest.

`measure-length` (moment)

Length of a measure. Used in some spacing situations.

`merge-differently-dotted` (boolean)

Merge note heads in collisions, even if they have a different number of dots. This is normal notation for some types of polyphonic music.

`merge-differently-dotted` only applies to opposing stem directions (i.e., voice 1 & 2).

`merge-differently-headed` (boolean)

Merge note heads in collisions, even if they have different note heads. The smaller of the two heads is rendered invisible. This is used in polyphonic guitar notation. The value of this setting is used by Sección “note-collision-interface” en *Referencia de Funcionamiento Interno*.

`merge-differently-headed` only applies to opposing stem directions (i.e., voice 1 & 2).

`minimum-distance` (dimension, in staff space)

Minimum distance between rest and notes or beam.

`minimum-length` (dimension, in staff space)

Try to make a spanner at least this long, normally in the horizontal direction. This requires an appropriate callback for the `springs-and-rods` property. If added to a `Tie`, this sets the minimum distance between noteheads.

`minimum-length-after-break` (dimension, in staff space)

If set, try to make a broken spanner starting a line this long. This requires an appropriate callback for the `springs-and-rods` property. If added to a `Tie`, this sets the minimum distance to the notehead.

- minimum-length-fraction** (number)  
Minimum length of ledger line as fraction of note head size.
- minimum-space** (dimension, in staff space)  
Minimum distance that the victim should move (after padding).
- minimum-X-extent** (pair of numbers)  
Minimum size of an object in X dimension, measured in **staff-space** units.
- minimum-Y-extent** (pair of numbers)  
Minimum size of an object in Y dimension, measured in **staff-space** units.
- neutral-direction** (direction)  
Which direction to take in the center of the staff.
- neutral-position** (number)  
Position (in half staff spaces) where to flip the direction of custos stem.
- next** (graphical (layout) object)  
Object that is next relation (e.g., the lyric syllable following an extender).
- no-alignment** (boolean)  
If set, don't place this grob in a **VerticalAlignment**; rather, place it using its own **Y-offset** callback.
- no-ledgers** (boolean)  
If set, don't draw ledger lines on this object.
- no-stem-extend** (boolean)  
If set, notes with ledger lines do not get stems extending to the middle staff line.
- non-break-align-symbols** (list)  
A list of symbols that determine which **NON-break-aligned** interfaces to align this to.
- non-default** (boolean)  
Set for manually specified clefs and keys.
- non-musical** (boolean)  
True if the grob belongs to a **NonMusicalPaperColumn**.
- nonstaff-nonstaff-spacing** (list)  
The spacing alist controlling the distance between the current non-staff line and the next non-staff line in the direction of **staff-affinity**, if both are on the same side of the related staff, and **staff-affinity** is either **UP** or **DOWN**. See **staff-staff-spacing** for a description of the alist structure.
- nonstaff-relatedstaff-spacing** (list)  
The spacing alist controlling the distance between the current non-staff line and the nearest staff in the direction of **staff-affinity**, if there are no non-staff lines between the two, and **staff-affinity** is either **UP** or **DOWN**. If **staff-affinity** is **CENTER**, then **nonstaff-relatedstaff-spacing** is used for the nearest staves on *both* sides, even if other non-staff lines appear between the current one and either of the staves. See **staff-staff-spacing** for a description of the alist structure.
- nonstaff-unrelatedstaff-spacing** (list)  
The spacing alist controlling the distance between the current non-staff line and the nearest staff in the opposite direction from **staff-affinity**, if there are no other non-staff lines between the two, and **staff-affinity** is either **UP** or **DOWN**. See **staff-staff-spacing** for a description of the alist structure.

**normalized-endpoints** (pair)

Represents left and right placement over the total spanner, where the width of the spanner is normalized between 0 and 1.

**note-collision-threshold** (dimension, in staff space)

Simultaneous notes that are this close or closer in units of **staff-space** will be identified as vertically colliding. Used by **Stem** grobs for notes in the same voice, and **NoteCollision** grobs for notes in different voices. Default value 1.

**note-names** (vector)

Vector of strings containing names for easy-notation note heads.

**number-range-separator** (markup)

For a measure counter extending over several measures (like with compressed multi-measure rests), this is the separator between the two printed numbers.

**number-type** (symbol)

Numbering style. Choices include **roman-lower**, **roman-upper** and **arabic**.

**output-attributes** (list)

An alist of attributes for the grob, to be included in output files. When the SVG typesetting backend is used, the attributes are assigned to a group (<g>) containing all of the stencils that comprise a given grob. For example,

```
'((id . 123) (class . foo) (data-whatever . "bar"))
```

produces

```
<g id="123" class="foo" data-whatever="bar"> ... </g>
```

In the Postscript backend, where there is no way to group items, the setting of the **output-attributes** property has no effect.

**outside-staff-horizontal-padding** (number)

By default, an outside-staff-object can be placed so that it is very close to another grob horizontally. If this property is set, the outside-staff-object is raised so that it is not so close to its neighbor.

**outside-staff-padding** (number)

The padding to place between grobs when spacing according to **outside-staff-priority**. Two grobs with different **outside-staff-padding** values have the larger value of padding between them.

**outside-staff-placement-directive** (symbol)

One of four directives telling how outside staff objects should be placed.

- **left-to-right-greedy** – Place each successive grob from left to right.
- **left-to-right-polite** – Place a grob from left to right only if it does not potentially overlap with another grob that has been placed on a pass through a grob array. If there is overlap, do another pass to determine placement.
- **right-to-left-greedy** – Same as **left-to-right-greedy**, but from right to left.
- **right-to-left-polite** – Same as **left-to-right-polite**, but from right to left.

**outside-staff-priority** (number)

If set, the grob is positioned outside the staff in such a way as to avoid all collisions. In case of a potential collision, the grob with the smaller **outside-staff-priority** is closer to the staff.



**packed-spacing** (boolean)

If set, the notes are spaced as tightly as possible.

**padding** (dimension, in staff space)

Add this much extra space between objects that are next to each other.

**padding-pairs** (list)

An alist mapping (*name* . *name*) to distances.

**page-break-penalty** (number)

Penalty for page break at this column. This affects the choices of the page breaker; it avoids a page break at a column with a positive penalty and prefers a page break at a column with a negative penalty.

**page-break-permission** (symbol)

Instructs the page breaker on whether to put a page break at this column. Can be **force** or **allow**.

**page-number** (number)

Page number on which this system ends up.

**page-turn-penalty** (number)

Penalty for a page turn at this column. This affects the choices of the page breaker; it avoids a page turn at a column with a positive penalty and prefers a page turn at a column with a negative penalty.

**page-turn-permission** (symbol)

Instructs the page breaker on whether to put a page turn at this column. Can be **force** or **allow**.

**parent-alignment-X** (number)

Specify on which point of the parent the object is aligned. The value -1 means aligned on parent's left edge, 0 on center, and 1 right edge, in X direction. Other numerical values may also be specified - the unit is half the parent's width. If unset, the value from **self-alignment-X** property will be used.

**parent-alignment-Y** (number)

Like **parent-alignment-X** but for the Y axis.

**parenthesis-friends** (list)

A list of Grob types, as symbols. When parentheses enclose a Grob that has 'parenthesis-friends, the parentheses widen to include any child Grobs with type among 'parenthesis-friends.

**parenthesized** (boolean)

Parenthesize this grob.

**positions** (pair of numbers)

Pair of staff coordinates (*start* . *end*), where *start* and *end* are vertical positions in **staff-space** units of the current staff. For slurs, this value selects which slur candidate to use; if extreme positions are requested, the closest one is taken.

**prefer-dotted-right** (boolean)

For note collisions, prefer to shift dotted up-note to the right, rather than shifting just the dot.

**protrusion** (number)

In an arpeggio bracket, the length of the horizontal edges.

**rank-on-page** (number)

0-based index of the system on a page.

**ratio** (number)

Parameter for slur shape. The higher this number, the quicker the slur attains its **height-limit**.

**remove-empty** (boolean)

If set, remove group if it contains no interesting items.

**remove-first** (boolean)

Remove the first staff of an orchestral score?

**remove-layer** (index or symbol)

When set as a positive integer, the **Keep\_alive\_together\_engraver** removes all **VerticalAxisGroup** grobs with a **remove-layer** larger than the smallest retained **remove-layer**. Set to **#f** to make a layer independent of the **Keep\_alive\_together\_engraver**. Set to **'()**, the layer does not participate in the layering decisions. The property can also be set as a symbol for common behaviors: **#'any** to keep the layer alive with any other layer in the group; **#'above** or **#'below** to keep the layer alive with the context immediately before or after it, respectively.

**replacement-alist** (list)

Alist of strings. The key is a string of the pattern to be replaced. The value is a string of what should be displayed. Useful for ligatures.

**restore-first** (boolean)

Print a natural before the accidental.

**rhythmic-location** (rhythmic location)

Where (bar number, measure position) in the score.

**right-bound-info** (list)

An alist of properties for determining attachments of spanners to edges.

**right-number-text** (markup)

When the measure counter extends over several measures (like with compressed multi-measure rests), this is the text on the right side of the dash. Usually unset.

**right-padding** (dimension, in staff space)

Space to insert on the right side of an object (e.g., between note and its accidentals).

**rotation** (list)

Number of degrees to rotate this object, and what point to rotate around. For example, **'(45 0 0)** rotates by 45 degrees around the center of this object.

**round-up-exceptions** (list)

A list of pairs where car is the numerator and cdr the denominator of a moment. Each pair in this list means that the multi-measure rests of the corresponding length will be rounded up to the longer rest. See *round-up-to-longer-rest*.

**round-up-to-longer-rest** (boolean)

Displays the longer multi-measure rest when the length of a measure is between two values of **usable-duration-logs**. For example, displays a breve instead of a whole in a 3/2 measure.

**rounded** (boolean)

Decide whether lines should be drawn rounded or not.

**same-direction-correction** (number)

Optical correction amount for stems that are placed in tight configurations. This amount is used for stems with the same direction to compensate for note head to stem distance.

**script-priority** (number)

A key for determining the order of scripts in a stack, by being added to the position of the script in the user input, the sum being the overall priority. Smaller means closer to the head.

**segno-kern** (number)

The space between the two thin lines of the segno bar line symbol, expressed as a multiple of the default staff-line thickness (i.e. the visual output is *not* influenced by changes to `Staff.StaffSymbol.thickness`).

**self-alignment-X** (number)

Specify alignment of an object. The value `-1` means left aligned, `0` centered, and `1` right-aligned in X direction. Other numerical values may also be specified - the unit is half the object width.

**self-alignment-Y** (number)

Like **self-alignment-X** but for the Y axis.

**shape** (symbol)

This setting determines what shape a grob has. Valid choices depend on the `stencil` callback reading this property.

**sharp-positions** (list)

Sharps in key signatures are placed within the specified ranges of staff-positions. The general form is a list of pairs, with one pair for each type of clef, in order of the staff-position at which each clef places C: (**alto treble tenor soprano baritone mezzosoprano bass**). If the list contains a single element it applies for all clefs. A single number in place of a pair sets accidentals within the octave ending at that staff-position.

**shorten-pair** (pair of numbers)

The lengths to shorten on both sides a hairpin or text-spanner such as a pedal bracket. Positive values shorten the hairpin or text-spanner, while negative values lengthen it.

**shortest-duration-space** (number)

Start with this multiple of `spacing-increment` space for the shortest duration. See also Sección “spacing-spanner-interface” en *Referencia de Funcionamiento Interno*.

**shortest-playing-duration** (moment)

The duration of the shortest note playing here.

**shortest-starter-duration** (moment)

The duration of the shortest note that starts here.

**show-control-points** (boolean)

For grobs printing Bézier curves, setting this property to true causes the control points and control polygon to be drawn on the page for ease of tweaking.

**side-axis** (number)

If the value is **X** (or equivalently `0`), the object is placed horizontally next to the other object. If the value is **Y** or `1`, it is placed vertically.

**side-relative-direction** (direction)

Multiply direction of `direction-source` with this to get the direction of this object.

**simple-Y** (boolean)

Should the Y placement of a spanner disregard changes in system heights?

**size** (number)

The ratio of the size of the object to its default size.

**skip-quanting** (boolean)

Should beam quanting be skipped?

**skyline-horizontal-padding** (number)

For determining the vertical distance between two staves, it is possible to have a configuration which would result in a tight interleaving of grobs from the top staff and the bottom staff. The larger this parameter is, the farther apart the staves are placed in such a configuration.

**skyline-vertical-padding** (number)

The amount by which the left and right skylines of a column are padded vertically, beyond the **Y-extents** and **extra-spacing-heights** of the constituent grobs in the column. Increase this to prevent interleaving of grobs from adjacent columns.

**slash-negative-kern** (number)

The space to remove between slashes in percent repeat glyphs. Larger values bring the two elements closer together.

**slope** (number)

The slope of this object.

**slur-padding** (number)

Extra distance between slur and script.

**snap-radius** (number)

The maximum distance between two objects that will cause them to snap to alignment along an axis.

**space-alist** (list)

An alist that specifies distances from this grob to other breakable items, using the format:

```
'((break-align-symbol . (spacing-style . space))
  (break-align-symbol . (spacing-style . space))
  ...)
```

Standard choices for *break-align-symbol* are listed in Sección “break-alignment-interface” en *Referencia de Funcionamiento Interno*. Additionally, three special break-align symbols available to **space-alist** are:

**first-note**

used when the grob is just left of the first note on a line

**next-note**

used when the grob is just left of any other note; if not set, the value of **first-note** gets used

**right-edge**

used when the grob is the last item on the line (only compatible with the **extra-space** spacing style)

Choices for *spacing-style* are:

**extra-space**

Put this much space between the two grobs. The space is stretchable when paired with **first-note** or **next-note**; otherwise it is fixed.

**minimum-space**

Put at least this much space between the left sides of both grobs, without allowing them to collide. The space is stretchable when paired with **first-note** or **next-note**; otherwise it is fixed. Not compatible with **right-edge**.

**fixed-space**

Only compatible with **first-note** and **next-note**. Put this much fixed space between the grob and the note.

**minimum-fixed-space**

Only compatible with **first-note** and **next-note**. Put at least this much fixed space between the left side of the grob and the left side of the note, without allowing them to collide.

**semi-fixed-space**

Only compatible with **first-note** and **next-note**. Put this much space between the grob and the note, such that half of the space is fixed and half is stretchable.

Rules for this spacing are much more complicated than this. See [Wanske] page 126–134, [Ross] page 143–147.

**space-to-barline** (boolean)

If set, the distance between a note and the following non-musical column will be measured to the bar line instead of to the beginning of the non-musical column. If there is a clef change followed by a bar line, for example, this means that we will try to space the non-musical column as though the clef is not there.

**spacing-increment** (dimension, in staff space)

The unit of length for note-spacing. Typically, the width of a note head. See also Sección “spacing-spanner-interface” en *Referencia de Funcionamiento Interno*.

**spacing-pair** (pair)

A pair of alignment symbols which set an object’s spacing relative to its left and right **BreakAlignments**.

For example, a **MultiMeasureRest** will ignore prefatory items at its bounds (i.e., clefs, key signatures and time signatures) using the following override:

```
\override MultiMeasureRest.spacing-pair =
    #'(staff-bar . staff-bar)
```

**spanner-id** (index or symbol)

An identifier to distinguish concurrent spanners.

**springs-and-rods** (boolean)

Dummy variable for triggering spacing routines.

**stacking-dir** (direction)

Stack objects in which direction?

**staff-affinity** (direction)

The direction of the staff to use for spacing the current non-staff line. Choices are **UP**, **DOWN**, and **CENTER**. If **CENTER**, the non-staff line will be placed equidistant between the two nearest staves on either side, unless collisions or other spacing constraints prevent this. Setting **staff-affinity** for a staff causes it to be treated as a non-staff line. Setting **staff-affinity** to **#f** causes a non-staff line to be treated as a staff.

**staff-padding** (dimension, in staff space)

Maintain this much space between reference points and the staff. Its effect is to align objects of differing sizes (like the dynamics **p** and **f**) on their baselines.

**staff-position** (number)

Vertical position, measured in half staff spaces, counted from the middle line.

**staff-space** (dimension, in staff space)

Amount of space between staff lines, expressed in global **staff-space**.

**staff-staff-spacing** (list)

When applied to a staff-group's **StaffGrouper** grob, this spacing alist controls the distance between consecutive staves within the staff-group. When applied to a staff's **VerticalAxisGroup** grob, it controls the distance between the staff and the nearest staff below it in the same system, replacing any settings inherited from the **StaffGrouper** grob of the containing staff-group, if there is one. This property remains in effect even when non-staff lines appear between staves. The alist can contain the following keys:

- **basic-distance** – the vertical distance, measured in staff-spaces, between the reference points of the two items when no collisions would result, and no stretching or compressing is in effect.
- **minimum-distance** – the smallest allowable vertical distance, measured in staff-spaces, between the reference points of the two items, when compressing is in effect.
- **padding** – the minimum required amount of unobstructed vertical whitespace between the bounding boxes (or skylines) of the two items, measured in staff-spaces.
- **stretchability** – a unitless measure of the dimension's relative propensity to stretch. If zero, the distance will not stretch (unless collisions would result).

**staffgroup-staff-spacing** (list)

The spacing alist controlling the distance between the last staff of the current staff-group and the staff just below it in the same system, even if one or more non-staff lines exist between the two staves. If the **staff-staff-spacing** property of the staff's **VerticalAxisGroup** grob is set, that is used instead. See **staff-staff-spacing** for a description of the alist structure.

**stem-attachment** (pair of numbers)

An (x . y) pair where the stem attaches to the notehead.

**stem-begin-position** (number)

User override for the begin position of a stem.

**stem-spacing-correction** (number)

Optical correction amount for stems that are placed in tight configurations. For opposite directions, this amount is the correction for two normal sized stems that overlap completely.

**stemlet-length** (number)

How long should be a stem over a rest?

**stencil** (stencil)

The symbol to print.

**stencils** (list)

Multiple stencils, used as intermediate value.

**strict-grace-spacing** (boolean)

If set, main notes are spaced normally, then grace notes are put left of the musical columns for the main notes.

**strict-note-spacing** (boolean)

If set, unbroken columns with non-musical material (clefs, bar lines, etc.) are not spaced separately, but put before musical columns.

**stroke-style** (string)

Set to "grace" to turn stroke through flag on.

**style** (symbol)

This setting determines in what style a grob is typeset. Valid choices depend on the **stencil** callback reading this property.

**text** (markup)

Text markup. See Sección "Formatting text" en *Referencia de la Notación*.

**text-direction** (direction)

This controls the ordering of the words. The default **RIGHT** is for roman text. Arabic or Hebrew should use **LEFT**.

**thick-thickness** (number)

Thickness of the thick line in a bar line, expressed as a multiple of the default staff-line thickness (i.e. the visual output is *not* influenced by changes to **Staff.StaffSymbol.thickness**).

**thickness** (number)

For grobs made up of lines, this is the thickness of the line. For slurs and ties, this is the distance between the two arcs of the curve's outline at its thickest point, not counting the diameter of the virtual "pen" that draws the arcs. This property is expressed as a multiple of the current staff-line thickness (i.e. the visual output is influenced by changes to **Staff.StaffSymbol.thickness**).

**tie-configuration** (list)

List of (*position* . *dir*) pairs, indicating the desired tie configuration, where *position* is the offset from the center of the staff in staff space and *dir* indicates the direction of the tie (1=>up, -1=>down, 0=>center). A non-pair entry in the list causes the corresponding tie to be formatted automatically.

**to-barline** (boolean)

If true, the spanner will stop at the bar line just before it would otherwise stop.

**toward-stem-shift** (number)

Amount by which scripts are shifted toward the stem if their direction coincides with the stem direction. 0.0 means centered on the note head (the default position of most scripts); 1.0 means centered on the stem. Interpolated values are possible.

**toward-stem-shift-in-column** (number)

Amount by which a script is shifted toward the stem if its direction coincides with the stem direction and it is associated with a **ScriptColumn** object. 0.0 means centered on the note head (the default position of most scripts); 1.0 means centered on the stem. Interpolated values are possible.

**transparent** (boolean)

This makes the grob invisible.

**tuplet-slur** (boolean)

Draw a slur instead of a bracket for tuplets.

**uniform-stretching** (boolean)

If set, items stretch proportionally to their natural separation based on durations. This looks better in complex polyphonic patterns.

**usable-duration-logs** (list)

List of **duration-logs** that can be used in typesetting the grob.

**use-skylines** (boolean)

Should skylines be used for side positioning?

**used** (boolean)

If set, this spacing column is kept in the spacing problem.

**vertical-skylines** (pair of skylines)

Two skylines, one above and one below this grob.

**voiced-position** (number)

The staff-position of a voiced **Rest**, negative if the rest has **direction** DOWN.

**when** (moment)

Global time step associated with this column.

**whiteout** (boolean-or-number)

If a number or true, the grob is printed over a white background to white-out underlying material, if the grob is visible. A number indicates how far the white background extends beyond the bounding box of the grob as a multiple of the staff-line thickness. The **LyricHyphen** grob uses a special implementation of whiteout: A positive number indicates how far the white background extends beyond the bounding box in multiples of **line-thickness**. The shape of the background is determined by **whiteout-style**. Usually **#f** by default.

**whiteout-style** (symbol)

Determines the shape of the **whiteout** background. Available are 'outline, 'rounded-box, and the default 'box. There is one exception: Use 'special for **LyricHyphen**.

**width** (dimension, in staff space)

The width of a grob measured in staff space.

**word-space** (dimension, in staff space)

Space to insert between words in texts.

**X-align-on-main-noteheads** (boolean)

If true, this grob will ignore suspended noteheads when aligning itself on **NoteColumn**.

**X-extent** (pair of numbers)

Extent (size) in the X direction, measured in staff-space units, relative to object's reference point.

**X-offset** (number)

The horizontal amount that this object is moved relative to its X-parent.

**X-positions** (pair of numbers)

Pair of X staff coordinates of a spanner in the form (*left* . *right*), where both *left* and *right* are in **staff-space** units of the current staff.

**Y-extent** (pair of numbers)

Extent (size) in the Y direction, measured in staff-space units, relative to object's reference point.



**Y-offset** (number)

The vertical amount that this object is moved relative to its Y-parent.

**zigzag-length** (dimension, in staff space)

The length of the lines of a zigzag, relative to **zigzag-width**. A value of 1 gives 60-degree zigzags.

**zigzag-width** (dimension, in staff space)

The width of one zigzag squiggle. This number is adjusted slightly so that the spanner line can be constructed from a whole number of squiggles.

## A.19 Funciones musicales disponibles

**\absolute** [music] - *music* (music)

Make *music* absolute. This does not actually change the music itself but rather hides it from surrounding **\relative** and **\fixed** commands.

**\acciaccatura** [music] - *music* (music)

Create an acciaccatura from the following music expression

**\accidentalStyle** [music] - *style* (symbol list)

Set accidental style to symbol list *style* in the form ‘**piano-cautionary**’. If *style* has a form like ‘**Staff.piano-cautionary**’, the settings are applied to that context. Otherwise, the context defaults to ‘**Staff**’, except for piano styles, which use ‘**GrandStaff**’ as a context.

**\addChordShape** [void] - *key-symbol* (symbol) *tuning* (pair) *shape-definition* (string or pair)

Add chord shape *shape-definition* to the *chord-shape-table* hash with the key (**cons** *key-symbol tuning*).

**\addInstrumentDefinition** [void] - *name* (string) *lst* (list)

Create instrument *name* with properties *list*.

**\addQuote** [void] - *name* (string) *music* (music)

Define *music* as a quotable music expression named *name*

**\afterGrace** [music] - *fraction* [non-negative rational, fraction, or moment] *main* (music) *grace* (music)

Create *grace* note(s) after a *main* music expression.

The musical position of the grace expression is after a given fraction of the main note’s duration has passed. If *fraction* is not specified as first argument, it is taken from **afterGraceFraction** which has a default value of 3/4.

**\allowPageTurn** [music]

Allow a page turn. May be used at toplevel (ie between scores or markups), or inside a score.

**\allowVoltaHook** [void] - *bar* (string)

Allow the volta bracket hook being drawn over bar line *bar*.

**\alterBroken** [music] - *property* (key list or symbol) *arg* (list) *item* (key list or music)

Override *property* for pieces of broken spanner *item* with values *arg*. *item* may either be music in the form of a starting spanner event, or a symbol list in the form ‘**Context.Grob**’ or just ‘**Grob**’. If *item* is in the form of a spanner event, *property* may also have the form ‘**Grob.property**’ for specifying a directed tweak.

**\ambitusAfter** [music] - *target* (symbol)

Move the ambitus after the break-align symbol *target*.

- `\appendToTag` [music] - *tag* (symbol) *more* (music) *music* (music)  
Append *more* to the back of music tagged with *tag*. A **post-event** can be added to the articulations of rhythmic events or chords; other expressions may be added to chords, sequential or simultaneous music.
- `\applyContext` [music] - *proc* (procedure)  
Modify context properties with Scheme procedure *proc*.
- `\applyMusic` [music] - *func* (procedure) *music* (music)  
Apply procedure *func* to *music*.
- `\applyOutput` [music] - *target* (symbol list or symbol) *proc* (procedure)  
Apply function *proc* to every layout object matched by *target* which takes the form **Context** or **Context.Grob**.
- `\appoggiatura` [music] - *music* (music)  
Create an appoggiatura from *music*
- `\assertBeamQuant` [music] - *l* (pair) *r* (pair)  
Testing function: check whether the beam quants *l* and *r* are correct
- `\assertBeamSlope` [music] - *comp* (procedure)  
Testing function: check whether the slope of the beam is the same as *comp*
- `\autoChange` [music] - *pitch* [pitch] *clef-1* [context modification] *clef-2* [context modification] *music* (music)  
Make voices that switch between staves automatically. As an option the pitch where to switch staves may be specified. The clefs for the staves are optional as well. Setting clefs works only for implicitly instantiated staves.
- `\balloonGrobText` [music] - *grob-name* (symbol) *offset* (pair of numbers) *text* (markup)  
Attach *text* to *grob-name* at offset *offset* (use like `\once`)
- `\balloonText` [post event] - *offset* (pair of numbers) *text* (markup)  
Attach *text* at *offset* (use like `\tweak`)
- `\bar` [music] - *type* (string)  
Insert a bar line of type *type*
- `\barNumberCheck` [music] - *n* (integer)  
Print a warning if the current bar number is not *n*.
- `\beamExceptions` (any type) - *music* (music)  
Extract a value suitable for setting **Timing.beamExceptions** from the given pattern with explicit beams in *music*. A bar check | has to be used between bars of patterns in order to reset the timing.
- `\bendAfter` [post event] - *delta* (real number)  
Create a fall or doit of pitch interval *delta*.
- `\bendHold` [post event] - *mus* (music)  
Sets the **'style** of a **BendSpanner** to **'hold**.
- `\bendStartLevel` [post event] - *idx* (non-negative integer) *mus* (music)  
Sets the **details.successive-level** of a **BendSpanner** to *idx*.
- `\bookOutputName` [void] - *newfilename* (string)  
Direct output for the current book block to *newfilename*.
- `\bookOutputSuffix` [void] - *newsuffix* (string)  
Set the output filename suffix for the current book block to *newsuffix*.

- `\breathe` [music]  
 Insert a breath mark.
- `\chordRepeats` [music] - *event-types* [list] *music* (music)  
 Walk through *music* putting the notes of the previous chord into repeat chords, as well as an optional list of *event-types* such as `#'(string-number-event)`.
- `\clef` [music] - *type* (string)  
 Set the current clef to *type*.
- `\compoundMeter` [music] - *args* (pair)  
 Create compound time signatures. The argument is a Scheme list of lists. Each list describes one fraction, with the last entry being the denominator, while the first entries describe the summands in the numerator. If the time signature consists of just one fraction, the list can be given directly, i.e. not as a list containing a single list. For example, a time signature of  $(3+1)/8 + 2/4$  would be created as `\compoundMeter #'((3 1 8) (2 4))`, and a time signature of  $(3+2)/8$  as `\compoundMeter #'((3 2 8))` or shorter `\compoundMeter #'(3 2 8)`.
- `\compressMMRests` [music] - *music* (music)  
 Remove the empty bars created by multi-measure rests, leaving just the first bar containing the MM rest itself.
- `\crossStaff` [music] - *notes* (music)  
 Create cross-staff stems
- `\cueClef` [music] - *type* (string)  
 Set the current cue clef to *type*.
- `\cueClefUnset` [music]  
 Unset the current cue clef.
- `\cueDuring` [music] - *what* (string) *dir* (direction) *main-music* (music)  
 Insert contents of quote *what* corresponding to *main-music*, in a CueVoice oriented by *dir*.
- `\cueDuringWithClef` [music] - *what* (string) *dir* (direction) *clef* (string) *main-music* (music)  
 Insert contents of quote *what* corresponding to *main-music*, in a CueVoice oriented by *dir*.
- `\deadNote` [music] - *note* (music)  
 Print *note* with a cross-shaped note head.
- `\defineBarLine` [void] - *bar* (string) *glyph-list* (list)  
 Define bar line settings for bar line *bar*. The list *glyph-list* must have three entries which define the appearance at the end of line, at the beginning of the next line, and the span bar, respectively.
- `\displayLilyMusic` [music] - *port* [output port] *music* (music)  
 Display the LilyPond input representation of *music* to *port*, defaulting to the console.
- `\displayMusic` [music] - *port* [output port] *music* (music)  
 Display the internal representation of *music* to *port*, default to the console.
- `\displayScheme` (any type) - *port* [output port] *expr* (any type)  
 Display the internal representation of *expr* to *port*, default to the console.
- `\dropNote` [music] - *num* (integer) *music* (music)  
 Drop a note of any chords in *music*, in *num* position from above.

`\endSpanners` [music] - *music* (music)

Terminate the next spanner prematurely after exactly one note without the need of a specific end spanner.

`\eventChords` [music] - *music* (music)

Compatibility function wrapping `EventChord` around isolated rhythmic events occurring since version 2.15.28, after expanding repeat chords ‘q’.

`\featherDurations` [music] - *factor* (moment) *argument* (music)

Adjust durations of music in *argument* by rational *factor*.

`\finger` [post event] - *finger* (integer or markup)

Apply *finger* as a fingering indication.

`\fixed` [music] - *pitch* (pitch) *music* (music)

Use the octave of *pitch* as the default octave for *music*.

`\footnote` [music] - *mark* [markup] *offset* (pair of numbers) *footnote* (markup) *item* (symbol list or music)

Make the markup *footnote* a footnote on *item*. The footnote is marked with a markup *mark* moved by *offset* with respect to the marked music.

If *mark* is not given or specified as `\default`, it is replaced by an automatically generated sequence number. If *item* is a symbol list of form ‘`Grob`’ or ‘`Context.Grob`’, then grobs of that type will be marked at the current time step in the given context (default `Bottom`).

If *item* is music, the music will get a footnote attached to a grob immediately attached to the event, like `\tweak` does. For attaching a footnote to an *indirectly* caused grob, write `\single\footnote`, use *item* to specify the grob, and follow it with the music to annotate.

Like with `\tweak`, if you use a footnote on a following post-event, the `\footnote` command itself needs to be attached to the preceding note or rest as a post-event with `-`.

`\grace` [music] - *music* (music)

Insert *music* as grace notes.

`\grobdescriptions` (any type) - *descriptions* (list)

Create a context modification from *descriptions*, a list in the format of `all-grob-descriptions`.

`\harmonicByFret` [music] - *fret* (number) *music* (music)

Convert *music* into mixed harmonics; the resulting notes resemble harmonics played on a fretted instrument by touching the strings at *fret*.

`\harmonicByRatio` [music] - *ratio* (number) *music* (music)

Convert *music* into mixed harmonics; the resulting notes resemble harmonics played on a fretted instrument by touching the strings at the point given through *ratio*.

`\harmonicNote` [music] - *note* (music)

Print *note* with a diamond-shaped note head.

`\harmonicsOn` [music]

Set the default note head style to a diamond-shaped style.

`\hide` [music] - *item* (symbol list or music)

Set *item*’s ‘`transparent`’ property to `#t`, making it invisible while still retaining its dimensions.

If *item* is a symbol list of form **GrobName** or **Context.GrobName**, the result is an override for the grob name specified by it. If *item* is a music expression, the result is the same music expression with an appropriate tweak applied to it.

**\incipit** [*music*] - *incipit-music* (*music*)

Output *incipit-music* before the main staff as an indication of its appearance in the original music.

**\inherit-acceptability** [void] - *to* (symbol) *from* (symbol)

When used in an output definition, will modify all context definitions such that context *to* is accepted as a child by all contexts that also accept *from*.

**\inStaffSegno** [*music*]

Put the segno variant 'varsegno' at this position into the staff, compatible with the repeat command.

**\instrumentSwitch** [*music*] - *name* (string)

Switch instrument to *name*, which must have been predefined with function **\addInstrumentDefinition**.

**\inversion** [*music*] - *around* (pitch) *to* (pitch) *music* (*music*)

Invert *music* about *around* and transpose from *around* to *to*.

**\invertChords** [*music*] - *num* (integer) *music* (*music*)

Invert any chords in *music* into their *num*-th position. (Chord inversions may be directed downwards using negative integers.)

**\keepWithTag** [*music*] - *tags* (symbol list or symbol) *music* (*music*)

Include only elements of *music* that are tagged with one of the tags in *tags*. *tags* may be either a single symbol or a list of symbols.

Each tag may be declared as a member of at most one tag group (defined with **\tagGroup**). If none of a *music* element's tags share a tag group with one of the specified *tags*, the element is retained.

**\key** [*music*] - *tonic* [pitch] *pitch-alist* [list of number pairs]

Set key to *tonic* and scale *pitch-alist*. If both are null, just generate **KeyChangeEvent**.

**\killCues** [*music*] - *music* (*music*)

Remove cue notes from *music*.

**\label** [*music*] - *label* (symbol)

Create *label* as a referable label.

**\language** [void] - *language* (string)

Set note names for language *language*.

**\languageRestore** [void]

Restore a previously-saved pitchnames alist.

**\languageSaveAndChange** [void] - *language* (string)

Store the previous pitchnames alist, and set a new one.

**\magnifyMusic** [*music*] - *mag* (positive number) *music* (*music*)

Magnify the notation of *music* without changing the staff-size, using *mag* as a size factor. Stems, beams, slurs, ties, and horizontal spacing are adjusted automatically.

**\magnifyStaff** [*music*] - *mag* (positive number)

Change the size of the staff, adjusting notation size and horizontal spacing automatically, using *mag* as a size factor.

`\makeClusters` [music] - *arg* (music)

Display chords in *arg* as clusters.

`\makeDefaultStringTuning` [void] - *symbol* (symbol) *pitches* (list)

This defines a string tuning *symbol* via a list of *pitches*. The *symbol* also gets registered in `defaultStringTunings` for documentation purposes.

`\mark` [music] - *label* [integer or markup]

Make the music for the `\mark` command.

`\markupMap` [music] - *path* (symbol list or symbol) *markupfun* (markup-function) *music* (music)

This applies the given markup function *markupfun* to all markup music properties matching *path* in *music*.

For example,

```
\new Voice { g'2 c'' }
\addlyrics {
  \markupMap LyricEvent.text
    \markup \with-color #red \etc
    { Oh yes! }
}
```

`\modalInversion` [music] - *around* (pitch) *to* (pitch) *scale* (music) *music* (music)

Invert *music* about *around* using *scale* and transpose from *around* to *to*.

`\modalTranspose` [music] - *from* (pitch) *to* (pitch) *scale* (music) *music* (music)

Transpose *music* from pitch *from* to pitch *to* using *scale*.

`\musicMap` [music] - *proc* (procedure) *mus* (music)

Apply *proc* to *mus* and all of the music it contains.

`\noPageBreak` [music]

Forbid a page break. May be used at toplevel (i.e., between scores or markups), or inside a score.

`\noPageTurn` [music]

Forbid a page turn. May be used at toplevel (i.e., between scores or markups), or inside a score.

`\octaveCheck` [music] - *pitch* (pitch)

Octave check.

`\offset` [music] - *property* (symbol list or symbol) *offsets* (any type) *item* (key list or music)

Offset the default value of *property* of *item* by *offsets*. If *item* is a string, the result is `\override` for the specified grob type. If *item* is a music expression, the result is the same music expression with an appropriate tweak applied.

`\omit` [music] - *item* (symbol list or music)

Set *item*'s 'stencil' property to `#f`, effectively omitting it without taking up space. If *item* is a symbol list of form `GrobName` or `Context.GrobName`, the result is an override for the grob name specified by it. If *item* is a music expression, the result is the same music expression with an appropriate tweak applied to it.

`\once` [music] - *music* (music)

Set *once* to `#t` on all layout instruction events in *music*. This will complain about music with an actual duration. As a special exception, if *music* might be the result of a `\tweak` command, no warning will be given in order to allow for `\once \propertyTweak` to work as both one-time override and proper tweak.

`\ottava` [music] - *octave* (integer)  
Set the octavation.

`\overrideProperty` [music] - *grob-property-path* (list of indexes or symbols) *value* (any type)  
Set the grob property specified by *grob-property-path* to *value*. *grob-property-path* is a symbol list of the form `Context.GrobName.property` or `GrobName.property`, possibly with subproperties given as well.

As opposed to `\override` which overrides the context-dependent defaults with which a grob is created, this command uses `Output_property_engraver` at the grob acknowledgement stage. This may be necessary for overriding values set after the initial grob creation.

`\overrideTimeSignatureSettings` [music] - *time-signature* (fraction, as pair) *base-moment* (fraction, as pair) *beat-structure* (list) *beam-exceptions* (list)  
Override `timeSignatureSettings` for time signatures of *time-signature* to have settings of *base-moment*, *beat-structure*, and *beam-exceptions*.

`\pageBreak` [music]  
Force a page break. May be used at toplevel (i.e., between scores or markups), or inside a score.

`\pageTurn` [music]  
Force a page turn between two scores or top-level markups.

`\palmMute` [music] - *note* (music)  
Print *note* with a triangle-shaped note head.

`\palmMuteOn` [music]  
Set the default note head style to a triangle-shaped style.

`\parallelMusic` [void] - *voice-ids* (list) *music* (music)  
Define parallel music sequences, separated by '|' (bar check signs), and assign them to the identifiers provided in *voice-ids*.  
*voice-ids*: a list of music identifiers (symbols containing only letters)  
*music*: a music sequence, containing BarChecks as limiting expressions.  
Example:

```
\parallelMusic A,B,C {
  c c | d d | e e |
  d d | e e | f f |
}
<==>
A = { c c | d d }
B = { d d | e e }
C = { e e | f f }
```

The last bar checks in a sequence are not copied to the result in order to facilitate ending the last entry at non-bar boundaries.

`\parenthesize` [music] - *arg* (music)  
Tag *arg* to be parenthesized.

`\partCombine` [music] - *chord-range* [pair of numbers] *part1* (music) *part2* (music)  
Take the music in *part1* and *part2* and return a music expression containing simultaneous voices, where *part1* and *part2* are combined into one voice where appropriate. Optional *chord-range* sets the distance in steps between notes that may be combined into a chord or unison.

- `\partCombineDown` [music] - *chord-range* [pair of numbers] *part1* (music) *part2* (music)  
Take the music in *part1* and *part2* and typeset so that they share a staff with stems directed downward.
- `\partCombineForce` [music] - *type* [symbol]  
Override the part-combiner.
- `\partCombineUp` [music] - *chord-range* [pair of numbers] *part1* (music) *part2* (music)  
Take the music in *part1* and *part2* and typeset so that they share a staff with stems directed upward.
- `\partial` [music] - *dur* (duration)  
Make a partial measure.
- `\phrasingSlurDashPattern` [music] - *dash-fraction* (number) *dash-period* (number)  
Set up a custom style of dash pattern for *dash-fraction* ratio of line to space repeated at *dash-period* interval for phrasing slurs.
- `\pitchedTrill` [music] - *main-note* (music) *secondary-note* (music)  
Print a trill with *main-note* as the main note of the trill and print *secondary-note* as a stemless note head in parentheses.
- `\pointAndClickOff` [void]  
Suppress generating extra code in final-format (e.g. pdf) files to point back to the lilypond source statement.
- `\pointAndClickOn` [void]  
Enable generation of code in final-format (e.g. pdf) files to reference the originating lilypond source statement; this is helpful when developing a score but generates bigger final-format files.
- `\pointAndClickTypes` [void] - *types* (symbol list or symbol)  
Set a type or list of types (such as `#'note-event`) for which point-and-click info is generated.
- `\preBend` [post event] - *mus* (music)  
Sets the `'style` of a `BendSpanner` to `'pre-bend`.
- `\preBendHold` [post event] - *mus* (music)  
Sets the `'style` of a `BendSpanner` to `'pre-bend-hold`.
- `\propertyOverride` [music] - *grob-property-path* (list of indexes or symbols) *value* (any type)  
Set the grob property specified by *grob-property-path* to *value*. *grob-property-path* is a symbol list of the form `Context.GrobName.property` or `GrobName.property`, possibly with subproperties given as well. This music function is mostly intended for use from Scheme as a substitute for the built-in `\override` command.
- `\propertyRevert` [music] - *grob-property-path* (list of indexes or symbols)  
Revert the grob property specified by *grob-property-path* to its previous value. *grob-property-path* is a symbol list of the form `Context.GrobName.property` or `GrobName.property`, possibly with subproperties given as well. This music function is mostly intended for use from Scheme as a substitute for the built-in `\revert` command.
- `\propertySet` [music] - *property-path* (symbol list or symbol) *value* (any type)  
Set the context property specified by *property-path* to *value*. This music function is mostly intended for use from Scheme as a substitute for the built-in `\set` command.
- `\propertyTweak` [music] - *prop* (key list or symbol) *value* (any type) *item* (key list or music)  
Add a tweak to the following *item*, usually music. This generally behaves like `\tweak` but will turn into an `\override` when *item* is a symbol list.



In that case, *item* specifies the grob path to override. This is mainly useful when using `\propertyTweak` as a component for building other functions like `\omit`. It is not the default behavior for `\tweak` since many input strings in `\lyricmode` can serve equally as music or as symbols which causes surprising behavior when tweaking lyrics using the less specific semantics of `\propertyTweak`.

*prop* can contain additional elements in which case a nested property (inside of an alist) is tweaked.

`\propertyUnset` [music] - *property-path* (symbol list or symbol)

Unset the context property specified by *property-path*. This music function is mostly intended for use from Scheme as a substitute for the built-in `\unset` command.

`\pushToTag` [music] - *tag* (symbol) *more* (music) *music* (music)

Add *more* to the front of music tagged with *tag*. A **post-event** can be added to the articulations of rhythmic events or chords; other expressions may be added to chords, sequential or simultaneous music.

`\quoteDuring` [music] - *what* (string) *main-music* (music)

Indicate a section of music to be quoted. *what* indicates the name of the quoted voice, as specified in an `\addQuote` command. *main-music* is used to indicate the length of music to be quoted; usually contains spacers or multi-measure rests.

`\raiseNote` [music] - *num* (integer) *music* (music)

Raise a note of any chords in *music*, in *num* position from below.

`\reduceChords` [music] - *music* (music)

Reduce chords contained in *music* to single notes, intended mainly for reusing music in RhythmicStaff. Does not reduce parallel music.

`\relative` [music] - *pitch* [pitch] *music* (music)

Make *music* relative to *pitch*. If *pitch* is omitted, the first note in *music* is given in absolute pitch.

`\removeWithTag` [music] - *tags* (symbol list or symbol) *music* (music)

Remove elements of *music* that are tagged with one of the tags in *tags*. *tags* may be either a single symbol or a list of symbols.

`\resetRelativeOctave` [music] - *pitch* (pitch)

Set the octave inside a `\relative` section.

`\retrograde` [music] - *music* (music)

Return *music* in reverse order.

`\revertTimeSignatureSettings` [music] - *time-signature* (pair)

Revert `timeSignatureSettings` for time signatures of *time-signature*.

`\rightHandFinger` [post event] - *finger* (integer or markup)

Apply *finger* as a fingering indication.

`\scaleDurations` [music] - *fraction* (non-negative rational, fraction, or moment) *music* (music)

Multiply the duration of events in *music* by *fraction*.

`\settingsFrom` (any type) - *ctx* [symbol] *music* (music)

Take the layout instruction events from *music*, optionally restricted to those applying to context type *ctx*, and return a context modification duplicating their effect.

`\shape` [music] - *offsets* (list) *item* (key list or music)

Offset control-points of *item* by *offsets*. The argument is a list of number pairs or list of such lists. Each element of a pair represents an offset to one of the coordinates

of a control-point. The y-coordinate of each number pair is scaled by staff space. If *item* is a string, the result is `\once\override` for the specified grob type. If *item* is a music expression, the result is the same music expression with an appropriate tweak applied.

`\shiftDurations` [music] - *dur* (integer) *dots* (integer) *arg* (music)

Change the duration of *arg* by adding *dur* to the `durlog` of *arg* and *dots* to the `dots` of *arg*.

`\single` [music] - *overrides* (music) *music* (music)

Convert *overrides* to tweaks and apply them to *music*. This does not convert `\revert`, `\set` or `\unset`.

`\skip` [music] - *dur* (duration)

Skip forward by *dur*.

`\slashedGrace` [music] - *music* (music)

Create slashed graces (slashes through stems, but no slur) from the following music expression

`\slurDashPattern` [music] - *dash-fraction* (number) *dash-period* (number)

Set up a custom style of dash pattern for *dash-fraction* ratio of line to space repeated at *dash-period* interval for slurs.

`\spacingTweaks` [music] - *parameters* (list)

Set the system stretch, by reading the 'system-stretch' property of the 'parameters' assoc list.

`\storePredefinedDiagram` [void] - *fretboard-table* (hash table) *chord* (music) *tuning* (pair)  
*diagram-definition* (string or pair)

Add predefined fret diagram defined by *diagram-definition* for the chord pitches *chord* and the stringTuning *tuning*.

`\stringTuning` (any type) - *chord* (music)

Convert *chord* to a string tuning. *chord* must be in absolute pitches and should have the highest string number (generally the lowest pitch) first.

`\styledNoteHeads` [music] - *style* (symbol) *heads* (symbol list or symbol) *music* (music)

Set *heads* in *music* to *style*.

`\tabChordRepeats` [music] - *event-types* [list] *music* (music)

Walk through *music* putting the notes, fingerings and string numbers of the previous chord into repeat chords, as well as an optional list of *event-types* such as `#'(articulation-event)`.

`\tabChordRepetition` [void]

Include the string and fingering information in a chord repetition. This function is deprecated; try using `\tabChordRepeats` instead.

`\tag` [music] - *tags* (symbol list or symbol) *music* (music)

Tag the following *music* with *tags* and return the result, by adding the single symbol or symbol list *tags* to the `tags` property of *music*.

`\tagGroup` [void] - *tags* (symbol list)

Define a tag group comprising the symbols in the symbol list *tags*. Tag groups must not overlap.

`\temporary` [music] - *music* (music)

Make any `\override` in *music* replace an existing grob property value only temporarily, restoring the old value when a corresponding `\revert` is executed. This is achieved by clearing the 'pop-first' property normally set on `\overrides`.

An `\override/\revert` sequence created by using `\temporary` and `\undo` on the same music containing overrides will cancel out perfectly or cause a warning.

Non-property-related music is ignored, warnings are generated for any property-changing music that isn't an `\override`.

`\tieDashPattern` [music] - *dash-fraction* (number) *dash-period* (number)

Set up a custom style of dash pattern for *dash-fraction* ratio of line to space repeated at *dash-period* interval for ties.

`\time` [music] - *beat-structure* [number list] *fraction* (fraction, as pair)

Set *fraction* as time signature, with optional number list *beat-structure* before it.

`\times` [music] - *fraction* (fraction, as pair) *music* (music)

Scale *music* in time by *fraction*.

`\tocItem` [music] - *label* [symbol list or symbol] *text* (markup)

Add a line to the table of contents, using the `\tocItemMarkup` paper variable markup and assigning it to *label* if one is provided. If a hierarchy of labels is given, make the current item a child of the corresponding objects.

`\transpose` [music] - *from* (pitch) *to* (pitch) *music* (music)

Transpose *music* from pitch *from* to pitch *to*.

`\transposedCueDuring` [music] - *what* (string) *dir* (direction) *pitch* (pitch) *main-music* (music)

Insert notes from the part *what* into a voice called *cue*, using the transposition defined by *pitch*. This happens simultaneously with *main-music*, which is usually a rest. The argument *dir* determines whether the cue notes should be notated as a first or second voice.

`\transposition` [music] - *pitch* (pitch)

Set instrument transposition

`\tuplet` [music] - *ratio* (fraction, as pair) *tuplet-span* [duration] *music* (music)

Scale the given *music* to tuplets. *ratio* is a fraction that specifies how many notes are played in place of the nominal value: it will be '3/2' for triplets, namely three notes being played in place of two. If the optional duration *tuplet-span* is specified, it is used instead of `\tupletSpannerDuration` for grouping the tuplets. For example,

```
\tuplet 3/2 4 { c8 c c c c c }
```

will result in two groups of three tuplets, each group lasting for a quarter note.

`\tupletSpan` [music] - *tuplet-span* [duration]

Set `\tupletSpannerDuration`, the length into which `\tuplet` without an explicit 'tuplet-span' argument of its own will group its tuplets, to the duration *tuplet-span*. To revert to the default of not subdividing the contents of a `\tuplet` command without explicit 'tuplet-span', use

```
\tupletSpan \default
```

`\tweak` [music] - *prop* (key list or symbol) *value* (any type) *music* (music)

Add a tweak to the following *music*. Layout objects created by *music* get their property *prop* set to *value*. If *prop* has the form 'Grob.property', like with

```
\tweak Accidental.color #red cis'
```

an indirectly created grob ('Accidental' is caused by 'NoteHead') can be tweaked; otherwise only directly created grobs are affected.

*prop* can contain additional elements in which case a nested property (inside of an alist) is tweaked.

If *music* is an 'event-chord', every contained 'rhythmic-event' is tweaked instead.

**\undo** [music] - *music* (music)

Convert **\override** and **\set** in *music* to **\revert** and **\unset**, respectively. Any reverts and unsets already in *music* cause a warning. Non-property-related music is ignored.

**\unfolded** [music] - *music* (music)

Mask *music* until the innermost enclosing repeat is unfolded.

**\unfoldRepeats** [music] - *types* [symbol list or symbol] *music* (music)

Force **\repeat volta**, **\repeat tremolo** or **\repeat percent** commands in *music* to be interpreted as **\repeat unfold**, if specified in the optional symbol-list *types*. The default for *types* is an empty list, which will force any of those commands in *music* to be interpreted as **\repeat unfold**. Possible entries are **volta**, **tremolo** or **percent**. Multiple entries are possible.

**\voices** [music] - *ids* (list of indexes or symbols) *music* (music)

Take the given key list of numbers (indicating the use of ‘**\voiceOne**’...) or symbols (indicating voice names, typically converted from strings by argument list processing) and assign the following **\**-separated music to contexts according to that list. Named rather than numbered contexts can be used for continuing one voice (for the sake of spanners and lyrics), usually requiring a **\voiceOne**-style override at the beginning of the passage and a **\oneVoice** override at its end.

The default

```
<< ... \ \ ... \ \ ... >>
```

construct would correspond to

```
\voices 1,2,3 << ... \ \ ... \ \ ... >>
```

**\void** [void] - *arg* (any type)

Accept a scheme argument, return a void expression. Use this if you want to have a scheme expression evaluated because of its side-effects, but its value ignored.

**\volta** [music] - *volta-numbers* (number list) *music* (music)

Mark *music* as being limited to the volte given in *volta-numbers* when the innermost enclosing repeat is unfolded. Volta number begins at 1 and increases by 1 with each repetition.

**\vshape** [music] - *offsets* (list) *item* (key list or music)

Like **\shape**, but additionally show control points for ease of tweaking.

**\withMusicProperty** [music] - *sym* (symbol) *val* (any type) *music* (music)

Set *sym* to *val* in *music*.

**\xNote** [music] - *note* (music)

Print *note* with a cross-shaped note head.

**\=** [post event] - *id* (index or symbol) *event* (post event)

This sets the **spanner-id** property of the following *event* to the given *id* (non-negative integer or symbol). This can be used to tell LilyPond how to connect overlapping or parallel slurs or phrasing slurs within a single Voice.

```
\fixed c' { c\=1( d\=2( e\=1) f\=2) }
```



## A.20 Identificadores de modificación de contextos

Se definen las siguientes instrucciones para su uso como modificaciones de contexto dentro de un bloque `\layout` o `\with`.

### `\RemoveAllEmptyStaves`

Remove staves which are considered to be empty according to the list of interfaces set by `keepAliveInterfaces`, including those in the first system.

- Sets grob property `remove-empty` in Sección ‘‘VerticalAxisGroup’’ en *Referencia de Funcionamiento Interno* to `#t`.
- Sets grob property `remove-first` in Sección ‘‘VerticalAxisGroup’’ en *Referencia de Funcionamiento Interno* to `#t`.

### `\RemoveEmptyStaves`

Remove staves which are considered to be empty according to the list of interfaces set by `keepAliveInterfaces`.

- Sets grob property `remove-empty` in Sección ‘‘VerticalAxisGroup’’ en *Referencia de Funcionamiento Interno* to `#t`.

## A.21 Predicados de tipo predefinidos

Predicates return `#t` when their argument is of the named type and `#f` if it isn’t.

### R5RS primary predicates

Primary predicates can be applied to any expression. They can be used on their own as predicates for LilyPond functions. The predicates here are part of the Scheme standard R5RS.

Type predicate	Description
<code>boolean?</code>	boolean
<code>char?</code>	character
<code>complex?</code>	complex number
<code>eof-object?</code>	end-of-file object
<code>input-port?</code>	input port
<code>integer?</code>	integer
<code>list?</code>	list ( <i>use <code>cheap-list?</code> for faster processing</i> )
<code>null?</code>	null
<code>number?</code>	number
<code>output-port?</code>	output port
<code>pair?</code>	pair
<code>port?</code>	port
<code>procedure?</code>	procedure
<code>rational?</code>	rational number
<code>real?</code>	real number
<code>string?</code>	string
<code>symbol?</code>	symbol
<code>vector?</code>	vector

### R5RS secondary predicates

Secondary predicates are only applicable to specific expressions (for example, to numbers). They will throw a type error when applied to expressions they are not intended for. The predicates here are part of the Scheme standard R5RS.

Type predicate	Description
char-alphabetic?	alphabetic character
char-lower-case?	lower-case character
char-numeric?	numeric character
char-upper-case?	upper-case character
char-whitespace?	whitespace character
even?	even number
exact?	exact number
inexact?	inexact number
negative?	negative number
odd?	odd number
positive?	positive number
zero?	zero

## Guile predicates

These predicates are defined by Guile but are not part of a Scheme standard.

Type predicate	Description
hash-table?	hash table

## LilyPond scheme predicates

These predicates are only available within LilyPond and defined in Scheme.

Type predicate	Description
boolean-or-symbol?	boolean or symbol
cheap-list?	list ( <i>use this instead of list? for faster processing</i> )
color?	color
exact-rational?	an exact rational number
fraction?	fraction, as pair
grob-list?	list of grobs
index?	non-negative integer
integer-or-markup?	integer or markup
key?	index or symbol
key-list?	list of indexes or symbols
key-list-or-music?	key list or music
key-list-or-symbol?	key list or symbol
markup?	markup
markup-command-list?	markup command list
markup-list?	markup list
moment-pair?	pair of moment objects
number-list?	number list
number-or-grob?	number or grob
number-or-pair?	number or pair
number-or-string?	number or string
number-pair?	pair of numbers
number-pair-list?	list of number pairs
rational-or-procedure?	an exact rational or procedure
rhythmic-location?	rhythmic location
scale?	non-negative rational, fraction, or moment
scheme?	any type
string-or-music?	string or music

<code>string-or-pair?</code>	string or pair
<code>string-or-symbol?</code>	string or symbol
<code>symbol-list?</code>	symbol list
<code>symbol-list-or-music?</code>	symbol list or music
<code>symbol-list-or-symbol?</code>	symbol list or symbol
<code>void?</code>	void

## LilyPond exported predicates

These predicates are only available within LilyPond and usually defined in C++.

Type predicate	Description
<code>ly:book?</code>	book
<code>ly:context?</code>	context
<code>ly:context-def?</code>	context definition
<code>ly:context-mod?</code>	context modification
<code>ly:dimension?</code>	dimension, in staff space
<code>ly:dir?</code>	direction
<code>ly:dispatcher?</code>	dispatcher
<code>ly:duration?</code>	duration
<code>ly:event?</code>	post event
<code>ly:font-metric?</code>	font metric
<code>ly:grob?</code>	graphical (layout) object
<code>ly:grob-array?</code>	array of grobs
<code>ly:grob-properties?</code>	grob properties
<code>ly:input-location?</code>	input location
<code>ly:item?</code>	item
<code>ly:iterator?</code>	iterator
<code>ly:lily-lexer?</code>	lily-lexer
<code>ly:lily-parser?</code>	lily-parser
<code>ly:listener?</code>	listener
<code>ly:moment?</code>	moment
<code>ly:music?</code>	music
<code>ly:music-function?</code>	music function
<code>ly:music-list?</code>	list of music objects
<code>ly:music-output?</code>	music output
<code>ly:otf-font?</code>	OpenType font
<code>ly:output-def?</code>	output definition
<code>ly:page-marker?</code>	page marker
<code>ly:pango-font?</code>	pango font
<code>ly:paper-book?</code>	paper book
<code>ly:paper-system?</code>	paper-system Prob
<code>ly:pitch?</code>	pitch
<code>ly:prob?</code>	property object
<code>ly:score?</code>	score
<code>ly:skyline?</code>	skyline
<code>ly:skyline-pair?</code>	pair of skylines
<code>ly:source-file?</code>	source file
<code>ly:spanner?</code>	spanner
<code>ly:spring?</code>	spring
<code>ly:stencil?</code>	stencil
<code>ly:stream-event?</code>	stream event

<code>ly:transform?</code>	coordinate transform
<code>ly:translator?</code>	translator
<code>ly:translator-group?</code>	translator group
<code>ly:unpure-pure-container?</code>	unpure/pure container

## A.22 Funciones de Scheme

`add-bar-glyph-print-procedure` *glyph proc* [Función]

Specify the single glyph *glyph* that calls print procedure *proc*. The procedure *proc* has to be defined in the form `(make-...-bar-line grob extent)` even if the *extent* is not used within the routine.

`ly:add-context-mod` *contextmods modification* [Función]

Adds the given context *modification* to the list *contextmods* of context modifications.

`add-grace-property` *context-name grob sym val* [Función]

Set *sym=val* for *grob* in *context-name*.

`ly:add-interface` *iface desc props* [Función]

Add a new grob interface. *iface* is the interface name, *desc* is the interface description, and *props* is the list of user-settable properties for the interface.

`ly:add-listener` *callback disp cl* [Función]

Add the single-argument procedure *callback* as listener to the dispatcher *disp*. Whenever *disp* hears an event of class *cl*, it calls *callback* with it.

`add-music-fonts` *node family name brace design-size-alist factor* [Función]

Set up music fonts.

Arguments:

- *node* is the font tree to modify.
- *family* is the family name of the music font.
- *name* is the basename for the music font. *name*-<designsize>.otf should be the music font,
- *brace* is the basename for the brace font. *brace*-*brace*.otf should have piano braces.
- *design-size-alist* is a list of `(rounded . designsize)`. *rounded* is a suffix for font filenames, while *designsize* should be the actual design size. The latter is used for text fonts loaded through pango/fontconfig.
- *factor* is a size factor relative to the default size that is being used. This is used to select the proper design size for the text fonts.

`add-new-clef` *clef-name clef-glyph clef-position transposition c0-position* [Función]

Append the entries for a clef symbol to supported clefs and *c0-pitch-alist*.

`ly:add-option` *sym val description* [Función]

Add a program option *sym*. *val* is the default value and *description* is a string description.

`add-simple-time-signature-style` *style proc* [Función]

Specify the procedure *proc* returning markup for a time signature style *style*. The procedure is called with one argument, the pair `(numerator . denominator)`.

`add-stroke-glyph` *stencil grob dir stroke-style flag-style* [Función]

Load and add a stroke (represented by a glyph in the font) to the given flag stencil.



- add-stroke-straight** *stencil grob dir log stroke-style offset length thickness stroke-thickness* [Función]  
 Add the stroke for acciaccatura to the given flag stencil. The stroke starts for up-flags at ‘upper-end-of-flag + (0,length/2)’ and ends at ‘(0, vertical-center-of-flag-end) - (flag-x-width/2, flag-x-width + flag-thickness)’. Here ‘length’ is the whole length, while ‘flag-x-width’ is just the x-extent and thus depends on the angle! Other combinations don’t look as good. For down-stems the y-coordinates are simply mirrored.
- alist->hash-table** *lst* [Función]  
 Convert alist to table
- ly:all-grob-interfaces** [Función]  
 Return the hash table with all grob interface descriptions.
- ly:all-options** [Función]  
 Get all option settings in an alist.
- ly:all-output-backend-commands** [Función]  
 Return the list of extra output backend commands that are used internally in lily/*stencil-interpret.cc*.
- ly:all-stencil-commands** [Función]  
 Return the list of stencil commands that can be defined in the output modules (*output-\*.scm*).
- ly:all-stencil-expressions** [Función]  
 Return all symbols recognized as stencil expressions.
- allow-volta-hook** *bar-glyph* [Función]  
 Allow the volta bracket hook being drawn over bar line *bar-glyph*.
- alterations-in-key** *pitch-list* [Función]  
 Count number of sharps minus number of flats.
- ly:angle** *x y* [Función]  
 Calculates angle in degrees of given vector. With one argument, *x* is a number pair indicating the vector. With two arguments, *x* and *y* specify the respective coordinates.
- angle-0-2pi** *angle* [Función]  
 Take *angle* (in radians) and maps it between 0 and 2pi.
- angle-0-360** *angle* [Función]  
 Take *angle* (in degrees) and maps it between 0 and 360 degrees.
- arrow-stencil** *x y thick staff-space grob* [Función]  
 Returns a right-pointing, filled arrow-head, where *x* determines the basic horizontal position and *y* determines the basic vertical position. Both values are adjusted using *staff-space*, which is *StaffSymbol*’s staff space. *thick* is the used line thickness.
- arrow-stencil-maker** *start? end?* [Función]  
 Return a function drawing a line from current point to *destination*, with optional arrows of *max-size* on start and end controlled by *start?* and *end?*.
- ly:assoc-get** *key alist default-value strict-checking* [Función]  
 Return value if *key* in *alist*, else *default-value* (or *#f* if not specified). If *strict-checking* is set to *#t* and *key* is not in *alist*, a programming\_error is output.

<b>ly:axis-group-interface::add-element</b> <i>grob grob-element</i>	[Función]
Set <i>grob</i> the parent of <i>grob-element</i> on all axes of <i>grob</i> .	
<b>ly:bar-line::calc-anchor</b> <i>grob</i>	[Función]
Calculate the anchor position of a bar line. The anchor is used for the correct placement of bar numbers etc.	
<b>bar-line::calc-break-visibility</b> <i>grob</i>	[Función]
Calculate the visibility of a bar line at line breaks.	
<b>bar-line::calc-glyph-name</b> <i>grob</i>	[Función]
Determine the <i>glyph-name</i> of the bar line depending on the line break status.	
<b>bar-line::calc-glyph-name-for-direction</b> <i>glyph dir</i>	[Función]
Determine the <i>glyph-name</i> of the bar line depending on the line break status.	
<b>bar-line::compound-bar-line</b> <i>grob bar-glyph extent</i>	[Función]
Build the bar line stencil.	
<b>bar-line::draw-filled-box</b> <i>x-ext y-ext thickness extent grob</i>	[Función]
Return a straight bar-line created by <b>ly:round-filled-box</b> looking at <i>x-ext</i> , <i>y-ext</i> , <i>thickness</i> . The blot is calculated by <b>bar-line::calc-blot</b> , which needs <i>extent</i> and <i>grob</i> . <i>y-ext</i> is not necessarily of same value as <i>extent</i> .	
<b>ly:bar-line::print</b> <i>grob</i>	[Función]
The print routine for bar lines.	
<b>bar-line::widen-bar-extent-on-span</b> <i>grob extent</i>	[Función]
Widens the bar line <i>extent</i> towards span bars adjacent to <i>grob grob</i> .	
<b>base-length</b> <i>time-signature time-signature-settings</i>	[Función]
Get <i>baseMoment</i> rational value for <i>time-signature</i> from <i>time-signature-settings</i> .	
<b>ly:basic-progress</b> <i>str rest</i>	[Función]
A Scheme callable function to issue a basic progress message <i>str</i> . The message is formatted with <i>format</i> and <i>rest</i> .	
<b>beam-exceptions</b> <i>time-signature time-signature-settings</i>	[Función]
Get <i>beamExceptions</i> value for <i>time-signature</i> from <i>time-signature-settings</i> .	
<b>beat-structure</b> <i>base-length time-signature time-signature-settings</i>	[Función]
Get <i>beatStructure</i> value in <i>base-length</i> units for <i>time-signature</i> from <i>time-signature-settings</i> .	
<b>bend::arrow-head-stencil</b> <i>thickness x-y-coords height width dir</i>	[Función]
Returns an arrow head stencil. Calculated from the given dimensions <i>height</i> and <i>width</i> , translated to <i>x-y-coords</i> , the end of the bend-spanners (curved) line.	
<b>bend::calc-bend-x-begin</b> <i>bend-spanner bounding-noteheads factor quarter-tone-diffs</i>	[Función]
Calculates the starting values in X-direction of the bend. After a line break, the values from the right bound are taken minus 1.5 staff-spaces. For bends-down or if <i>grob-property</i> 'style equals to 'pre-bend, 'hold or 'pre-bend-hold, <i>interval-center</i> is applied the topmost note head of the starting note heads. In any other case the right edge of the starting note head is used. The value of <i>BendSpanner.details.horizontal-left-padding</i> is added, which may be changed by an appropriate override. Returns a list of the same length as the amount of bend-starting note heads.	

- bend::calc-bend-x-end** *bend-spanner top-left-tab-nhd top-right-tab-nhd* [Función]  
 Calculates the ending X-coordinate of *bend-spanner*. At the line end take the items of **BreakAlignGroup** into account and a little padding. Ends an unbroken spanner or the last of a broken one in the middle of the topmost note head of its bounding note column.
- bend::target-cautionary** *spanner* [Función]  
 Sets 'display-cautionary of all relevant note heads of spanners right bound to true. As a result they appear parenthesized. This procedure is the default value of 'before-line-breaking.
- bend::text-string** *spanner* [Función]  
 Takes a spanner-grob, calculates a list with the quarter tone diffs between the pitches of starting and ending bound. Because bending to different amounts is very unlikely, only the first element of this list is returned as a string.
- bend-spanner::print** *grob* [Función]  
 Returns the final stencil. A line and curve, arrow head and a text representing the amount a string is bent.
- ly:book?** *x* [Función]  
 Is *x* a Book object?
- ly:book-add-bookpart!** *book-smob book-part* [Función]  
 Add *book-part* to *book-smob* book part list.
- ly:book-add-score!** *book-smob score* [Función]  
 Add *score* to *book-smob* score list.
- ly:book-book-parts** *book* [Función]  
 Return book parts in *book*.
- book-first-page** *layout props* [Función]  
 Return the 'first-page-number of the entire book
- ly:book-header** *book* [Función]  
 Return header in *book*.
- ly:book-paper** *book* [Función]  
 Return paper in *book*.
- ly:book-process** *book-smob default-paper default-layout output* [Función]  
 Print book. *output* is passed to the backend unchanged. For example, it may be a string (for file based outputs) or a socket (for network based output).
- ly:book-process-to-systems** *book-smob default-paper default-layout output* [Función]  
 Print book. *output* is passed to the backend unchanged. For example, it may be a string (for file based outputs) or a socket (for network based output).
- ly:book-scores** *book* [Función]  
 Return scores in *book*.
- ly:book-set-header!** *book module* [Función]  
 Set the book header.
- box-grob-stencil** *grob* [Función]  
 Make a box of exactly the extents of the grob. The box precisely encloses the contents.

<b>box-stencil</b> <i>stencil thickness padding</i>	[Función]
Add a box around <i>stencil</i> , producing a new stencil.	
<b>ly:bp</b> <i>num</i>	[Función]
<i>num</i> bigpoints (1/72th inch).	
<b>ly:bracket</b> <i>a iv t p</i>	[Función]
Make a bracket in direction <i>a</i> . The extent of the bracket is given by <i>iv</i> . The wings protrude by an amount of <i>p</i> , which may be negative. The thickness is given by <i>t</i> .	
<b>bracketify-stencil</b> <i>stil axis thick protrusion padding</i>	[Función]
Add brackets around <i>stil</i> , producing a new stencil.	
<b>break-alignable-interface::self-alignment-of-anchor</b> <i>g</i>	[Función]
Return a value for <i>g</i> 's <b>self-alignment-X</b> that will place <i>g</i> on the same side of the reference point defined by a <b>break-aligned</b> item such as a <b>Clef</b> .	
<b>break-alignable-interface::self-alignment-opposite-of-anchor</b> <i>g</i>	[Función]
Return a value for <i>g</i> 's <b>self-alignment-X</b> that will place <i>g</i> on the opposite side of the reference point defined by a <b>break-aligned</b> item such as a <b>Clef</b> .	
<b>break-alignment-list</b> <i>end-of-line middle begin-of-line</i>	[Función]
Return a callback that calculates a value based on a grob's break direction.	
<b>ly:broadcast</b> <i>disp ev</i>	[Función]
Send the stream event <i>ev</i> to the dispatcher <i>disp</i> .	
<b>calc-harmonic-pitch</b> <i>pitch music</i>	[Función]
Calculate the harmonic pitches in <i>music</i> given <i>pitch</i> as the non-harmonic pitch.	
<b>ly:camel-case-&gt;lisp-identifier</b> <i>name-sym</i>	[Función]
Convert FooBar_Bla to foo-bar-bla style symbol.	
<b>centered-stencil</b> <i>stencil</i>	[Función]
Center stencil <i>stencil</i> in both the X and Y directions.	
<b>centered-text-interface::print</b> <i>grob</i>	[Función]
Print some text between two non-musical columns according to the <b>spacing-pair</b> property.	
<b>ly:chain-assoc-get</b> <i>key achain default-value strict-checking</i>	[Función]
Return value for <i>key</i> from a list of alists <i>achain</i> . If no entry is found, return <i>default-value</i> or <b>#f</b> if <i>default-value</i> is not specified. With <i>strict-checking</i> set to <b>#t</b> , a <code>programming_error</code> is output in such cases.	
<b>change-pitches</b> <i>music converter</i>	[Función]
Recurse through <i>music</i> , applying <i>converter</i> to pitches. Converter is typically a transposer or an inverter as defined above in this module, but may be user-defined. The converter function must take a single pitch as its argument and return a new pitch. These are LilyPond scheme pitches, e.g. (ly:make-pitch 0 2 0)	
<b>check-context-path</b> <i>path . lambda*:G59</i>	[Función]
Check a context property path specification <i>path</i> , a symbol list (or a single symbol), for validity and possibly complete it. Returns the completed specification, or <b>#f</b> when rising an error (using optionally <i>location</i> ).	
<b>ly:check-expected-warnings</b>	[Función]
Check whether all expected warnings have really been triggered.	

**check-grob-path** *path . rest* [Función]

Check a grob path specification *path*, a symbol list (or a single symbol), for validity and possibly complete it. Returns the completed specification, or **#f** if invalid. If optional *parser* is given, a syntax error is raised in that case, optionally using *location*. If an optional keyword argument **#:start** *start* is given, the parsing starts at the given index in the sequence 'Context.Grob.property.sub-property...', with the default of '0' implying the full path. If there is no valid first element of *path* fitting at the given path location, an optionally given **#:default** *default* is used as the respective element instead without checking it for validity at this position.

The resulting path after possibly prepending *default* can be constrained in length by optional arguments **#:min** *min* and **#:max** *max*, defaulting to '1' and unlimited, respectively.

**check-music-path** *path . rest* [Función]

Check a music property path specification *path*, a symbol list (or a single symbol), for validity and possibly complete it. Returns the completed specification, or **#f** when rising an error (using optionally *location*).

**circle-stencil** *stencil thickness padding* [Función]

Add a circle around *stencil*, producing a new stencil.

**clef-transposition-markup** *oct style* [Función]

The transposition sign formatting function. *oct* is supposed to be a string holding the transposition number, *style* determines the way the transposition number is displayed.

**ly:cm** *num* [Función]  
*num* cm.

**collect-book-music-for-book** *book music* [Función]  
Book music handler.

**collect-bookpart-for-book** *book-part* [Función]  
Toplevel book-part handler.

**collect-music-aux** *score-handler music* [Función]  
Pass *music* to *score-handler*, with preprocessing for page layout instructions.

**collect-music-for-book** *music* [Función]  
Top-level music handler.

**ly:command-line-code** [Función]  
The Scheme code specified on command-line with **-e**.

**ly:command-line-options** [Función]  
The Scheme options specified on command-line with **-d**.

**ly:connect-dispatchers** *to from* [Función]  
Make the dispatcher *to* listen to events from *from*.

**constante-hairpin** *grob* [Función]

Create hairpin based on a list of *coords* in (cons *x y*) form. *x* is the portion of the width consumed for a given line and *y* is the portion of the height. For example, '((0 . 0) (0.3 . 0.7) (0.8 . 0.9) (1.0 . 1.0))' means that at the point where the hairpin has consumed 30% of its width, it must be at 70% of its height. Once it is to 80% width, it must be at 90% height. It finishes at 100% width and 100% height. If *coords* does not begin with '(0 . 0)' the final hairpin may have an open tip. For example '(0 . 0.5)' will cause an open end of 50%

of the usual height. *mirrored?* indicates if the hairpin is mirrored over the Y-axis or if just the upper part is drawn. Returns a function that accepts a hairpin grob as an argument and draws the stencil based on its coordinates.

```
#(define simple-hairpin
  (elbowed-hairpin '((0 . 0)(1.0 . 1.0)) #t))

\relative c' {
  \override Hairpin #'stencil = #simple-hairpin
  a\p\< a a a\f
}
```

**construct-chord-elements** *root duration modifications* [Función]

Build a chord on *root* using modifiers in *modifications*. **NoteEvents** have duration *duration*.

Notes: Natural 11 is left from chord if not explicitly specified.

Entry point for the parser.

**ly:context?** *x* [Función]

Is *x* a **Context** object?

**ly:context-current-moment** *context* [Función]

Return the current moment of *context*.

**ly:context-def?** *x* [Función]

Is *x* a **Context\_def** object?

**ly:context-def-lookup** *def sym val* [Función]

Return the value of *sym* in context definition *def* (e.g., **\Voice**). If no value is found, return *val* or **()** if *val* is undefined. *sym* can be any of ‘**default-child**’, ‘**consists**’, ‘**description**’, ‘**aliases**’, ‘**accepts**’, ‘**property-ops**’, ‘**context-name**’, ‘**group-type**’.

**ly:context-def-modify** *def mod* [Función]

Return the result of applying the context-mod *mod* to the context definition *def*. Does not change *def*.

**ly:context-event-source** *context* [Función]

Return **event-source** of context *context*.

**ly:context-events-below** *context* [Función]

Return a **stream-distributor** that distributes all events from *context* and all its subcontexts.

**ly:context-find** *context name* [Función]

Find a parent of *context* that has name or alias *name*. Return **#f** if not found.

**ly:context-grob-definition** *context name* [Función]

Return the definition of *name* (a symbol) within *context* as an alist.

**ly:context-id** *context* [Función]

Return the ID string of *context*, i.e., for **\context Voice = "one"** ... return the string **one**.

**ly:context-matched-pop-property** *context grob cell* [Función]

This undoes a particular **\override**, **\once \override** or **\once \revert** when given the specific alist pair to undo.

**ly:context-mod?** *x* [Función]

Is *x* a **Context\_mod** object?

<b>ly:context-mod-apply!</b> <i>context mod</i>	[Función]
Apply the context modification <i>mod</i> to <i>context</i> .	
<b>ly:context-name</b> <i>context</i>	[Función]
Return the name of <i>context</i> , i.e., for <code>\context Voice = "one"</code> ... return the symbol <code>Voice</code> .	
<b>ly:context-now</b> <i>context</i>	[Función]
Return <code>now-moment</code> of context <i>context</i> .	
<b>ly:context-parent</b> <i>context</i>	[Función]
Return the parent of <i>context</i> , <code>#f</code> if none.	
<b>ly:context-property</b> <i>context sym def</i>	[Función]
Return the value for property <i>sym</i> in <i>context</i> . If <i>def</i> is given, and property value is <code>'()</code> , return <i>def</i> .	
<b>ly:context-property-where-defined</b> <i>context name</i>	[Función]
Return the context above <i>context</i> where <i>name</i> is defined.	
<b>ly:context-pushpop-property</b> <i>context grob eltprop val</i>	[Función]
Do <code>\temporary \override</code> or <code>\revert</code> operation in <i>context</i> . The grob definition <i>grob</i> is extended with <i>eltprop</i> (if <i>val</i> is specified) or reverted (if unspecified).	
<b>ly:context-set-property!</b> <i>context name val</i>	[Función]
Set value of property <i>name</i> in context <i>context</i> to <i>val</i> .	
<b>context-spec-music</b> <i>m context . lambda*:G70</i>	[Función]
Add <code>\context context = id \with mods</code> to <i>m</i> .	
<b>ly:context-unset-property</b> <i>context name</i>	[Función]
Unset value of property <i>name</i> in context <i>context</i> .	
<b>copy-repeat-chord</b> <i>original-chord repeat-chord duration event-types</i>	[Función]
Copies all events in <i>event-types</i> (be sure to include <code>rhythmic-events</code> ) from <i>original-chord</i> over to <i>repeat-chord</i> with their articulations filtered as well. Any duration is replaced with the specified <i>duration</i> .	
<b>count-list</b> <i>lst</i>	[Función]
Given <i>lst</i> as <code>(E1 E2 ...)</code> , return <code>((E1 . 1) (E2 . 2) ...)</code> .	
<b>create-glyph-flag</b> <i>flag-style dir-modifier grob</i>	[Función]
Create a flag stencil by looking up the glyph from the font.	
<b>cross-staff-connect</b> <i>stem</i>	[Función]
Set cross-staff property of the stem to this function to connect it to other stems automatically	
<b>css-&gt;colorlist</b> <i>code</i>	[Función]
Turn a CSS-like color string into an <code>rgb-color</code> list. The given string may be either a predefined color name or a hexadecimal color code, in which case it must be prefixed with <code>#</code> and entered between double quotes. Alpha channel information is supported (as eight-character color codes, or four chars in shorthand mode), and will be passed to the output backend – that may or may not use it.	
<b>cue-substitute</b> <i>quote-music</i>	[Función]
Must happen after <code>quote-substitute</code> .	
<b>cyclic-base-value</b> <i>value cycle</i>	[Función]
Take <i>value</i> and modulo-maps it between 0 and base <i>cycle</i> .	

**ly:debug** *str rest* [Función]

A Scheme callable function to issue a debug message *str*. The message is formatted with *format* and *rest*.

**default-flag** *grob* [Función]

Create a flag stencil for the stem. Its style will be derived from the '*style* Flag property. By default, *lilypond* uses a C++ Function (which is slightly faster) to do exactly the same as this function. However, if one wants to modify the default flags, this function can be used to obtain the default flag stencil, which can then be modified at will. The correct way to do this is:

```
\override Flag #'stencil = #default-flag
\override Flag #'style = #'mensural
```

**ly:default-scale** [Función]

Get the global default scale.

**define-bar-line** *bar-glyph eol-glyph bol-glyph span-glyph* [Función]

Define a bar glyph *bar-glyph* and its substitute at the end of a line (*eol-glyph*), at the beginning of a new line (*bol-glyph*) and as a span bar (*span-glyph*) respectively.

**define-event-class** *class parent* [Función]

Defines a new event *class* derived from *parent*, a previously defined event class.

**define-fonts** *paper define-font define-pango-pf* [Función]

Return a string of all fonts used in *paper*, invoking the functions *define-font* and *define-pango-pf* for producing the actual font definition.

**define-tag-group** *tags* [Función]

Define a tag-group consisting of the given *tags*, a list of symbols. Returns *#f* if successful, and an error message if there is a conflicting tag group definition.

**degrees->radians** *angle-degrees* [Función]

Convert the given angle from degrees to radians.

**descend-to-context** *m context . lambda\*:G72* [Función]

Like *context-spec-music*, but only descending.

**determine-split-list** *evl1 evl2 chord-range* [Función]

*evl1* and *evl2* should be ascending. *chord-range* is a pair of numbers (min . max) defining the distance in steps between notes that may be combined into a chord or unison.

**determine-string-fret-finger** *context notes specified-info rest* [Función]

Determine string numbers and frets for playing *notes* as a chord, given specified information *specified-info*. *specified-info* is a list with two list elements, specified strings *defined-strings* and specified fingerings *defined-fingers*. Only a fingering of 0 will affect the fret selection, as it specifies an open string. If *defined-strings* is '(), the context property *defaultStrings* will be used as a list of defined strings. Will look for predefined fretboards if *predefinedFretboardTable* is not *#f*. If *rest* is present, it contains the *FretBoard* grob, and a fretboard will be created. Otherwise, a list of (*string fret finger*) lists will be returned. If the context-property *supportNonIntegerFret* is set *#t*, micro-tones are supported for *TabStaff*, but not for *FretBoards*.

**ly:dimension?** *d* [Función]

Is *d* a dimension? Used to distinguish length variables from normal numbers.



<b>ly:dir?</b> <i>s</i>	[Función]
Is <i>s</i> a direction? Valid directions are -1, 0, or 1, where -1 represents left or down, 1 represents right or up, and 0 represents a neutral direction.	
<b>dir-basename</b> <i>file . rest</i>	[Función]
Strip suffixes in <i>rest</i> , but leave directory component for <i>file</i> .	
<b>ly:directed</b> <i>direction magnitude</i>	[Función]
Calculates an ( <i>x</i> . <i>y</i> ) pair with optional <i>magnitude</i> (defaulting to 1.0) and <i>direction</i> specified either as an angle in degrees or a coordinate pair giving the direction. If <i>magnitude</i> is a pair, the respective coordinates are scaled independently, useful for ellipse drawings.	
<b>ly:disconnect-dispatchers</b> <i>to from</i>	[Función]
Stop the dispatcher <i>to</i> listening to events from <i>from</i> .	
<b>ly:dispatcher?</b> <i>x</i>	[Función]
Is <i>x</i> a Dispatcher object?	
<b>display-lily-music</b> <i>expr . lambda*:G54</i>	[Función]
Display the music expression using LilyPond syntax	
<b>display-music</b> <i>music . lambda*:G40</i>	[Función]
Display music, not done with <b>music-map</b> for clarity of presentation.	
<b>display-scheme-music</b> <i>obj . lambda*:G42</i>	[Función]
Displays ‘obj’, typically a music expression, in a friendly fashion, which often can be read back in order to generate an equivalent expression.	
<b>dodecaphonic-no-repeat-rule</b> <i>context pitch barnum measurepos</i>	[Función]
An accidental rule that typesets an accidental before every note (just as in the dodecaphonic accidental style) <i>except</i> if the note is immediately preceded by a note with the same pitch. This is a common accidental style in contemporary notation.	
<b>ly:duration?</b> <i>x</i>	[Función]
Is <i>x</i> a Duration object?	
<b>ly:duration&lt;?</b> <i>p1 p2</i>	[Función]
Is <i>p1</i> shorter than <i>p2</i> ?	
<b>ly:duration-&gt;string</b> <i>dur</i>	[Función]
Convert <i>dur</i> to a string.	
<b>ly:duration-dot-count</b> <i>dur</i>	[Función]
Extract the dot count from <i>dur</i> .	
<b>duration-dot-factor</b> <i>dotcount</i>	[Función]
Given a count of the dots used to extend a musical duration, return the numeric factor by which they increase the duration.	
<b>ly:duration-factor</b> <i>dur</i>	[Función]
Extract the compression factor from <i>dur</i> . Return it as a pair.	
<b>ly:duration-length</b> <i>dur</i>	[Función]
The length of the duration as a <b>moment</b> .	
<b>duration-length</b> <i>dur</i>	[Función]
Return the overall length of a duration, as a number of whole notes. (Not to be confused with <b>ly:duration-length</b> , which returns a less-useful moment object.)	

- `duration-line::calc grob` [Función]  
Return list of values needed to print a stencil for `DurationLine`.
- `duration-line::print grob` [Función]  
Return the stencil of `DurationLine`.
- `ly:duration-log dur` [Función]  
Extract the duration log from *dur*.
- `duration-log-factor lognum` [Función]  
Given a logarithmic duration number, return the length of the duration, as a number of whole notes.
- `ly:duration-scale dur` [Función]  
Extract the compression factor from *dur*. Return it as a rational.
- `duration-visual dur` [Función]  
Given a duration object, return the visual part of the duration (base note length and dot count), in the form of a duration object with non-visual scale factor 1.
- `duration-visual-length dur` [Función]  
Given a duration object, return the length of the visual part of the duration (base note length and dot count), as a number of whole notes.
- `dynamic-text-spanner::before-line-breaking grob` [Función]  
Monitor left bound of `DynamicTextSpanner` for absolute dynamics. If found, ensure `DynamicText` does not collide with spanner text by changing 'attach-dir and 'padding. Reads the 'right-padding property of `DynamicText` to fine tune space between the two text elements.
- `ly:effective-prefix` [Función]  
Return effective prefix.
- `ellipse-stencil stencil thickness x-padding y-padding` [Función]  
Add an ellipse around *stencil*, padded by the padding pair, producing a new stencil.
- `ly:encode-string-for-pdf str` [Función]  
Encode the given string to either Latin1 (which is a subset of the `PDFDocEncoding`) or if that's not possible to full UTF-16BE with Byte-Order-Mark (BOM).
- `ly:engraver-announce-end-grob engraver grob cause` [Función]  
Announce the end of a grob (i.e., the end of a spanner) originating from given *engraver* instance, with *grob* being a grob. *cause* should either be another grob or a music event.
- `ly:engraver-make-grob engraver grob-name cause` [Función]  
Create a grob originating from given *engraver* instance, with given *grob-name*, a symbol. *cause* should either be another grob or a music event.
- `ly:error str rest` [Función]  
A Scheme callable function to issue the error *str*. The error is formatted with `format` and *rest*.
- `eval-carefully symbol module . default` [Función]  
Check whether all symbols in expr *symbol* are reachable in module *module*. In that case evaluate, otherwise print a warning and set an optional *default*.

<b>ly:event?</b> <i>obj</i>	[Función]
Is <i>obj</i> a proper (non-rhythmic) event object?	
<b>event-chord-notes</b> <i>event-chord</i>	[Función]
Return a list of all notes from <i>event-chord</i> .	
<b>event-chord-pitches</b> <i>event-chord</i>	[Función]
Return a list of all pitches from <i>event-chord</i> .	
<b>event-chord-reduce</b> <i>music</i>	[Función]
Reduces event chords in <i>music</i> to their first note event, retaining only the chord articulations. Returns the modified <i>music</i> .	
<b>event-chord-wrap!</b> <i>music</i>	[Función]
Wrap isolated rhythmic events and non-postevent events in <i>music</i> inside of an <b>EventChord</b> . Chord repeats ‘q’ are expanded using the default settings of the parser.	
<b>ly:event-deep-copy</b> <i>m</i>	[Función]
Copy <i>m</i> and all sub expressions of <i>m</i> .	
<b>event-has-articulation?</b> <i>event-type stream-event</i>	[Función]
Is <i>event-type</i> in the <b>articulations</b> list of <i>stream-event</i> ?	
<b>ly:event-property</b> <i>sev sym val</i>	[Función]
Get the property <i>sym</i> of stream event <i>sev</i> . If <i>sym</i> is undefined, return <i>val</i> or ‘()’ if <i>val</i> is not specified.	
<b>ly:event-set-property!</b> <i>ev sym val</i>	[Función]
Set property <i>sym</i> in event <i>ev</i> to <i>val</i> .	
<b>expand-repeat-chords!</b> <i>event-types music</i>	[Función]
Walks through <i>music</i> and fills repeated chords (notable by having a duration in <b>duration</b> ) with the notes from their respective predecessor chord.	
<b>expand-repeat-notes!</b> <i>music</i>	[Función]
Walks through <i>music</i> and gives pitchless notes (not having a pitch in <b>pitch</b> or a drum type in <b>drum-type</b> ) the pitch(es) from the predecessor note/chord if available.	
<b>ly:expect-warning</b> <i>str rest</i>	[Función]
A Scheme callable function to register a warning to be expected and subsequently suppressed. If the warning is not encountered, a warning about the missing warning will be shown. The message should be translated with ( <b>_ ...</b> ) and changing parameters given after the format string.	
<b>extract-beam-exceptions</b> <i>music</i>	[Función]
Creates a value useful for setting <b>beamExceptions</b> from <i>music</i> .	
<b>extract-music</b> <i>music pred?</i>	[Función]
Return a flat list of all music matching <i>pred?</i> inside of <i>music</i> , not recursing into matches themselves.	
<b>extract-named-music</b> <i>music music-name</i>	[Función]
Return a flat list of all music named <i>music-name</i> (either a single event symbol or a list of alternatives) inside of <i>music</i> , not recursing into matches themselves.	

**ly:extract-subfont-from-collection** *collection-file-name idx subfont-file-name* [Función]

Extract the subfont of index *idx* in TrueType collection (TTC) or OpenType/CFF collection (OTC) file *collection-file-name* and write it to file *subfont-file-name*.

**extract-typed-music** *music type* [Función]

Return a flat list of all music with *type* (either a single type symbol or a list of alternatives) inside of *music*, not recursing into matches themselves.

**ly:find-file** *name* [Función]

Return the absolute file name of *name*, or *#f* if not found.

**find-named-props** *prop-name grob-descriptions* [Función]

Used by `\magnifyMusic` and `\magnifyStaff`. When *grob-descriptions* is equal to the `all-grob-descriptions` alist (defined in `scm/define-grobs.scm`), this will find all grobs that can have a value for the *prop-name* property, and return them as a list in the following format:

```
'((grob prop-name)
  (grob prop-name)
  ...)
```

**find-pitch-entry** *keysig pitch accept-global accept-local* [Función]

Return the first entry in *keysig* that matches *pitch* by notename and octave. Alteration is not considered. *accept-global* states whether key signature entries should be included. *accept-local* states whether local accidentals should be included. If no matching entry is found, *#f* is returned.

**finger-glide::print** *grob* [Función]

The stencil printing procedure for grob `FingerGlideSpanner`. Depending on the grob property `style` several forms of appearance are printed. Possible settings for grob property `style` are `zigzag`, `trill`, `dashed-line`, `dotted-line`, `stub-left`, `stub-right`, `stub-both`, `bow`, `none` and `line`, which is the default.

**first-assoc** *keys lst* [Función]

Return first successful assoc of key from *keys* in *lst*.

**first-member** *members lst* [Función]

Return first successful member (of member) from *members* in *lst*.

**flared-hairpin** *grob* [Función]

Create hairpin based on a list of *coords* in `(cons x y)` form. *x* is the portion of the width consumed for a given line and *y* is the portion of the height. For example, `'((0 . 0) (0.3 . 0.7) (0.8 . 0.9) (1.0 . 1.0))` means that at the point where the hairpin has consumed 30% of its width, it must be at 70% of its height. Once it is to 80% width, it must be at 90% height. It finishes at 100% width and 100% height. If *coords* does not begin with `'(0 . 0)` the final hairpin may have an open tip. For example `'(0 . 0.5)` will cause an open end of 50% of the usual height. *mirrored?* indicates if the hairpin is mirrored over the Y-axis or if just the upper part is drawn. Returns a function that accepts a hairpin grob as an argument and draws the stencil based on its coordinates.

```
#(define simple-hairpin
  (elbowed-hairpin '((0 . 0)(1.0 . 1.0)) #t))

\relative c' {
  \override Hairpin #'stencil = #simple-hairpin
  a\p\< a a a\ff
}
```

<b>flat-flag</b> <i>grob</i>	[Función]
Flat flag style. The angles of the flags are both 0 degrees	
<b>flatten-list</b> <i>x</i>	[Función]
Unnest list.	
<b>flip-stencil</b> <i>axis stil</i>	[Función]
Flip stencil <i>stil</i> in the direction of <i>axis</i> . Value X (or 0) for <i>axis</i> flips it horizontally. Value Y (or 1) flips it vertically. <i>stil</i> is flipped in place; its position, the coordinates of its bounding box, remains the same.	
<b>fold-some-music</b> <i>pred? proc init music</i>	[Función]
This works recursively on music like <b>fold</b> does on a list, calling ‘( <i>pred?</i> <i>music</i> )’ on every music element. If <b>#f</b> is returned for an element, it is processed recursively with the same initial value of ‘previous’, otherwise ‘( <i>proc music previous</i> )’ replaces ‘previous’ and no recursion happens. The top <i>music</i> is processed using <i>init</i> for ‘previous’.	
<b>ly:font-config-add-directory</b> <i>dir</i>	[Función]
Add directory <i>dir</i> to FontConfig.	
<b>ly:font-config-add-font</b> <i>font</i>	[Función]
Add font <i>font</i> to FontConfig.	
<b>ly:font-config-display-fonts</b>	[Función]
Dump a list of all fonts visible to FontConfig.	
<b>ly:font-config-get-font-file</b> <i>name</i>	[Función]
Get the file for font <i>name</i> .	
<b>ly:font-design-size</b> <i>font</i>	[Función]
Given the font metric <i>font</i> , return the design size, relative to the current output-scale.	
<b>ly:font-file-name</b> <i>font</i>	[Función]
Given the font metric <i>font</i> , return the corresponding file name.	
<b>ly:font-get-glyph</b> <i>font name</i>	[Función]
Return a stencil from <i>font</i> for the glyph named <i>name</i> . If the glyph is not available, return an empty stencil.	
Note that this command can only be used to access glyphs from fonts loaded with <b>ly:system-font-load</b> ; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings <b>fetaMusic</b> and <b>fetaBraces</b> , respectively.	
<b>ly:font-glyph-name-to-charcode</b> <i>font name</i>	[Función]
Return the character code for glyph <i>name</i> in <i>font</i> .	
Note that this command can only be used to access glyphs from fonts loaded with <b>ly:system-font-load</b> ; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings <b>fetaMusic</b> and <b>fetaBraces</b> , respectively.	
<b>ly:font-glyph-name-to-index</b> <i>font name</i>	[Función]
Return the index for <i>name</i> in <i>font</i> .	
Note that this command can only be used to access glyphs from fonts loaded with <b>ly:system-font-load</b> ; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings <b>fetaMusic</b> and <b>fetaBraces</b> , respectively.	

- ly:font-index-to-charcode** *font index* [Función]  
 Return the character code for *index* in *font*.  
 Note that this command can only be used to access glyphs from fonts loaded with **ly:system-font-load**; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings **fetaMusic** and **fetaBraces**, respectively.
- ly:font-magnification** *font* [Función]  
 Given the font metric *font*, return the magnification, relative to the current output-scale.
- ly:font-metric?** *x* [Función]  
 Is *x* a **Font\_metric** object?
- ly:font-name** *font* [Función]  
 Given the font metric *font*, return the corresponding name.
- font-name-split** *font-name* [Función]  
 Return (FONT-NAME . DESIGN-SIZE) from *font-name* string or **#f**.
- ly:font-sub-fonts** *font* [Función]  
 Given the font metric *font* of an OpenType font, return the names of the subfonts within *font*.
- for-some-music** *stop? music* [Función]  
 Walk through *music*, process all elements calling *stop?* and only recurse if this returns **#f**.
- ly:format** *str rest* [Función]  
 LilyPond specific format, supporting *~a* and *~[0-9]f*. Basic support for *~s* is also provided.
- ly:format-output** *context* [Función]  
 Given a global context in its final state, process it and return the **Music\_output** object in its final state.
- fret->pitch** *fret* [Función]  
 Calculate a pitch given *fret* for the harmonic.
- fret-parse-terse-definition-string** *props definition-string* [Función]  
 Parse a fret diagram string that uses terse syntax; return a pair containing: *props*, modified to include the string-count determined by the definition-string, and a fret-indication list with the appropriate values
- function-chain** *arg function-list* [Función]  
 Applies a list of functions in *function-list* to *arg*. Each element of *function-list* is structured (cons function '(arg2 arg3 ...)). If function takes arguments besides *arg*, they are provided in *function-list*.  
 Example: Executing '(function-chain 1 `((,+ 1) (,- 2) (,+ 3) (,/)))' returns '1/3'.
- ly:generic-bound-extent** *grob common* [Función]  
 Determine the extent of *grob* relative to *common* along the X axis, finding its extent as a bound when it has **bound-alignment-interfaces** property list set and otherwise the full extent.
- ly:get-all-function-documentation** [Función]  
 Get a hash table with all LilyPond Scheme extension functions.
- ly:get-all-translators** [Función]  
 Return a list of all translator objects that may be instantiated.

- get-bound-note-heads** *spanner* [Función]  
 Takes a spanner grob and returns a pair containing all note heads of the initial starting and the final NoteColumn.
- ly:get-cff-offset** *font-file-name idx* [Función]  
 Get the offset of 'CFF' table for *font-file-name*, returning it as an integer. The optional *idx* argument is useful for OpenType/CFF collections (OTC) only; it specifies the font index within the OTC. The default value of *idx* is 0.
- get-chord-shape** *shape-code tuning base-chord-shapes* [Función]  
 Return the chord shape associated with *shape-code* and *tuning* in the hash-table *base-chord-shapes*.
- ly:get-context-mods** *contextmod* [Función]  
 Returns the list of context modifications stored in *contextmod*.
- ly:get-font-format** *font-file-name idx* [Función]  
 Get the font format for *font-file-name*, returning it as a symbol. The optional *idx* argument is useful for TrueType Collections (TTC) and OpenType/CFF collections (OTC) only; it specifies the font index within the TTC/OTC. The default value of *idx* is 0.
- ly:get-option** *var* [Función]  
 Get a global option setting.
- get-postscript-bbox** *string* [Función]  
 Extract the bbox from STRING, or return #f if not present.
- ly:get-spacing-spec** *from-scm to-scm* [Función]  
 Return the spacing spec going between the two given grobs, *from-scm* and *to-scm*.
- get-tweakable-music** *mus* [Función]  
 When tweaking music, returns a list of music expressions where the tweaks should be applied. Relevant for music wrappers and event chords.
- ly:gettext** *original* [Función]  
 A Scheme wrapper function for *gettext*.
- ly:grob?** *x* [Función]  
 Is *x* a Grob object?
- grob::all-objects** *grob* [Función]  
 Return a list of the names and contents of all properties having type **ly:grob?** or **ly:grob-array?** for all interfaces supported by grob *grob*.
- grob::compose-function** *func data* [Función]  
 This creates a callback entity to be stored in a grob property, based on the grob property *data* (which can be plain data, a callback itself, or an unpure-pure-container).  
 Function or unpure-pure-container *func* accepts a grob and a value and returns another value. Depending on the type of *data*, *func* is used for building a grob callback or an unpure-pure-container.
- grob::display-objects** *grob* [Función]  
 Display all objects stored in properties of grob *grob*.
- grob::name** *grob* [Función]  
 Return the name of the grob *grob* as a symbol.

- grob::offset-function** *func data . rest* [Función]  
 This creates a callback entity to be stored in a grob property, based on the grob property *data* (which can be plain data, a callback itself, or an unpure-pure-container).  
 Function *func* accepts a grob and returns a value that is added to the value resulting from *data*. Optional argument *plus* defaults to + but may be changed to allow for using a different underlying accumulation.  
 If *data* is #f or '(), it is not included in the sum.
- grob::rhythmic-location** *grob* [Función]  
 Return a pair consisting of the measure number and moment within the measure of grob *grob*.
- grob::unpure-Y-extent-from-stencil** *pure-function* [Función]  
 The unpure height will come from a stencil whereas the pure height will come from *pure-function*.
- grob::when** *grob* [Función]  
 Return the global timestep (a moment) of grob *grob*.
- ly:grob-alist-chain** *grob global* [Función]  
 Get an alist chain for grob *grob*, with *global* as the global default. If unspecified, *font-defaults* from the layout block is taken.
- ly:grob-array?** *x* [Función]  
 Is *x* a Grob\_array object?
- ly:grob-array->list** *grob-arr* [Función]  
 Return the elements of *grob-arr* as a Scheme list.
- ly:grob-array-length** *grob-arr* [Función]  
 Return the length of *grob-arr*.
- ly:grob-array-ref** *grob-arr index* [Función]  
 Retrieve the *index*th element of *grob-arr*.
- ly:grob-basic-properties** *grob* [Función]  
 Get the immutable properties of *grob*.
- ly:grob-chain-callback** *grob proc sym* [Función]  
 Find the callback that is stored as property *sym* of grob *grob* and chain *proc* to the head of this, meaning that it is called using *grob* and the previous callback's result.
- ly:grob-common-refpoint** *grob other axis* [Función]  
 Find the common refpoint of *grob* and *other* for *axis*.
- ly:grob-common-refpoint-of-array** *grob others axis* [Función]  
 Find the common refpoint of *grob* and *others* (a grob-array) for *axis*.
- ly:grob-default-font** *grob* [Función]  
 Return the default font for grob *grob*.
- grob-elts::X-extent** *grob* [Función]  
 Take the grob *grob*, get its 'elements, calculate their 'X-extent and return the minimum and maximum value as a pair. If 'elements is empty return '(0 . 0)
- ly:grob-extent** *grob refp axis* [Función]  
 Get the extent in *axis* direction of *grob* relative to the grob *refp*.



- ly:grob-get-vertical-axis-group-index** *grob* [Función]  
 Get the index of the vertical axis group the grob *grob* belongs to; return -1 if none is found.
- ly:grob-interfaces** *grob* [Función]  
 Return the interfaces list of grob *grob*.
- ly:grob-layout** *grob* [Función]  
 Get \layout definition from grob *grob*.
- ly:grob-object** *grob sym val* [Función]  
 Return the value of a pointer in grob *grob* of property *sym*. When *sym* is undefined in *grob*, it returns *val* if specified or '()' (end-of-list) otherwise. The kind of properties this taps into differs from regular properties. It is used to store links between grobs, either grobs or grob arrays. For instance, a note head has a **stem** property, the stem grob it belongs to. Just after line breaking, all those grobs are scanned and replaced by their relevant broken versions when applicable.
- ly:grob-original** *grob* [Función]  
 Return the unbroken original grob of *grob*.
- ly:grob-parent** *grob axis* [Función]  
 Get the parent of *grob*. *axis* is 0 for the X-axis, 1 for the Y-axis.
- ly:grob-pq<?** *a b* [Función]  
 Compare two grob priority queue entries. This is an internal function.
- ly:grob-properties?** *x* [Función]  
 Is *x* a `Grob_properties` object?
- ly:grob-property** *grob sym val* [Función]  
 Return the value for property *sym* of *grob*. If no value is found, return *val* or '()' if *val* is not specified.
- ly:grob-property-data** *grob sym* [Función]  
 Return the value for property *sym* of *grob*, but do not process callbacks.
- ly:grob-pure-height** *grob refp beg end val* [Función]  
 Return the pure height of *grob* given reftype *refp*. If no value is found, return *val* or '()' if *val* is not specified.
- ly:grob-pure-property** *grob sym beg end val* [Función]  
 Return the pure value for property *sym* of *grob*. If no value is found, return *val* or '()' if *val* is not specified.
- ly:grob-relative-coordinate** *grob refp axis* [Función]  
 Get the coordinate in *axis* direction of *grob* relative to the grob *refp*.
- ly:grob-robust-relative-extent** *grob refp axis* [Función]  
 Get the extent in *axis* direction of *grob* relative to the grob *refp*, or (0,0) if empty.
- ly:grob-script-priority-less** *a b* [Función]  
 Compare two grobs by script priority. For internal use.
- ly:grob-set-nested-property!** *grob symlist val* [Función]  
 Set nested property *symlist* in grob *grob* to value *val*.
- ly:grob-set-object!** *grob sym val* [Función]  
 Set *sym* in grob *grob* to value *val*.

- ly:grob-set-parent!** *grob axis parent-grob* [Función]  
 Set *parent-grob* the parent of *grob* in axis *axis*.
- ly:grob-set-property!** *grob sym val* [Función]  
 Set *sym* in *grob* to value *val*.
- ly:grob-spanned-rank-interval** *grob* [Función]  
 Returns a pair with the **rank** of the furthest left column and the **rank** of the furthest right column spanned by *grob*.
- ly:grob-staff-position** *sg* [Función]  
 Return the Y-position of *sg* relative to the staff.
- ly:grob-suicide!** *grob* [Función]  
 Kill *grob*.
- ly:grob-system** *grob* [Función]  
 Return the system *grob* of *grob*.
- grob-transformer** *property func* [Función]  
 Create an override value good for applying *func* to either pure or unpure values. *func* is called with the respective *grob* as first argument and the default value (after resolving all callbacks) as the second.
- ly:grob-translate-axis!** *grob d a* [Función]  
 Translate *grob* on axis *a* over distance *d*.
- ly:grob-vertical<?** *a b* [Función]  
 Does *a* lie above *b* on the page?
- ly:gulp-file** *name size* [Función]  
 Read *size* characters from the file *name*, and return its contents in a string. If *size* is undefined, the entire file is read. The file is looked up using the search path.
- ly:gulp-file-utf8** *name size* [Función]  
 Read *size* characters from the file *name*, and return its contents in a string decoded from UTF-8. If *size* is undefined, the entire file is read. The file is looked up using the search path.
- ly:has-glyph-names?** *font-file-name idx* [Función]  
 Does the font for *font-file-name* have glyph names? The optional *idx* argument is useful for TrueType Collections (TTC) and OpenType/CFF collections (OTC) only; it specifies the font index within the TTC/OTC. The default value of *idx* is 0.
- ly:hash-table-keys** *tab* [Función]  
 Return a list of keys in *tab*.
- hook-stencil** *x y staff-space thick blot grob* [Función]  
 Returns a hook-stencil, where *x* determines the horizontal position and *y* determines the basic vertical position. The final stencil is adjusted vertically using *staff-space*, which is **StaffSymbol**'s staff space, and uses *blot*, which is the current '**blot-diameter**'. The stencil's thickness is usually taken from *grob* '**details**', *thick* serves as a fallback value.
- ly:in-event-class?** *ev cl* [Función]  
 Does event *ev* belong to event class *cl*?
- ly:inch** *num* [Función]  
*num* inches.

<b>ly:input-both-locations</b> <i>sip</i>	[Función]
Return input location in <i>sip</i> as (file-name first-line first-column last-line last-column).	
<b>ly:input-file-line-char-column</b> <i>sip</i>	[Función]
Return input location in <i>sip</i> as (file-name line char column).	
<b>ly:input-location?</b> <i>x</i>	[Función]
Is <i>x</i> a Input object?	
<b>ly:input-message</b> <i>sip msg rest</i>	[Función]
Print <i>msg</i> as a GNU compliant error message, pointing to the location in <i>sip</i> . <i>msg</i> is interpreted similar to <b>format</b> 's argument, using <i>rest</i> .	
<b>ly:input-warning</b> <i>sip msg rest</i>	[Función]
Print <i>msg</i> as a GNU compliant warning message, pointing to the location in <i>sip</i> . <i>msg</i> is interpreted similar to <b>format</b> 's argument, using <i>rest</i> .	
<b>ly:interpret-music-expression</b> <i>mus ctx</i>	[Función]
Interpret the music expression <i>mus</i> in the global context <i>ctx</i> . The context is returned in its final state.	
<b>interval-center</b> <i>x</i>	[Función]
Center the number-pair <i>x</i> , if an interval.	
<b>interval-index</b> <i>interval dir</i>	[Función]
Interpolate <i>interval</i> between between left ( <i>dir</i> =-1) and right ( <i>dir</i> =+1).	
<b>interval-length</b> <i>x</i>	[Función]
Length of the number-pair <i>x</i> , if an interval.	
<b>ly:intlog2</b> <i>d</i>	[Función]
The 2-logarithm of 1/ <i>d</i> .	
<b>invalidate-alterations</b> <i>context</i>	[Función]
Invalidate alterations in <i>context</i> .	
Elements of ' <b>localAlterations</b> corresponding to local alterations of the key signature have the form '((octave . notename) . (alter barnum . measurepos)). Replace them with a version where <b>alter</b> is set to ' <b>clef</b> to force a repetition of accidentals.	
Entries that conform with the current key signature are not invalidated.	
<b>ly:item?</b> <i>g</i>	[Función]
Is <i>g</i> an Item object?	
<b>ly:item-break-dir</b> <i>it</i>	[Función]
The break status direction of item <i>it</i> . -1 means end of line, 0 unbroken, and 1 beginning of line.	
<b>ly:item-get-column</b> <i>it</i>	[Función]
Return the PaperColumn or NonMusicalPaperColumn associated with this Item.	
<b>ly:iterator?</b> <i>x</i>	[Función]
Is <i>x</i> a Music_iterator object?	
<b>layout-line-thickness</b> <i>grob</i>	[Función]
Get the line thickness of the <i>grob</i> 's corresponding layout.	

<b>layout-set-absolute-staff-size</b> <i>sz</i>	[Función]
Set the absolute staff size inside of a <code>\layout{}</code> block. <i>sz</i> is in points.	
<b>layout-set-staff-size</b> <i>sz</i>	[Función]
Set the staff size inside of a <code>\layout{}</code> block. <i>sz</i> is in points.	
<b>ly:length</b> <i>x y</i>	[Función]
Calculates magnitude of given vector. With one argument, <i>x</i> is a number pair indicating the vector. With two arguments, <i>x</i> and <i>y</i> specify the respective coordinates.	
<b>ly:lily-lexer?</b> <i>x</i>	[Función]
Is <i>x</i> a <code>Lily_lexer</code> object?	
<b>ly:lily-parser?</b> <i>x</i>	[Función]
Is <i>x</i> a <code>Lily_parser</code> object?	
<b>lilypond-main</b> <i>files</i>	[Función]
Entry point for LilyPond.	
<b>ly:line-interface::line</b> <i>grob startx starty endx endy</i>	[Función]
Make a line using layout information from grob <i>grob</i> .	
<b>list-insert-separator</b> <i>lst between</i>	[Función]
Create new list, inserting <i>between</i> between elements of <i>lst</i> .	
<b>list-join</b> <i>lst intermediate</i>	[Función]
Put <i>intermediate</i> between all elts of <i>lst</i> .	
<b>ly:listened-event-class?</b> <i>disp cl</i>	[Función]
Does <i>disp</i> listen to any event type in the list <i>cl</i> ?	
<b>ly:listened-event-types</b> <i>disp</i>	[Función]
Return a list of all event types that <i>disp</i> listens to.	
<b>ly:listener?</b> <i>x</i>	[Función]
Is <i>x</i> a <code>Listener</code> object?	
<b>lookup-markup-command</b> <i>code</i>	[Función]
Return (FUNCTION . SIGNATURE) for a markup command, or return <code>#f</code>	
<b>lyric-text::print</b> <i>grob</i>	[Función]
Allow interpretation of tildes as lyric tying marks.	
<b>ly:make-book</b> <i>paper header scores</i>	[Función]
Make a <code>\book</code> of <i>paper</i> and <i>header</i> (which may be <code>#f</code> as well) containing <code>\scores</code> .	
<b>ly:make-book-part</b> <i>scores</i>	[Función]
Make a <code>\bookpart</code> containing <code>\scores</code> .	
<b>make-bow-stencil</b> <i>start stop thickness angularity bow-height orientation</i>	[Función]
Create a bow stencil. It starts at point <i>start</i> , ends at point <i>stop</i> . <i>thickness</i> is the thickness of the bow. The higher the value of number <i>angularity</i> , the more angular the shape of the bow. <i>bow-height</i> determines the height of the bow. <i>orientation</i> determines, whether the bow is concave or convex. Both variables are supplied to support independent usage.	
Done by calculating a horizontal unit-bow first, then moving all control-points to the correct positions. Limitation: s-curves are currently not supported.	

<b>make-c-time-signature-markup</b> <i>fraction</i>	[Función]
Make markup for the ‘C’ time signature style.	
<b>make-circle-stencil</b> <i>radius thickness fill</i>	[Función]
Make a circle of radius <i>radius</i> and thickness <i>thickness</i> .	
<b>make-clef-set</b> <i>clef-name</i>	[Función]
Generate the clef setting commands for a clef with name <i>clef-name</i> .	
<b>make-connected-line</b> <i>points grob</i>	[Función]
Takes a list of points, <i>points</i> . Returns a line connecting <i>points</i> , using <code>ly:line-interface::line</code> , gets layout information from <i>grob</i>	
<b>make-connected-path-stencil</b> <i>pointlist thickness x-scale y-scale connect fill</i>	[Función]
Make a connected path described by the list <i>pointlist</i> , beginning at point ‘(0 . 0)’, with thickness <i>thickness</i> , and scaled by <i>x-scale</i> in the X direction and <i>y-scale</i> in the Y direction. <i>connect</i> and <i>fill</i> are boolean arguments that specify if the path should be connected or filled, respectively.	
<b>ly:make-context-mod</b> <i>mod-list</i>	[Función]
Creates a context modification, optionally initialized via the list of modifications <i>mod-list</i> .	
<b>make-cue-clef-set</b> <i>clef-name</i>	[Función]
Generate the clef setting commands for a cue clef with name <i>clef-name</i> .	
<b>make-cue-clef-unset</b>	[Función]
Reset the clef settings for a cue clef.	
<b>ly:make-dispatcher</b>	[Función]
Return a newly created dispatcher.	
<b>ly:make-duration</b> <i>length dotcount num den</i>	[Función]
<i>length</i> is the negative logarithm (base 2) of the duration: 1 is a half note, 2 is a quarter note, 3 is an eighth note, etc. The number of dots after the note is given by the optional argument <i>dotcount</i> .	
The duration factor is optionally given by integers <i>num</i> and <i>den</i> , alternatively by a single rational number.	
A duration is a musical duration, i.e., a length of time described by a power of two (whole, half, quarter, etc.) and a number of augmentation dots.	
<b>make-duration-of-length</b> <i>moment</i>	[Función]
Make duration of the given <i>moment</i> length.	
<b>make-ellipse-stencil</b> <i>x-radius y-radius thickness fill</i>	[Función]
Make an ellipse of x radius <i>x-radius</i> , y radius <i>y-radius</i> , and thickness <i>thickness</i> with fill defined by <i>fill</i> .	
<b>make-filled-box-stencil</b> <i>xext yext</i>	[Función]
Make a filled box.	
<b>ly:make-global-context</b> <i>output-def</i>	[Función]
Set up a global interpretation context, using the output block <i>output-def</i> . The context is returned.	

- ly:make-global-translator** *global* [Función]  
 Create a translator group and connect it to the global context *global*. The translator group is returned.
- make-glyph-time-signature-markup** *style fraction* [Función]  
 Make markup for a symbolic time signature. If the music font does not have a glyph for the requested style and fraction, issue a warning and make a numbered time signature instead.
- ly:make-grob-properties** *alist* [Función]  
 This packages the given property list *alist* in a grob property container stored in a context property with the name of a grob.
- make-grob-property-override** *grob gprop val* [Función]  
 Make a Music expression that overrides *gprop* to *val* in *grob*. This is a `\temporary \override`, making it possible to `\revert` to any previous value afterwards.
- make-grob-property-revert** *grob gprop* [Función]  
 Revert the grob property *gprop* for *grob*.
- make-grob-property-set** *grob gprop val* [Función]  
 Make a Music expression that overrides a *gprop* to *val* in *grob*. Does a pop first, i.e. this is not a `\temporary \override`.
- make-harmonic** *mus* [Función]  
 Convert music variable *mus* to harmonics.
- make-line-stencil** *width startx starty endx endy* [Función]  
 Make a line stencil of given linewidth and set its extents accordingly.
- ly:make-listener** *callback* [Función]  
 This is a compatibility wrapper for creating a "listener" for use with **ly:add-listener** from a *callback* taking a single argument. Since listeners are equivalent to callbacks, this is no longer needed.
- make-modal-inverter** *around to scale* [Función]  
 Wrapper function for inverter-factory
- make-modal-transposer** *from to scale* [Función]  
 Wrapper function for transposer-factory.
- ly:make-moment** *m g gn gd* [Función]  
 Create the moment with rational main timing *m*, and optional grace timing *g*.  
 A *moment* is a point in musical time. It consists of a pair of rationals (*m*, *g*), where *m* is the timing for the main notes, and *g* the timing for grace notes. In absence of grace notes, *g* is zero.  
 For compatibility reasons, it is possible to write two numbers specifying numerator and denominator instead of the rationals. These forms cannot be mixed, and the two-argument form is disambiguated by the sign of the second argument: if it is positive, it can only be a denominator and not a grace timing.
- ly:make-music** *props* [Función]  
 Make a C++ Music object and initialize it with *props*.  
 This function is for internal use and is only called by **make-music**, which is the preferred interface for creating music objects.

**make-music** *name . music-properties* [Función]

Create a music object of given name, and set its properties according to **music-properties**, a list of alternating property symbols and values. E.g:

```
(make-music 'OverrideProperty
  'symbol 'Stem
  'grob-property 'thickness
  'grob-value (* 2 1.5))
```

Instead of a successive symbol and value, an entry in the list may also be an alist or a music object in which case its elements, respectively its *mutable* property list (properties not inherent to the type of the music object) will get taken.

The argument list will be interpreted left-to-right, so later entries override earlier ones.

**ly:make-music-function** *signature func* [Función]

Make a function to process music, to be used for the parser. *func* is the function, and *signature* describes its arguments. *signature*'s cdr is a list containing either **ly:music?** predicates or other type predicates. Its car is the syntax function to call.

**ly:make-music-relative!** *music pitch* [Función]

Make *music* relative to *pitch*, return final pitch.

**ly:make-output-def** [Función]

Make an output definition.

**make-oval-stencil** *x-radius y-radius thickness fill* [Función]

Make an oval from two Bezier curves, of x radius *x-radius*, y radius *y-radius*, and thickness *thickness* with fill defined by *fill*.

**ly:make-page-label-marker** *label* [Función]

Return page marker with label *label*.

**ly:make-page-permission-marker** *symbol permission* [Función]

Return page marker with page breaking and turning permissions.

**ly:make-pango-description-string** *chain size* [Función]

Make a PangoFontDescription string for the property alist *chain* at size *size*.

**ly:make-paper-outputter** *port alist default-callback* [Función]

Create an outputter dumping to *port*. *alist* should map symbols to procedures. See **output-ps.scm** for an example. If *default\_callback* is given, it is called for unsupported expressions

**make-part-combine-context-changes** *state-machine split-list* [Función]

Generate a sequence of part combiner context changes from a split list

**make-part-combine-marks** *state-machine split-list* [Función]

Generate a sequence of part combiner events from a split list

**make-partial-ellipse-stencil** *x-radius y-radius start-angle end-angle  
thick connect fill* [Función]

Create an elliptical arc *x-radius* is the X radius of the arc. *y-radius* is the Y radius of the arc. *start-angle* is the starting angle of the arc in degrees. *end-angle* is the ending angle of the arc in degrees. *thick* is the thickness of the line. *connect* is a boolean flag indicating if the end should be connected to the start by a line. *fill* is a boolean flag indicating if the shape should be filled.

**make-path-stencil** *path thickness x-scale y-scale fill* [Función]

Make a stencil based on the path described by the list *path*, with thickness *thickness*, and scaled by *x-scale* in the X direction and *y-scale* in the Y direction. *fill* is a boolean argument that specifies if the path should be filled. Valid path commands are: moveto rmoveto lineto rlineto curveto rcurveto closepath, and their standard SVG single letter equivalents: M m L l C c Z z.

**ly:make-pitch** *octave note alter* [Función]

*octave* is specified by an integer, zero for the octave containing middle C. *note* is a number indexing the global default scale, with 0 corresponding to pitch C and 6 usually corresponding to pitch B. Optional *alter* is a rational number of 200-cent whole tones for alteration.

**ly:make-prob** *type init rest* [Función]

Create a Prob object.

**make-repeat** *name times main alts* [Función]

Create a repeat music expression, with all properties initialized properly.

**ly:make-rotation** *angle center* [Función]

Make a transform rotating by *angle* in degrees. If *center* is given as a pair of coordinates, it is the center of the rotation, otherwise the rotation is around (0 . 0).

**ly:make-scale** *steps* [Función]

Create a scale. The argument is a vector of rational numbers, each of which represents the number of 200 cent tones of a pitch above the tonic.

**ly:make-scaling** *scale scaley* [Función]

Create a scaling transform from argument *scale* and optionally *scaley*. When both arguments are given, they must be real and give the scale in x and y direction. If only *scale* is given, it may also be complex to indicate a scaled rotation in the manner of complex number rotations, or a pair of reals for specifying different scales in x and y direction like with the first calling convention.

**ly:make-score** *music* [Función]

Return score with *music* encapsulated in it.

**make-semitone->pitch** *pitches* [Función]

Convert *pitches*, an unordered list of note values covering (after disregarding octaves) all absolute pitches in need of conversion, into a function converting semitone numbers (absolute pitch missing enharmonic information) back into note values.

For a key signature without accidentals

c cis d es e f fis g gis a bes b

might be a good choice, covering Bb major to A major and their parallel keys, and melodic/harmonic C minor to A minor.

**ly:make-spring** *ideal min-dist* [Función]

Make a spring. *ideal* is the ideal distance of the spring, and *min-dist* is the minimum distance.

**ly:make-stencil** *expr xext yext* [Función]

Stencils are device independent output expressions. They carry two pieces of information:

1. A specification of how to print this object. This specification is processed by the output backends, for example `scm/output-ps.scm`.
2. The vertical and horizontal extents of the object, given as pairs. If an extent is unspecified (or if you use `empty-interval` as its value), it is taken to be empty.



- make-stencil-boxer** *thickness padding callback* [Función]  
 Return function that adds a box around the grob passed as argument.
- make-stencil-circler** *thickness padding callback* [Función]  
 Return function that adds a circle around the grob passed as argument.
- ly:make-stream-event** *cl proplist* [Función]  
 Create a stream event of class *cl* with the given mutable property list.
- make-tmpfile** *basename* [Función]  
 Returns a temp file as port. If *basename* is *#f*, a file under *\$TMPDIR* is created.
- ly:make-transform** *xx yx xy yy x0 y0* [Función]  
 Create a transform. Without options, it is an identity transform. Given four arguments *xx*, *yx*, *xy*, and *yy*, it is a linear transform, given six arguments (with *x0* and *y0* last), it is an affine transform. Transforms can be called as functions on other transforms (concatening them) or on points given either as complex number or real number pair. See also **ly:make-rotation**, **ly:make-scaling**, and **ly:make-translation**.
- ly:make-translation** *x y* [Función]  
 Make a transform translating by *x* and *y*. If only *x* is given, it can also be a complex number or a pair of numbers indicating the offset to use.
- make-transparent-box-stencil** *xext yext* [Función]  
 Make a transparent box.
- ly:make-unpure-pure-container** *unpure pure* [Función]  
 Make an unpure-pure container. *unpure* should be an unpure expression, and *pure* should be a pure expression. If *pure* is omitted, the value of *unpure* will be used twice, except that a callback is given two extra arguments that are ignored for the sake of pure calculations.
- map-selected-alist-keys** *function keys alist* [Función]  
 Return *alist* with *function* applied to all of the values in list *keys*.  
 For example:  

```
guile> (map-selected-alist-keys - '(a b) '((a . 1) (b . -2) (c . 3) (d . 4)))
((a . -1) (b . 2) (c . 3) (d . 4))
```
- map-some-music** *map? music* [Función]  
 Walk through *music*, transform all elements calling *map?* and only recurse if this returns *#f*. *elements* or *articulations* that are not music expressions are discarded: this allows some amount of filtering.  
*map-some-music* may overwrite the original *music*.
- markup-command-list?** *x* [Función]  
 Determine if 'x' is a markup command list, ie. a list composed of a markup list function and its arguments.
- markup-list?** *arg* [Función]  
 Return a true value if 'x' is a list of markups or markup command lists.
- measure-counter::text** *grob* [Función]  
 A number for a measure count. Broken measures are numbered in parentheses. When the counter spans several measures (like with compressed multi-measure rests), it displays a measure range.

- mensural-flag** *grob* [Función]  
 Mensural flags: Create the flag stencil by loading the glyph from the font. Flags are always aligned with staff lines, so we need to check the end point of the stem: For stems ending on staff lines, use different flags than for notes between staff lines. The idea is that flags are always vertically aligned with the staff lines, regardless of whether the note head is on a staff line or between two staff lines. In other words, the inner end of a flag always touches a staff line.
- ly:message** *str rest* [Función]  
 A Scheme callable function to issue the message *str*. The message is formatted with **format** and *rest*.
- midi-program** *instrument* [Función]  
 Return the program of the instrument.
- ly:minimal-breaking** *pb* [Función]  
 Break (pages and lines) the **Paper\_book** object *pb* without looking for optimal spacing: stack as many lines on a page before moving to the next one.
- ly:mm** *num* [Función]  
*num* mm.
- mmrest-of-length** *mus* [Función]  
 Create a multi-measure rest of exactly the same length as *mus*.
- modern-straight-flag** *grob* [Función]  
 Modern straight flag style (for composers like Stockhausen, Boulez, etc.). The angles are 18 and 22 degrees and thus smaller than for the ancient style of Bach, etc.
- ly:module->alist** *mod* [Función]  
 Dump the contents of module *mod* as an alist.
- ly:module-copy** *dest src* [Función]  
 Copy all bindings from module *src* into *dest*.
- ly:modules-lookup** *modules sym def* [Función]  
 Look up *sym* in the list *modules*, returning the first occurrence. If not found, return *def* or **#f** if *def* isn't specified.
- ly:moment?** *x* [Función]  
 Is *x* a **Moment** object?
- ly:moment<?** *a b* [Función]  
 Compare two moments.
- ly:moment-add** *a b* [Función]  
 Add two moments.
- ly:moment-div** *a b* [Función]  
 Divide two moments.
- ly:moment-grace** *mom* [Función]  
 Extract grace timing as a rational number from *mom*.
- ly:moment-grace-denominator** *mom* [Función]  
 Extract denominator from grace timing.

<code>ly:moment-grace-numerator <i>mom</i></code>	[Función]
Extract numerator from grace timing.	
<code>ly:moment-main <i>mom</i></code>	[Función]
Extract main timing as a rational number from <i>mom</i> .	
<code>ly:moment-main-denominator <i>mom</i></code>	[Función]
Extract denominator from main timing.	
<code>ly:moment-main-numerator <i>mom</i></code>	[Función]
Extract numerator from main timing.	
<code>ly:moment-mod <i>a b</i></code>	[Función]
Modulo of two moments.	
<code>ly:moment-mul <i>a b</i></code>	[Función]
Multiply two moments.	
<code>ly:moment-sub <i>a b</i></code>	[Función]
Subtract two moments.	
<code>ly:music? <i>obj</i></code>	[Función]
Is <i>obj</i> a music object?	
<code>music-&gt;make-music <i>obj</i></code>	[Función]
Generate an expression that, once evaluated, may return an object equivalent to <i>obj</i> , that is, for a music expression, a (make-music ...) form.	
<code>music-clone <i>music . music-properties</i></code>	[Función]
Clone <i>music</i> and set properties according to <i>music-properties</i> , a list of alternating property symbols and values:	
(music-clone start-span 'span-direction STOP)	
Only properties that are not overridden by <i>music-properties</i> are actually fully cloned.	
<code>ly:music-compress <i>m factor</i></code>	[Función]
Compress music object <i>m</i> by scale <i>factor</i> .	
<code>ly:music-deep-copy <i>m origin</i></code>	[Función]
Copy <i>m</i> and all sub expressions of <i>m</i> . <i>m</i> may be an arbitrary type; cons cells and music are copied recursively. If <i>origin</i> is given, it is used as the origin for one level of music by calling <code>ly:set-origin!</code> on the copy.	
<code>ly:music-duration-compress <i>mus fact</i></code>	[Función]
Compress <i>mus</i> by factor <i>fact</i> , which is a Moment.	
<code>ly:music-duration-length <i>mus</i></code>	[Función]
Extract the duration field from <i>mus</i> and return the length.	
<code>music-filter <i>pred? music</i></code>	[Función]
Filter out music expressions that do not satisfy <i>pred?</i> .	
<code>ly:music-function? <i>x</i></code>	[Función]
Is <i>x</i> a Music_function object?	
<code>ly:music-function-extract <i>x</i></code>	[Función]
Return the Scheme function inside <i>x</i> .	

<b>ly:music-function-signature</b> <i>x</i>	[Función]
Return the function signature inside <i>x</i> .	
<b>music-is-of-type?</b> <i>mus type</i>	[Función]
Does <i>mus</i> belong to the music class <i>type</i> ?	
<b>ly:music-length</b> <i>mus</i>	[Función]
Get the length of music expression <i>mus</i> and return it as a <b>Moment</b> object.	
<b>ly:music-list?</b> <i>lst</i>	[Función]
Is <i>lst</i> a list of music objects?	
<b>music-map</b> <i>function music</i>	[Función]
Apply <i>function</i> to <i>music</i> and all of the music it contains. First it recurses over the children, then the function is applied to <i>music</i> .	
<b>ly:music-mutable-properties</b> <i>mus</i>	[Función]
Return an alist containing the mutable properties of <i>mus</i> . The immutable properties are not available, since they are constant and initialized by the <b>make-music</b> function.	
<b>ly:music-output?</b> <i>x</i>	[Función]
Is <i>x</i> a <b>Music_output</b> object?	
<b>music-pitches</b> <i>music</i>	[Función]
Return a list of all pitches from <i>music</i> .	
<b>ly:music-property</b> <i>mus sym val</i>	[Función]
Return the value for property <i>sym</i> of music expression <i>mus</i> . If no value is found, return <i>val</i> or '()' if <i>val</i> is not specified.	
<b>music-selective-filter</b> <i>descend? pred? music</i>	[Función]
Recursively filter out music expressions that do not satisfy <i>pred?</i> , but refrain from filtering the subexpressions of music that does not satisfy <i>descend?</i> .	
<b>music-selective-map</b> <i>descend? function music</i>	[Función]
Apply <i>function</i> recursively to <i>music</i> , but refrain from mapping subexpressions of music that does not satisfy <i>descend?</i> .	
<b>music-separator?</b> <i>m</i>	[Función]
Is <i>m</i> a separator?	
<b>ly:music-set-property!</b> <i>mus sym val</i>	[Función]
Set property <i>sym</i> in music expression <i>mus</i> to <i>val</i> .	
<b>ly:music-start</b> <i>mus</i>	[Función]
Get the start of music expression <i>mus</i> and return it as a <b>Moment</b> object.	
<b>ly:music-transpose</b> <i>m p</i>	[Función]
Transpose <i>m</i> such that central C is mapped to <i>p</i> . Return <i>m</i> .	
<b>music-type-predicate</b> <i>types</i>	[Función]
Returns a predicate function that can be used for checking music to have one of the types listed in <i>types</i> .	
<b>neo-modern-accidental-rule</b> <i>context pitch barnum measurepos</i>	[Función]
An accidental rule that typesets an accidental if it differs from the key signature <i>and</i> does not directly follow a note on the same staff line. This rule should not be used alone because it does neither look at bar lines nor different accidentals at the same note name.	

<b>no-flag</b> <i>grob</i>	[Función]
No flag: Simply return empty stencil.	
<b>normal-flag</b> <i>grob</i>	[Función]
Create a default flag.	
<b>note-column::main-extent</b> <i>grob</i>	[Función]
Return extent of the noteheads in the 'main column' (i.e., excluding any suspended noteheads), or extent of the rest (if there are no heads).	
<b>ly:note-column-accidentals</b> <i>note-column</i>	[Función]
Return the <code>AccidentalPlacement</code> grob from <i>note-column</i> if any, or <code>SCM_EOL</code> otherwise.	
<b>ly:note-column-dot-column</b> <i>note-column</i>	[Función]
Return the <code>DotColumn</code> grob from <i>note-column</i> if any, or <code>SCM_EOL</code> otherwise.	
<b>ly:note-head::stem-attachment</b> <i>font-metric glyph-name direction</i>	[Función]
Get attachment in <i>font-metric</i> for attaching a stem to notehead <i>glyph-name</i> in the direction <i>direction</i> (default UP).	
<b>note-name-&gt;markup</b> <i>pitch lowercase?</i>	[Función]
Return pitch markup for <i>pitch</i> , including accidentals printed as glyphs. If <i>lowercase?</i> is set to false, the note names are capitalized.	
<b>note-name-&gt;string</b> <i>pitch . language</i>	[Función]
Return pitch string for <i>pitch</i> , without accidentals or octaves. Current input language is used for pitch names, except if an other <i>language</i> is specified.	
<b>note-to-cluster</b> <i>music</i>	[Función]
Replace <code>NoteEvents</code> by <code>ClusterNoteEvents</code> .	
<b>ly:number-&gt;string</b> <i>s</i>	[Función]
Convert <i>s</i> to a string without generating many decimals.	
<b>number-format</b> <i>number-type num . custom-format</i>	[Función]
Print NUM accordingly to the requested NUMBER-TYPE. Choices include <code>roman-lower</code> (by default), <code>roman-upper</code> , <code>arabic</code> and <code>custom</code> . In the latter case, CUSTOM-FORMAT must be supplied and will be applied to NUM.	
<b>offset-fret</b> <i>fret-offset diagram-definition</i>	[Función]
Add <i>fret-offset</i> to each fret indication in <i>diagram-definition</i> and return the resulting verbose <i>fret-diagram-definition</i> .	
<b>offsetter</b> <i>property offsets</i>	[Función]
Apply <i>offsets</i> to the default values of <i>property</i> of <i>grob</i> . Offsets are restricted to immutable properties and values of type <code>number</code> , <code>number-pair</code> , or <code>number-pair-list</code> .	
<b>old-straight-flag</b> <i>grob</i>	[Función]
Old straight flag style (for composers like Bach). The angles of the flags are both 45 degrees.	
<b>ly:one-line-auto-height-breaking</b> <i>pb</i>	[Función]
Put each score on a single line, and put each line on its own page. Modify the paper-width setting so that every page is wider than the widest line. Modify the paper-height setting to fit the height of the tallest line.	
<b>ly:one-line-breaking</b> <i>pb</i>	[Función]
Put each score on a single line, and put each line on its own page. Modify the paper-width setting so that every page is wider than the widest line.	

- ly:one-page-breaking** *pb* [Función]  
 Put each score on a single page. The paper-height settings are modified so each score fits on one page, and the height of the page matches the height of the full score.
- ly:optimal-breaking** *pb* [Función]  
 Optimally break (pages and lines) the **Paper\_book** object *pb* to minimize badness in both vertical and horizontal spacing.
- ly:option-usage** *port* [Función]  
 Print **ly:set-option** usage. Optional *port* argument for the destination defaults to current output port.
- ly:otf->cff** *otf-file-name idx* [Función]  
 Convert the contents of an OTF file to a CFF file, returning it as a string. The optional *idx* argument is useful for OpenType/CFF collections (OTC) only; it specifies the font index within the OTC. The default value of *idx* is 0.
- ly:otf-font?** *font* [Función]  
 Is *font* an OpenType font?
- ly:otf-font-glyph-info** *font glyph* [Función]  
 Given the font metric *font* of an OpenType font, return the information about named glyph *glyph* (a string).
- ly:otf-font-table-data** *font tag* [Función]  
 Extract a table *tag* from *font*. Return empty string for non-existent *tag*.
- ly:otf-glyph-count** *font* [Función]  
 Return the number of glyphs in *font*.
- ly:otf-glyph-list** *font* [Función]  
 Return a list of glyph names for *font*.
- ly:output-def?** *x* [Función]  
 Is *x* a **Output\_def** object?
- ly:output-def-clone** *def* [Función]  
 Clone output definition *def*.
- ly:output-def-lookup** *def sym val* [Función]  
 Return the value of *sym* in output definition *def* (e.g., **\paper**). If no value is found, return *val* or '()' if *val* is undefined.
- ly:output-def-parent** *def* [Función]  
 Return the parent output definition of *def*.
- ly:output-def-scope** *def* [Función]  
 Return the variable scope inside *def*.
- ly:output-def-set-variable!** *def sym val* [Función]  
 Set an output definition *def* variable *sym* to *val*.
- ly:output-description** *output-def* [Función]  
 Return the description of translators in *output-def*.
- ly:output-find-context-def** *output-def context-name* [Función]  
 Return an alist of all context defs (matching *context-name* if given) in *output-def*.

<code>ly:output-formats</code>	[Función]
Formats passed to <code>--format</code> as a list of strings, used for the output.	
<code>output-module? module</code>	[Función]
Returns <code>#t</code> if <i>module</i> belongs to an output module usually carrying context definitions ( <code>\midi</code> or <code>\layout</code> ).	
<code>ly:outputter-close outputter</code>	[Función]
Close port of <i>outputter</i> .	
<code>ly:outputter-dump-stencil outputter stencil</code>	[Función]
Dump stencil <i>expr</i> onto <i>outputter</i> .	
<code>ly:outputter-dump-string outputter str</code>	[Función]
Dump <i>str</i> onto <i>outputter</i> .	
<code>ly:outputter-output-scheme outputter expr</code>	[Función]
Output <i>expr</i> to the paper outputter.	
<code>ly:outputter-port outputter</code>	[Función]
Return output port for <i>outputter</i> .	
<code>oval-stencil stencil thickness x-padding y-padding</code>	[Función]
Add an oval around <i>stencil</i> , padded by the padding pair, producing a new stencil.	
<code>override-head-style heads style</code>	[Función]
Override style for <i>heads</i> to <i>style</i> .	
<code>override-time-signature-setting time-signature setting</code>	[Función]
Override the time signature settings for the context in <i>time-signature</i> , with the new setting alist <i>setting</i> .	
<code>ly:page-marker? x</code>	[Función]
Is <i>x</i> a <code>Page_marker</code> object?	
<code>ly:page-turn-breaking pb</code>	[Función]
Optimally break (pages and lines) the <code>Paper_book</code> object <i>pb</i> such that page turns only happen in specified places, returning its pages.	
<code>ly:pango-font? f</code>	[Función]
Is <i>f</i> a pango font?	
<code>ly:pango-font-physical-fonts f</code>	[Función]
Return alist of ( <code>ps-name file-name font-index</code> ) lists for Pango font <i>f</i> .	
<code>pango-pf-file-name pango-pf</code>	[Función]
Return the file-name of the pango physical font <i>pango-pf</i> .	
<code>pango-pf-font-name pango-pf</code>	[Función]
Return the font-name of the pango physical font <i>pango-pf</i> .	
<code>pango-pf-fontindex pango-pf</code>	[Función]
Return the fontindex of the pango physical font <i>pango-pf</i> .	
<code>ly:paper-book? x</code>	[Función]
Is <i>x</i> a <code>Paper_book</code> object?	

<code>ly:paper-book-header</code> <i>pb</i>	[Función]
Return the header definition ( <code>\header</code> ) in <code>Paper_book</code> object <i>pb</i> .	
<code>ly:paper-book-pages</code> <i>pb</i>	[Función]
Return pages in <code>Paper_book</code> object <i>pb</i> .	
<code>ly:paper-book-paper</code> <i>pb</i>	[Función]
Return the paper output definition ( <code>\paper</code> ) in <code>Paper_book</code> object <i>pb</i> .	
<code>ly:paper-book-performances</code> <i>pb</i>	[Función]
Return performances in <code>Paper_book</code> object <i>pb</i> .	
<code>ly:paper-book-scopes</code> <i>pb</i>	[Función]
Return scopes in <code>Paper_book</code> object <i>pb</i> .	
<code>ly:paper-book-systems</code> <i>pb</i>	[Función]
Return systems in <code>Paper_book</code> object <i>pb</i> .	
<code>ly:paper-column::break-align-width</code> <i>col align-syms</i>	[Función]
Determine the extent along the X-axis of a grob used for break-alignment organized by column <i>col</i> . The grob is specified by <i>align-syms</i> , which contains either a single <code>break-align-symbol</code> or a list of such symbols.	
<code>ly:paper-column::print</code>	[Función]
Optional stencil for <code>PaperColumn</code> or <code>NonMusicalPaperColumn</code> . Draws the rank number of each column, its moment in time, a blue arrow showing the ideal distance, and a red arrow showing the minimum distance between columns.	
<code>ly:paper-fonts</code> <i>def</i>	[Función]
Return a list containing the fonts from output definition <i>def</i> (e.g., <code>\paper</code> ).	
<code>ly:paper-get-font</code> <i>def chain</i>	[Función]
Find a font metric in output definition <i>def</i> satisfying the font-qualifiers in alist chain <i>chain</i> , and return it. (An alist chain is a list of alists, containing grob properties.)	
<code>ly:paper-get-number</code> <i>def sym</i>	[Función]
Return the value of variable <i>sym</i> in output definition <i>def</i> as a double.	
<code>ly:paper-outputscales</code> <i>def</i>	[Función]
Return the output-scale for output definition <i>def</i> .	
<code>ly:paper-score-paper-systems</code> <i>paper-score</i>	[Función]
Return vector of <code>paper_system</code> objects from <i>paper-score</i> .	
<code>ly:paper-system?</code> <i>obj</i>	[Función]
Is <i>obj</i> a C++ Prob object of type <code>paper-system</code> ?	
<code>ly:paper-system-minimum-distance</code> <i>sys1 sys2</i>	[Función]
Measure the minimum distance between these two paper-systems, using their stored skylines if possible and falling back to their extents otherwise.	
<code>parenthesize-stencil</code> <i>stencil half-thickness width angularity padding</i>	[Función]
Add parentheses around <i>stencil</i> , returning a new stencil.	
<code>ly:parse-file</code> <i>name</i>	[Función]
Parse a single .ly file. Upon failure, throw <code>ly-file-failed</code> key.	



- ly:parse-init** *name* [Función]  
 Parse the init file *name*.
- ly:parse-string-expression** *parser-smob ly-code filename line* [Función]  
 Parse the string *ly-code* with *parser-smob*. Return the contained music expression. *filename* and *line* are optional source indicators.
- parse-terse-string** *terse-definition* [Función]  
 Parse a **fret-diagram-terse** definition string *terse-definition* and return a marking list, which can be used with a fretboard grob.
- ly:parsed-undead-list!** [Función]  
 Return the list of objects that have been found live that should have been dead, and clear that list.
- ly:parser-clear-error** *parser* [Función]  
 Clear error flag for *parser*, defaulting to current parser.
- ly:parser-clone** *closures location* [Función]  
 Return a clone of current parser. An association list of port positions to closures can be specified in *closures* in order to have **\$** and **#** interpreted in their original lexical environment. If *location* is a valid location, it becomes the source of all music expressions inside.
- ly:parser-define!** *symbol val* [Función]  
 Bind *symbol* to *val* in current parser's module.
- ly:parser-error** *msg input* [Función]  
 Display an error message and make current parser fail. Without a current parser, trigger an ordinary error.
- ly:parser-has-error?** *parser* [Función]  
 Does *parser* (defaulting to current parser) have an error flag?
- ly:parser-include-string** *ly-code* [Función]  
 Include the string *ly-code* into the input stream for current parser. Can only be used in immediate Scheme expressions (**\$** instead of **#**).
- ly:parser-lookup** *symbol* [Función]  
 Look up *symbol* in current parser's module. Return '() if not defined.
- ly:parser-output-name** *parser* [Función]  
 Return the base name of the output file. If *parser* is left off, use currently active parser.
- ly:parser-parse-string** *parser-smob ly-code* [Función]  
 Parse the string *ly-code* with *parser-smob*. Upon failure, throw **ly-file-failed** key.
- ly:parser-set-note-names** *names* [Función]  
 Replace current note names in parser. *names* is an alist of symbols. This only has effect if the current mode is notes.
- percussion?** *instrument* [Función]  
 Return **#t** if the instrument should use MIDI channel 9.
- ly:performance-headers** *performance* [Función]  
 Return the list of headers with the innermost first.

<b>ly:performance-write</b> <i>performance filename name</i>	[Función]
Write <i>performance</i> to <i>filename</i> storing <i>name</i> as the name of the performance in the file metadata.	
<b>ly:pitch?</b> <i>x</i>	[Función]
Is <i>x</i> a Pitch object?	
<b>ly:pitch&lt;?</b> <i>p1 p2</i>	[Función]
Is <i>p1</i> lexicographically smaller than <i>p2</i> ?	
<b>ly:pitch-alteration</b> <i>pp</i>	[Función]
Extract the alteration from pitch <i>pp</i> .	
<b>ly:pitch-diff</b> <i>pitch root</i>	[Función]
Return pitch <i>delta</i> such that <i>root</i> transposed by <i>delta</i> equals <i>pitch</i> .	
<b>ly:pitch-negate</b> <i>p</i>	[Función]
Negate <i>p</i> .	
<b>ly:pitch-notename</b> <i>pp</i>	[Función]
Extract the note name from pitch <i>pp</i> .	
<b>ly:pitch-octave</b> <i>pp</i>	[Función]
Extract the octave from pitch <i>pp</i> .	
<b>ly:pitch-quartertones</b> <i>pp</i>	[Función]
Calculate the number of quarter tones of <i>pp</i> from middle C.	
<b>ly:pitch-semitones</b> <i>pp</i>	[Función]
Calculate the number of semitones of <i>pp</i> from middle C.	
<b>ly:pitch-steps</b> <i>p</i>	[Función]
Number of steps counted from middle C of the pitch <i>p</i> .	
<b>ly:pitch-tones</b> <i>pp</i>	[Función]
Calculate the number of tones of <i>pp</i> from middle C as a rational number.	
<b>ly:pitch-transpose</b> <i>p delta</i>	[Función]
Transpose <i>p</i> by the amount <i>delta</i> , where <i>delta</i> is relative to middle C.	
<b>ly:pointer-group-interface::add-grob</b> <i>grob sym grob-element</i>	[Función]
Add <i>grob-element</i> to <i>grob</i> 's <i>sym</i> grob array.	
<b>polar-&gt;rectangular</b> <i>radius angle-in-degrees</i>	[Función]
Return polar coordinates ( <i>radius</i> , <i>angle-in-degrees</i> ) as rectangular coordinates ( <b>x-length</b> . <b>y-length</b> ).	
<b>ly:position-on-line?</b> <i>sg spos</i>	[Función]
Return whether <i>spos</i> is on a line of the staff associated with the grob <i>sg</i> (even on an extender line).	
<b>ly:prob?</b> <i>x</i>	[Función]
Is <i>x</i> a Prob object?	
<b>ly:prob-immutable-properties</b> <i>prob</i>	[Función]
Retrieve an alist of immutable properties.	

<b>ly:prob-mutable-properties</b> <i>prob</i>	[Función]
Retrieve an alist of mutable properties.	
<b>ly:prob-property</b> <i>prob sym val</i>	[Función]
Return the value for property <i>sym</i> of Prob object <i>prob</i> . If no value is found, return <i>val</i> or '()' if <i>val</i> is not specified.	
<b>ly:prob-property?</b> <i>obj sym</i>	[Función]
Is boolean prop <i>sym</i> of <i>sym</i> set?	
<b>ly:prob-set-property!</b> <i>obj sym value</i>	[Función]
Set property <i>sym</i> of <i>obj</i> to <i>value</i> .	
<b>ly:prob-type?</b> <i>obj type</i>	[Función]
Is <i>obj</i> the specified prob-type?	
<b>ly:programming-error</b> <i>str rest</i>	[Función]
A Scheme callable function to issue the internal warning <i>str</i> . The message is formatted with <i>format</i> and <i>rest</i> .	
<b>ly:progress</b> <i>str rest</i>	[Función]
A Scheme callable function to print progress <i>str</i> . The message is formatted with <i>format</i> and <i>rest</i> .	
<b>ly:property-lookup-stats</b> <i>sym</i>	[Función]
Return hash table with a property access corresponding to <i>sym</i> . Choices are <i>prob</i> , <i>grob</i> , and <i>context</i> .	
<b>ly:protects</b>	[Función]
Return hash of protected objects.	
<b>ly:pt</b> <i>num</i>	[Función]
<i>num</i> printer points.	
<b>ly:pure-call</b> <i>data grob start end rest</i>	[Función]
Convert property <i>data</i> (unpure-pure container or procedure) to value in a pure context defined by <i>grob</i> , <i>start</i> , <i>end</i> , and possibly <i>rest</i> arguments.	
<b>pure-chain-offset-callback</b> <i>grob start end prev-offset</i>	[Función]
Sometimes, a chained offset callback is unpure and there is no way to write a pure function that estimates its behavior. In this case, we use a pure equivalent that will simply pass the previous calculated offset value.	
<b>ly:randomize-rand-seed</b>	[Función]
Randomize C random generator.	
<b>ratio-&gt;fret</b> <i>ratio</i>	[Función]
Calculate a fret number given <i>ratio</i> for the harmonic.	
<b>ratio-&gt;pitch</b> <i>ratio</i>	[Función]
Calculate a pitch given <i>ratio</i> for the harmonic.	
<b>read-lily-expression</b> <i>chr port</i>	[Función]
Read a lilypond music expression enclosed within <i>#{</i> and <i>#}</i> from <i>port</i> and return the corresponding Scheme music expression. '\$' and '#' introduce immediate and normal Scheme forms.	

- recording-group-emulate** *music odef* [Función]  
 Interpret *music* according to *odef*, but store all events in a chronological list, similar to the `Recording_group_engraver` in LilyPond version 2.8 and earlier.
- ly:register-stencil-expression** *symbol* [Función]  
 Add *symbol* as head of a stencil expression.
- ly:register-translator** *creator name description* [Función]  
 Register a translator *creator* (usually a descriptive alist or a function/closure returning one when given a context argument) with the given symbol *name* and the given *description* alist.
- ly:relative-group-extent** *elements common axis* [Función]  
 Determine the extent of *elements* relative to *common* in the *axis* direction.
- remove-grace-property** *context-name grob sym* [Función]  
 Remove all *sym* for *grob* in *context-name*.
- remove-whitespace** *strg* [Función]  
 Remove characters satisfying `char-whitespace?` from string *strg*.
- ly:rename-file** *oldname newname* [Función]  
 Rename *oldname* to *newname*. In contrast to Guile's `rename-file`, this replaces the destination if it already exists. On Windows, fall back to copying the file contents if *newname* cannot be deleted.
- ly:reset-all-fonts** [Función]  
 Forget all about previously loaded fonts.
- retrieve-glyph-flag** *flag-style dir dir-modifier grob* [Función]  
 Load the correct flag glyph from the font.
- retrograde-music** *music* [Función]  
 Returns *music* in retrograde (reversed) order.
- revert-fontSize** *func-name mag* [Función]  
 Used by `\magnifyMusic` and `\magnifyStaff`. Calculate the previous `fontSize` value (before scaling) by factoring out the magnification factor *mag* (if *func-name* is `'magnifyMusic`), or by factoring out the context property `magnifyStaffValue` (if *func-name* is `'magnifyStaff`). Revert the `fontSize` in the appropriate context accordingly.  
 With `\magnifyMusic`, the scaling is reverted after the music block it operates on. `\magnifyStaff` does not operate on a music block, so the scaling from a previous call (if there is one) is reverted before the new scaling takes effect.
- revert-head-style** *heads* [Función]  
 Revert style for *heads*.
- revert-props** *func-name mag props* [Función]  
 Used by `\magnifyMusic` and `\magnifyStaff`. Revert each prop in *props* in the appropriate context. *func-name* is either `'magnifyMusic` or `'magnifyStaff`. The *props* list is formatted like:  

```
'((Stem thickness)
  (Slur line-thickness)
  ...)
```
- ly:round-filled-box** *xext yext blot* [Función]  
 Make a `Stencil` object that prints a black box of dimensions *xext*, *yext* and roundness *blot*.

- ly:round-polygon** *points blot extroversion filled-scm* [Función]  
 Make a **Stencil** object that prints a black polygon with corners at the points defined by *points* (list of coordinate pairs) and roundness *blot*. Optional *extroversion* shifts the outline outward, with the default of 0 keeping the middle of the line just on the polygon.
- rounded-box-stencil** *stencil thickness padding blot* [Función]  
 Add a rounded box around *stencil*, producing a new stencil.
- ly:run-translator** *mus output-def* [Función]  
 Process *mus* according to *output-def*. An interpretation context is set up, and *mus* is interpreted with it. The context is returned in its final state.  
 Optionally, this routine takes an object-key to uniquely identify the score block containing it.
- scale-beam-thickness** *mag* [Función]  
 Used by `\magnifyMusic`. Scaling **Beam.beam-thickness** exactly to the *mag* value will not work. This uses two reference values for **beam-thickness** to determine an acceptable value when scaling, then does the equivalent of a `\temporary \override` with the new value.
- scale-fontSize** *func-name mag* [Función]  
 Used by `\magnifyMusic` and `\magnifyStaff`. Look up the current **fontSize** in the appropriate context and scale it by the magnification factor *mag*. *func-name* is either `'magnifyMusic` or `'magnifyStaff`.
- scale-layout** *paper scale* [Función]  
 Return a clone of the paper, scaled by the given scale factor.
- scale-props** *func-name mag allowed-to-shrink? props* [Función]  
 Used by `\magnifyMusic` and `\magnifyStaff`. For each prop in *props*, find the current value of the requested prop, scale it by the magnification factor *mag*, and do the equivalent of a `\temporary \override` with the new value in the appropriate context. If *allowed-to-shrink?* is `#f`, don't let the new value be less than the current value. *func-name* is either `'magnifyMusic` or `'magnifyStaff`. The *props* list is formatted like:  

```
'((Stem thickness)
  (Slur line-thickness)
  ...)
```
- ly:score?** *x* [Función]  
 Is *x* a **Score** object?
- ly:score-add-output-def!** *score def* [Función]  
 Add an output definition *def* to *score*.
- ly:score-embedded-format** *score layout* [Función]  
 Run *score* through *layout* (an output definition) scaled to correct output-scale already, returning a list of layout-lines.
- ly:score-error?** *score* [Función]  
 Was there an error in the score?
- ly:score-header** *score* [Función]  
 Return score header.
- ly:score-music** *score* [Función]  
 Return score music.

<code>ly:score-output-defs</code> <i>score</i>	[Función]
All output definitions in a score.	
<code>ly:score-set-header!</code> <i>score module</i>	[Función]
Set the score header.	
<code>scorify-music</code> <i>music</i>	[Función]
Preprocess <i>music</i> .	
<code>seconds-&gt;moment</code> <i>s context</i>	[Función]
Return a moment equivalent to <i>s</i> seconds at the current tempo.	
<code>select-head-glyph</code> <i>style log</i>	[Función]
Select a note head glyph string based on note head style <i>style</i> and duration-log <i>log</i> .	
<code>self-alignment-interface::self-aligned-on-breakable</code> <i>grob</i>	[Función]
Return the X-offset that places <i>grob</i> according to its self-alignment-X over the reference point defined by the break-align-anchor-alignment of a break-aligned item such as a Clef.	
<code>ly:separation-item::print</code>	[Función]
Optional stencil for PaperColumn or NonMusicalPaperColumn. This function draws horizontal-skylines of each PaperColumn, showing the shapes used to determine the minimum distances between PaperColumns at the note-spacing step, before staves have been spaced (vertically) on the page.	
<code>sequential-music-to-chord-exceptions</code> <i>seq . rest</i>	[Función]
Transform sequential music SEQ of type <<c d e>>-\markup{ foobar } to (cons CDE-PITCHES FOOBAR-MARKUP), or to (cons DE-PITCHES FOOBAR-MARKUP) if OMIT-ROOT is given and non-false.	
<code>session-save</code>	[Función]
Save identifiers for use with session-replay.	
<code>set-accidental-style</code> <i>style . rest</i>	[Función]
Set accidental style to <i>style</i> . Optionally take a context argument, e.g. 'Staff or 'Voice. The context defaults to Staff, except for piano styles, which use GrandStaff as a context.	
<code>ly:set-default-scale</code> <i>scale</i>	[Función]
Set the global default scale. This determines the tuning of pitches with no accidentals or key signatures. The first pitch is C. Alterations are calculated relative to this scale. The number of pitches in this scale determines the number of scale steps that make up an octave. Usually the 7-note major scale.	
<code>set-global-staff-size</code> <i>sz</i>	[Función]
Set the default staff size, where SZ is thought to be in PT.	
<code>ly:set-grob-creation-callback</code> <i>cb</i>	[Función]
Specify a procedure that will be called every time a new grob is created. The callback will receive as arguments the grob that was created, the name of the C++ source file that caused the grob to be created, and the corresponding line number in the C++ source file. Call with #f as argument to unset the callback.	
<code>ly:set-grob-modification-callback</code> <i>cb</i>	[Función]
Specify a procedure that will be called every time LilyPond modifies a grob property. The callback will receive as arguments the grob that is being modified, the name of the C++	

file in which the modification was requested, the line number in the C++ file in which the modification was requested, the name of the function in which the modification was requested, the property to be changed, and the new value for the property. Call with **#f** as argument to unset the callback.

**ly:set-middle-C!** *context* [Función]  
Set the `middleCPosition` variable in *context* based on the variables `middleCClefPosition` and `middleCOffset`.

**set-mus-properties!** *m alist* [Función]  
Set all of *alist* as properties of *m*.

**ly:set-option** *var val* [Función]  
Set a program option.

**ly:set-origin!** *m origin* [Función]  
This sets the origin given in *origin* to *m*. *m* will typically be a music expression or a list of music. List structures are searched recursively, but recursion stops at the changed music expressions themselves. *origin* is generally of type `ly:input-location?`, defaulting to `(*location*)`. Other valid values for *origin* are a music expression which is then used as the source of location information, or **#f** or `'()` in which case no action is performed. The return value is *m* itself.

**ly:set-property-cache-callback** *cb* [Función]  
Specify a procedure that will be called whenever lilypond calculates a callback function and caches the result. The callback will receive as arguments the grob whose property it is, the name of the property, the name of the callback that calculated the property, and the new (cached) value of the property. Call with **#f** as argument to unset the callback.

**shift-one-duration-log** *music shift dot* [Función]  
Add *shift* to `duration-log` of `'duration` in *music* and optionally *dot* to any note encountered. The number of dots in the shifted music may not be less than zero.

**shift-right-at-line-begin** *g* [Función]  
Shift an item to the right, but only at the start of the line.

**skip->rest** *mus* [Función]  
Replace *mus* by `RestEvent` of the same duration if it is a `SkipEvent`. Useful for extracting parts from crowded scores.

**skip-of-length** *mus* [Función]  
Create a skip of exactly the same length as *mus*.

**ly:skyline?** *x* [Función]  
Is *x* a `Skyline` object?

**ly:skyline-empty?** *sky* [Función]  
Return whether *sky* is empty.

**ly:skyline-pair?** *x* [Función]  
Is *x* a `Skyline_pair` object?

**ly:smob-protects** [Función]  
Return LilyPond's internal smob protection list.

- ly:solve-spring-rod-problem** *springs rods length ragged* [Función]  
 Solve a spring and rod problem for *count* objects, that are connected by *count-1* *springs*, and an arbitrary number of *rods*. *count* is implicitly given by *springs* and *rods*. The *springs* argument has the format (*ideal*, *inverse\_hook*) and *rods* is of the form (*idx1*, *idx2*, *distance*).  
*length* is a number, *ragged* a boolean.  
 The function returns a list containing the force (positive for stretching, negative for compressing and **#f** for non-satisfied constraints) followed by *spring-count+1* positions of the objects.
- ly:source-file?** *x* [Función]  
 Is *x* a **Source\_file** object?
- ly:source-files** *parser-smob* [Función]  
 A list of input files that have been opened up to here, including the files that have been closed already. a **PARSER** may optionally be specified.
- ly:span-bar::before-line-breaking** *grob* [Función]  
 A dummy callback that kills the *Grob* *grob* if it contains no elements.
- ly:span-bar::calc-glyph-name** *grob* [Función]  
 Return the 'glyph-name' of the corresponding *BarLine* *grob*. The corresponding *SpanBar* glyph is computed within **span-bar::compound-bar-line**.
- span-bar::compound-bar-line** *grob bar-glyph extent* [Función]  
 Build the stencil of the span bar.
- ly:span-bar::print** *grob* [Función]  
 The print routine for span bars.
- ly:span-bar::width** *grob* [Función]  
 Compute the width of the *SpanBar* stencil.
- Span\_stem\_engraver** *ctx* [Función]  
 Connect cross-staff stems to the stems above in the system
- ly:spanner?** *g* [Función]  
 Is *g* a *spanner* object?
- ly:spanner-bound** *spanner dir* [Función]  
 Get one of the bounds of *spanner*. *dir* is -1 for left, and 1 for right.
- ly:spanner-broken-into** *spanner* [Función]  
 Return broken-into list for *spanner*.
- ly:spanner-set-bound!** *spanner dir item* [Función]  
 Set *grob item* as bound in direction *dir* for *spanner*.
- ly:spawn** *command rest* [Función]  
 Simple interface to *g-spawn-sync* *str*. The error is formatted with **format** and *rest*.
- split-list-by-separator** *lst pred* [Función]  
 Split *lst* at each element that satisfies *pred*, and return the parts (with the separators removed) as a list of lists. For example, executing '(**split-list-by-separator** '(a 0 b c 1 d) **number?**)' returns '((a) (b c) (d))'.



<b>ly:spring?</b> <i>x</i>	[Función]
Is <i>x</i> a <b>Spring</b> object?	
<b>ly:spring-set-inverse-compress-strength!</b> <i>spring strength</i>	[Función]
Set the inverse compress <i>strength</i> of <i>spring</i> .	
<b>ly:spring-set-inverse-stretch-strength!</b> <i>spring strength</i>	[Función]
Set the inverse stretch <i>strength</i> of <i>spring</i> .	
<b>stack-lines</b> <i>dir padding baseline stils</i>	[Función]
Stack vertically with a baseline skip.	
<b>stack-stencil-line</b> <i>space stencils</i>	[Función]
Adjoin a list of <i>stencils</i> along the X axis, leaving <i>space</i> between the end of each stencil and the beginning of the following stencil. Stencils with empty Y extent are not given <i>space</i> before them and don't avoid overlapping other stencils.	
<b>stack-stencils</b> <i>axis dir padding stils</i>	[Función]
Stack stencils <i>stils</i> in direction <i>axis</i> , <i>dir</i> , using <i>padding</i> .	
<b>stack-stencils-padding-list</b> <i>axis dir paddings stils</i>	[Función]
Stack stencils <i>stils</i> in direction <i>axis</i> , <i>dir</i> , using a list of <i>paddings</i> .	
<b>ly:staff-symbol-line-thickness</b> <i>grob</i>	[Función]
Returns the current staff-line thickness in the staff associated with <i>grob</i> , expressed as a multiple of the current staff-space height.	
<b>ly:staff-symbol-staff-radius</b> <i>grob</i>	[Función]
Returns the radius of the staff associated with <i>grob</i> .	
<b>ly:staff-symbol-staff-space</b> <i>grob</i>	[Función]
Returns the current staff-space height in the staff associated with <i>grob</i> , expressed as a multiple of the default height of a staff-space in the traditional five-line staff.	
<b>ly:stderr-redirect</b> <i>fd-or-file-name mode</i>	[Función]
Redirect stderr to <i>fd</i> if the first parameter is an integer, or to <i>file-name</i> , opened with <i>mode</i> .	
<b>ly:stencil?</b> <i>x</i>	[Función]
Is <i>x</i> a <b>Stencil</b> object?	
<b>ly:stencil-add</b> <i>args</i>	[Función]
Combine stencils. Takes any number of arguments.	
<b>ly:stencil-aligned-to</b> <i>stil axis dir</i>	[Función]
Align <i>stil</i> using its own extents. <i>dir</i> is a number. -1 and 1 are left and right, respectively. Other values are interpolated (so 0 means the center).	
<b>ly:stencil-combine-at-edge</b> <i>first axis direction second padding</i>	[Función]
Construct a stencil by putting <i>second</i> next to <i>first</i> . <i>axis</i> can be 0 (x-axis) or 1 (y-axis). <i>direction</i> can be -1 (left or down) or 1 (right or up). The stencils are juxtaposed with <i>padding</i> as extra space. <i>first</i> and <i>second</i> may also be '()' or #f.	
<b>ly:stencil-empty?</b> <i>stil axis</i>	[Función]
Return whether <i>stil</i> is empty. If an optional <i>axis</i> is supplied, the emptiness check is restricted to that axis.	
<b>ly:stencil-expr</b> <i>stil</i>	[Función]
Return the expression of <i>stil</i> .	

**ly:stencil-extent** *stil axis* [Función]  
 Return a pair of numbers signifying the extent of *stil* in *axis* direction (0 or 1 for x and y axis, respectively).

**ly:stencil-in-color** *stc r g b a* [Función]  
 Put *stc* in a different color. Accepts either three values for *r*, *g*, *b* and an optional value for *a*, or a single CSS-like string.

**ly:stencil-outline** *stil outline* [Función]  
 Return a stencil with the stencil expression (inking) of stencil *stil* but with outline and dimensions from stencil *outline*.

**ly:stencil-rotate** *stil angle x y* [Función]  
 Return a stencil *stil* rotated *angle* degrees around the relative offset (x, y). E.g., an offset of (-1, 1) will rotate the stencil around the left upper corner.

**ly:stencil-rotate-absolute** *stil angle x y* [Función]  
 Return a stencil *stil* rotated *angle* degrees around point (x, y), given in absolute coordinates.

**ly:stencil-scale** *stil x y* [Función]  
 Scale stencil *stil* using the horizontal and vertical scaling factors *x* and *y*. Negative values will flip or mirror *stil* without changing its origin; this may result in collisions unless it is repositioned.

**ly:stencil-stack** *first axis direction second padding mindist* [Función]  
 Construct a stencil by stacking *second* next to *first*. *axis* can be 0 (x-axis) or 1 (y-axis). *direction* can be -1 (left or down) or 1 (right or up). The stencils are juxtaposed with *padding* as extra space. *first* and *second* may also be '()' or #f. As opposed to **ly:stencil-combine-at-edge**, metrics are suited for successively accumulating lines of stencils. Also, *second* stencil is drawn last.

If *mindist* is specified, reference points are placed apart at least by this distance. If either of the stencils is spacing, *padding* and *mindist* do not apply.

**ly:stencil-translate** *stil offset* [Función]  
 Return a *stil*, but translated by *offset* (a pair of numbers).

**ly:stencil-translate-axis** *stil amount axis* [Función]  
 Return a copy of *stil* but translated by *amount* in *axis* direction.

**stencil-whiteout** *stil . lambda\*:G28* [Función]  
*style*, *thickness* and *line-thickness* are optional arguments. If set, *style* determines the shape of the white background. Given 'outline the white background is produced by **stencil-whiteout-outline**, given 'rounded-box it is produced by **stencil-whiteout-box** with rounded corners, given other arguments (e.g. 'box) or when unspecified it defaults to **stencil-whiteout-box** with square corners. If *thickness* is specified it determines how far, as a multiple of *line-thickness*, the white background extends past the extents of stencil *stil*. If *thickness* has not been specified, an appropriate default is chosen based on *style*.

**stencil-whiteout-box** *stil . lambda\*:G26* [Función]  
*thickness* is how far, as a multiple of line-thickness, the white outline extends past the extents of stencil *stil*.

**stencil-whiteout-outline** *stil . lambda\*:G24* [Función]  
 This function works by creating a series of white or *color* stencils radially offset from the original stencil with angles from 0 to 2\*pi, at an increment of *angle-inc*, and with radii

from `radial-inc` to *thickness*. *thickness* is how big the white outline is, as a multiple of line-thickness. *radial-increments* is how many copies of the white stencil we make on our way out to thickness. *angle-increments* is how many copies of the white stencil we make between 0 and  $2\pi$ .

**straight-flag** *flag-thickness* *flag-spacing* *upflag-angle* *upflag-length* *downflag-angle* *downflag-length* [Función]

Create a stencil for a straight flag. *flag-thickness* and *flag-spacing* are given in staff spaces, *upflag-angle* and *downflag-angle* are given in degrees, and *upflag-length* and *downflag-length* are given in staff spaces.

All lengths are scaled according to the font size of the note.

**ly:stream-event?** *obj* [Función]

Is *obj* a `Stream_event` object?

**ly:string-percent-encode** *str* [Función]

Encode all characters in string *str* with hexadecimal percent escape sequences, with the following exceptions: characters `-`, `.`, `/`, and `_`; and characters in ranges `0-9`, `A-Z`, and `a-z`.

**ly:string-substitute** *a b s* [Función]

Replace string *a* by string *b* in string *s*.

**style-note-heads** *heads style music* [Función]

Set *style* for all *heads* in *music*. Works both inside of and outside of chord construct.

**symbol-concatenate** *ame* [Función]

Like `string-concatenate`, but for symbols.

**ly:system-font-load** *name* [Función]

Load the OpenType system font *name.otf*. Fonts loaded with this command must contain three additional SFNT font tables called LILC, LILF, and LILY, needed for typesetting musical elements. Currently, only the Emmentaler and the Emmentaler-Brace fonts fulfill these requirements.

Note that only `ly:font-get-glyph` and derived code (like `\lookup`) can access glyphs from the system fonts; text strings are handled exclusively via the Pango interface.

**tag-group-get** *tag* [Función]

Return the tag group (as a list of symbols) that the given *tag* symbol belongs to, `#f` if none.

**tags-keep-predicate** *tags* [Función]

Returns a predicate that returns `#f` for any music that is to be removed by `\keepWithTag` on the given symbol or list of symbols *tags*.

**tags-remove-predicate** *tags* [Función]

Returns a predicate that returns `#f` for any music that is to be removed by `\removeWithTag` on the given symbol or list of symbols *tags*.

**teaching-accidental-rule** *context pitch barnum measurepos* [Función]

An accidental rule that typesets a cautionary accidental if it is included in the key signature *and* does not directly follow a note on the same staff line.

**ly:text-interface::interpret-markup** [Función]

Convert a text markup into a stencil. Takes three arguments, *layout*, *props*, and *markup*. *layout* is a `\layout` block; it may be obtained from a grob with `ly:grob-layout`. *props* is an alist chain, i.e. a list of alists. This is typically obtained with `(ly:grob-alist-chain grob (ly:output-def-lookup layout 'text-font-defaults))`. *markup* is the markup text to be processed.

<code>ly:time-signature::print</code> <i>grob</i>	[Función]
Print routine for time signatures.	
<code>ly:transform?</code> <i>x</i>	[Función]
Is <i>x</i> a Transform object?	
<code>ly:transform-&gt;list</code> <i>transform</i>	[Función]
Convert a transform matrix to a list of six values. Values are <i>xx</i> , <i>yx</i> , <i>xy</i> , <i>yy</i> , <i>x0</i> , <i>y0</i> .	
<code>ly:translate-cpp-warning-scheme</code> <i>str</i>	[Función]
Translates a string in C++ printf format and modifies it to use it for scheme formatting.	
<code>ly:translator?</code> <i>x</i>	[Función]
Is <i>x</i> a Translator object?	
<code>ly:translator-context</code> <i>trans</i>	[Función]
Return the context of the translator object <i>trans</i> .	
<code>ly:translator-description</code> <i>creator</i>	[Función]
Return an alist of properties of translator definition <i>creator</i> .	
<code>ly:translator-group?</code> <i>x</i>	[Función]
Is <i>x</i> a Translator_group object?	
<code>ly:translator-name</code> <i>creator</i>	[Función]
Return the type name of the translator definition <i>creator</i> . The name is a symbol.	
<code>ly:transpose-key-alist</code> <i>l</i> <i>pit</i>	[Función]
Make a new key alist of <i>l</i> transposed by pitch <i>pit</i> .	
<code>ly:ttf-&gt;pfa</code> <i>ttf-file-name</i> <i>idx</i>	[Función]
Convert the contents of a TrueType font file to PostScript Type 42 font, returning it as a string. The optional <i>idx</i> argument is useful for TrueType collections (TTC) only; it specifies the font index within the TTC. The default value of <i>idx</i> is 0.	
<code>ly:ttf-ps-name</code> <i>ttf-file-name</i> <i>idx</i>	[Función]
Extract the PostScript name from a TrueType font. The optional <i>idx</i> argument is useful for TrueType collections (TTC) only; it specifies the font index within the TTC. The default value of <i>idx</i> is 0.	
<code>ly:type1-&gt;pfa</code> <i>type1-file-name</i>	[Función]
Convert the contents of a Type 1 font in PFB format to PFA format. If the file is already in PFA format, pass through it.	
<code>unfold-repeats</code> <i>types</i> <i>music</i>	[Función]
Replace repeats of the types given by <i>types</i> with unfolded repeats. If <i>types</i> is an empty list, <i>repeated-music</i> is taken, unfolding all.	
<code>unfold-repeats-fully</code> <i>music</i>	[Función]
Unfolds repeats and expands the resulting <i>unfolded-repeated-music</i> .	
<code>uniq-list</code> <i>lst</i>	[Función]
Uniq <i>lst</i> , assuming that it is sorted. Uses <code>equal?</code> for comparisons.	
<code>ly:unit</code>	[Función]
Return the unit used for lengths as a string.	

- unity-if-multimeasure** *context dur* [Función]  
 Given a context and a duration, return 1 if the duration is longer than the `measureLength` in that context, and `#f` otherwise. This supports historic use of `Completion_heads_engraver` to split `c1*3` into three whole notes.
- ly:unpure-call** *data grob rest* [Función]  
 Convert property *data* (unpure-pure container or procedure) to value in an unpure context defined by *grob* and possibly *rest* arguments.
- ly:unpure-pure-container?** *x* [Función]  
 Is *x* a `Unpure_pure_container` object?
- ly:unpure-pure-container-pure-part** *pc* [Función]  
 Return the pure part of *pc*.
- ly:unpure-pure-container-unpure-part** *pc* [Función]  
 Return the unpure part of *pc*.
- ly:usage** [Función]  
 Print usage message.
- ly:verbose-output?** [Función]  
 Was verbose output requested, i.e. loglevel at least `DEBUG`?
- ly:version** [Función]  
 Return the current LilyPond version as a list, e.g., `(1 3 127 uu1)`.
- ly:version?** *op ver* [Función]  
 Use operator *op* to compare the currently executed LilyPond version with a given version *ver*, which is passed as a list of numbers.
- voicify-music** *m . lambda\*:G79* [Función]  
 Recursively split chords that are separated with `\\`. Optional *id* can be a list of context ids to use. If numeric, they also indicate a voice type override. If *id* is just a single number, that's where numbering starts.
- volta-bracket::calc-hook-visibility** *bar-glyph* [Función]  
 Determine the visibility of the volta bracket end hook, returning `#t` if *no* hook should be drawn.
- ly:volta-bracket::calc-shorten-pair** *grob* [Función]  
 Calculate the `shorten-pair` values for an ideal placement of the volta brackets relative to the bar lines.
- volta-spec-music** *number-list music* [Función]  
 Add `\volta number-list` to *music*.
- ly:warning** *str rest* [Función]  
 A Scheme callable function to issue the warning *str*. The message is formatted with `format` and *rest*.
- ly:warning-located** *location str rest* [Función]  
 A Scheme callable function to issue the warning *str* at the specified location in an input file. The message is formatted with `format` and *rest*.
- ly:wide-char->utf-8** *wc* [Función]  
 Encode the Unicode codepoint *wc*, an integer, as UTF-8.
- write-me** *message x* [Función]  
 Return *x*. Display *message* and write *x*. Handy for debugging, possibly turned off.

## Apéndice B Hoja de referencia rápida

Sintaxis	Descripción	Ejemplo
<code>1 2 8 16</code>	duraciones	
<code>c4. c4..</code>	puntillos	
<code>c d e f g a b</code>	escala	
<code>fis bes</code>	alteración	
<code>\clef treble \clef bass</code>	claves	
<code>\time 3/4 \time 4/4</code>	compás	
<code>r4 r8</code>	silencio	
<code>d ~ d</code>	ligadura de unión	

`\key es \major`

armadura



`note'`

subir una octava



`note,`

bajar una octava



`c( d e)`

ligadura de expresión



`c\ ( c( d) e\)`

ligadura de fraseo



`a8[ b]`

barra



`<< \new Staff ... >>`

más pentagramas



`c-> c-.`

articulaciones



`c2\mf c\s fz`

matices dinámicos



`a\< a a\!`

crescendo



`a\> a a\!`

decrescendo



`< >`

acorde



`\partial 8`

anacrusa o compás  
incompleto



`\tuplet 3/2 {f g a}`

tresillos



`\grace`

mordentes



`\lyricmode { twinkle }`

escribir la letra

twinkle

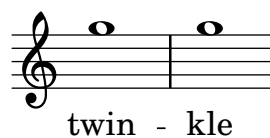
`\new Lyrics`

imprimir la letra

twinkle

`twin -- kle`

separador de sílabas





`\chordmode { c:dim f:maj7 }`      acordes



`\new ChordNames`      imprimir los nombres de los acordes

$C^\circ \quad F^\Delta$

`<<\{e f\} \\\{c d\}>>`      polifonía



`s4 s8 s16`      silencios de separación

## Apéndice C GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.  
<https://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

### 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both

covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

#### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its

Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements.”

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

#### 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <https://www.gnu.org/licenses/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

#### 11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

## ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.3  
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover  
Texts.  A copy of the license is included in the section entitled ``GNU  
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being list their titles, with  
the Front-Cover Texts being list, and with the Back-Cover Texts  
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.



## Apéndice D Índice de instrucciones de LilyPond

Este índice relaciona todas las instrucciones y palabras clave de LilyPond, con enlaces a aquellas secciones del manual que describen o se ocupan de su uso.

<b>!</b>		<b>&lt;</b>	
!..... 7		<...>..... 176	
\! ..... 131		<>..... 176, 354	
		<..... 176	
<b>"</b>		\< ..... 131	
" " ..... 117			
		<b>=</b>	
<b>%</b>		\=..... 140, 855	
%..... 507, 511		=..... 10	
%{ ... %} ..... 507, 511			
<b>,</b>		<b>&gt;</b>	
' ..... 1		>..... 176	
		\> ..... 131	
<b>(</b>			
(..... 140		<b>?</b>	
\( ..... 143		?..... 7	
<b>)</b>			
)..... 140		<b>[</b>	
\) ..... 143		[..... 100	
		\[ ..... 466	
<b>,</b>			
,..... 1		<b>]</b>	
<b>—</b>		]..... 100	
—..... 128		\] ..... 466	
-! ..... 800			
-+ ..... 801		<b>^</b>	
-- ..... 800		^..... 446	
-..... 800			
-> ..... 800		<b>-</b>	
-^ ..... 800		-..... 294	
-_ ..... 800			
<b>.</b>			
..... 50		<b> </b>	
		..... 117	
<b>/</b>			
/..... 446		<b>~</b>	
/+ ..... 446		~..... 58	
<b>:</b>			
:..... 174			

## A

`\abs-fontsize` ..... 266, 740  
`\absolute` ..... 844  
`\accent` ..... 128, 800  
`\accentus` ..... 478, 802  
`\accepts` ..... 638, 639, 640  
`\acciaccatura` ..... 120, 844  
`\accidental` ..... 774  
`\accidentalStyle` ..... 30, 844  
`AccidentalSuggestion` ..... 130  
`add-grace-property` ..... 123  
`add-stem-support` ..... 241  
`add-toc-item!` ..... 537  
`\addChordShape` ..... 404, 844  
`\addInstrumentDefinition` ..... 844  
`additionalPitchPrefix` ..... 451  
`\addlyrics` ..... 288, 290, 291  
`\addQuote` ..... 224, 844  
`\aeolian` ..... 23  
`\afterGrace` ..... 121, 844  
`afterGraceFraction` ..... 807  
`\aikenHeads` ..... 44  
`\aikenHeadsMinor` ..... 45  
`\aikenThinHeads` ..... 44  
`\aikenThinHeadsMinor` ..... 45  
`\alias` ..... 638  
`alignAboveContext` ..... 214, 642  
`alignBelowContext` ..... 214, 310, 642  
`\allowPageTurn` ..... 589, 844  
`\allowVoltaHook` ..... 844  
`\alterBroken` ..... 688, 844  
`\alternative` ..... 159  
`\ambitusAfter` ..... 844  
`AmbitusLine` ..... 39  
`annotate-spacing` ..... 618  
`\appendToTag` ..... 545, 845  
`\applyContext` ..... 627, 845  
`\applyMusic` ..... 845  
`\applyOutput` ..... 845  
`\appoggiatura` ..... 120, 845  
`\arabicStringNumbers` ..... 370  
`\arpeggio` ..... 152  
`arpeggio-direction` ..... 152  
`\arpeggioArrowDown` ..... 152  
`\arpeggioArrowUp` ..... 152  
`\arpeggioBracket` ..... 152  
`\arpeggioNormal` ..... 152  
`\arpeggioParenthesis` ..... 152  
`\arpeggioParenthesisDashed` ..... 152  
`\arrow-head` ..... 275, 765  
`\articulate` ..... 564  
`articulation-event` ..... 226  
`\ascendens` ..... 480, 486  
`\assertBeamQuant` ..... 845  
`\assertBeamSlope` ..... 845  
`associatedVoice` ..... 288, 290, 323  
`\auctum` ..... 480, 486  
`aug` ..... 443  
`\augmentum` ..... 486  
`auto-first-page-number` ..... 577  
`\auto-footnote` ..... 789  
`autoBeaming` ..... 91, 623  
`\autoBeamOff` ..... 88, 359  
`\autoBeamOn` ..... 88

`\autoBreaksOff` ..... 582  
`\autoBreaksOn` ..... 582  
`\autoChange` ..... 356, 845  
`\autoLineBreaksOff` ..... 582  
`\autoLineBreaksOn` ..... 582  
`automaticBars` ..... 675  
`\autoPageBreaksOff` ..... 586  
`\autoPageBreaksOn` ..... 586

## B

`\backslashed-digit` ..... 789  
`Balloon_engraver` ..... 248  
`\balloonGrobText` ..... 248, 845  
`\balloonLengthOff` ..... 248  
`\balloonLengthOn` ..... 248  
`\balloonText` ..... 248, 845  
`banjo-c-tuning` ..... 419  
`banjo-modal-tuning` ..... 419  
`banjo-open-d-tuning` ..... 419  
`banjo-open-dm-tuning` ..... 419  
`\bar` ..... 104, 111, 845  
`barCheckSynchronize` ..... 117  
`BarNumber` ..... 112  
`\barNumberCheck` ..... 118, 845  
`barNumberVisibility` ..... 112  
`bartype` ..... 111  
`base-shortest-duration` ..... 607  
`baseMoment` ..... 91, 96  
`\bassFigureExtendersOff` ..... 458  
`\bassFigureExtendersOn` ..... 458  
`\bassFigureStaffAlignmentDown` ..... 461  
`\bassFigureStaffAlignmentNeutral` ..... 461  
`\bassFigureStaffAlignmentUp` ..... 461  
`\beam` ..... 765  
`\beamExceptions` ..... 91, 845  
`beatStructure` ..... 91, 96  
`\bendAfter` ..... 146, 845  
`\bendHold` ..... 845  
`\bendStartLevel` ..... 845  
`binding-offset` ..... 575  
`\blackTriangleMarkup` ..... 451  
`blank-after-score-page-penalty` ..... 577  
`blank-last-page-penalty` ..... 577  
`blank-page-penalty` ..... 576  
`\bold` ..... 266, 740  
`\book` ..... 507, 510  
`\bookOutputName` ..... 509, 845  
`\bookOutputSuffix` ..... 509, 845  
`\bookpart` ..... 508, 510, 586  
`bookTitleMarkup` ..... 520  
`bottom-margin` ..... 570  
`\box` ..... 274, 740  
`\bracket` ..... 138, 766  
`bracket` ..... 362  
`\bracket` ..... 274  
`\break` ..... 582  
`break-align-symbols` ..... 681  
`break-visibility` ..... 671  
`breakable` ..... 89  
`breakbefore` ..... 518  
`\breathe` ..... 144, 846  
`BreathingSign` ..... 145  
`\breve` ..... 49, 62

## C

<code>\cadenzaOff</code> .....	79
<code>\cadenzaOn</code> .....	79
<code>\caesura</code> .....	478
<code>\caps</code> .....	740
<code>\cavum</code> .....	480, 486
<code>\center-align</code> .....	269, 750
<code>\center-column</code> .....	271, 750
<code>\change</code> .....	354
<code>\char</code> .....	790
<code>check-consistency</code> .....	574
<code>choral</code> .....	34
<code>choral-cautionary</code> .....	34
<code>chordChanges</code> .....	408, 449
<code>\chordmode</code> .....	6, 14, 402, 660
<code>chordNameExceptions</code> .....	452, 453
<code>chordNameLowercaseMinor</code> .....	451
<code>ChordNames</code> .....	402
<code>chordNameSeparator</code> .....	452, 455
<code>chordNoteNamer</code> .....	451
<code>chordPrefixSpacer</code> .....	453
<code>\chordRepeats</code> .....	373, 846
<code>chordRootNamer</code> .....	451
<code>\chords</code> .....	449, 660
<code>\circle</code> .....	274, 766
<code>\circulus</code> .....	478, 802
<code>\clef</code> .....	18, 846
<code>clip-regions</code> .....	548
<code>\cm</code> .....	663
<code>\coda</code> .....	128, 801
<code>color</code> .....	243
<code>\column</code> .....	271, 750
<code>\column-lines</code> .....	796
<code>\combine</code> .....	275, 750
<code>common-shortest-duration</code> .....	607
<code>Completion_heads_engraver</code> .....	84
<code>Completion_rest_engraver</code> .....	84
<code>\compound-meter</code> .....	774
<code>\compoundMeter</code> .....	83, 846
<code>\compressEmptyMeasures</code> .....	233
<code>\compressMMRests</code> .....	65, 67, 233, 846
<code>\concat</code> .....	751
<code>\consists</code> .....	631, 638
<code>\context</code> .....	624, 633
<code>context-spec-music</code> .....	189
<code>controlpitch</code> .....	10
<code>countPercentRepeats</code> .....	171
<code>\cr</code> .....	131
<code>\cresc</code> .....	132
<code>crescendo-event</code> .....	226
<code>crescendoSpanner</code> .....	136
<code>crescendoText</code> .....	136
<code>\crescHairpin</code> .....	132
<code>\crescTextCresc</code> .....	132
<code>cross</code> .....	41
<code>\crossStaff</code> .....	359, 846
<code>\cueClef</code> .....	227, 846
<code>\cueClefUnset</code> .....	227, 846
<code>\cueDuring</code> .....	227, 846
<code>\cueDuringWithClef</code> .....	227, 846
<code>currentBarNumber</code> .....	112, 126
<code>\customTabClef</code> .....	775

## D

<code>\dashBang</code> .....	129
<code>\dashDash</code> .....	129
<code>\dashDot</code> .....	129
<code>\dashHat</code> .....	129
<code>\dashLarger</code> .....	129
<code>\dashPlus</code> .....	129
<code>\dashUnderscore</code> .....	129
<code>\deadNote</code> .....	41, 846
<code>\deadNotesOff</code> .....	41
<code>\deadNotesOn</code> .....	41
<code>\decr</code> .....	131
<code>\decresc</code> .....	132
<code>decrescendoSpanner</code> .....	136
<code>decrescendoText</code> .....	136
<code>default-staff-staff-spacing</code> .....	590
<code>default</code> .....	30, 31
<code>\default</code> .....	118, 526
<code>defaultBarType</code> .....	111
<code>\defaultchild</code> .....	641
<code>\defaultTimeSignature</code> .....	70
<code>\defineBarLine</code> .....	108, 846
<code>\deminutum</code> .....	480, 486
<code>\denies</code> .....	638, 639, 640
<code>\descendens</code> .....	480, 486
<code>dim</code> .....	443
<code>\dim</code> .....	132
<code>\dimHairpin</code> .....	132
<code>\dimTextDecr</code> .....	132
<code>\dimTextDecresc</code> .....	132
<code>\dimTextDim</code> .....	132
<code>\dir-column</code> .....	751
<code>\discant</code> .....	784
<code>\displayLilyMusic</code> .....	565, 846
<code>\displayMusic</code> .....	846
<code>\displayScheme</code> .....	846
<code>\divisioMaior</code> .....	478
<code>\divisioMaxima</code> .....	478
<code>\divisioMinima</code> .....	478
<code>dodecaphonic</code> .....	35
<code>dodecaphonic-first</code> .....	36
<code>dodecaphonic-no-repeat</code> .....	36
<code>\dorian</code> .....	23
<code>\dotsDown</code> .....	50
<code>\dotsNeutral</code> .....	50
<code>\dotsUp</code> .....	50
<code>\doubleflat</code> .....	775
<code>\doublesharp</code> .....	775
<code>\downbow</code> .....	128, 366, 801
<code>\downmordent</code> .....	128, 800
<code>\downprall</code> .....	128, 800
<code>\draw-circle</code> .....	275, 766
<code>\draw-dashed-line</code> .....	766
<code>\draw-dotted-line</code> .....	767
<code>\draw-hline</code> .....	767
<code>\draw-line</code> .....	275, 768
<code>\draw-squiggle-line</code> .....	768
<code>\dropNote</code> .....	447, 846
<code>\drummode</code> .....	201, 420, 660
<code>drumPitchNames</code> .....	425
<code>drumPitchTable</code> .....	426
<code>\drums</code> .....	420, 660
<code>DrumStaff</code> .....	201
<code>drumStyleTable</code> .....	425

\dwn ..... 499  
\dynamic ..... 138, 741  
dynamic-event ..... 226  
\dynamicDown ..... 133  
DynamicLineSpanner ..... 133, 135  
\dynamicNeutral ..... 133  
\dynamicUp ..... 133

## E

\easyHeadsOff ..... 42  
\easyHeadsOn ..... 42  
\ellipse ..... 768  
\endcr ..... 131  
\enddecr ..... 131  
\endSpanners ..... 669, 847  
\epistemFinis ..... 478  
\epistemInitium ..... 478  
\epsfile ..... 275, 769  
\espressivo ..... 128, 132, 800  
\etc ..... 693  
\eventChords ..... 847  
\expandEmptyMeasures ..... 233  
explicitClefVisibility ..... 673  
explicitKeySignatureVisibility ..... 673  
extra-offset ..... 590  
\eyeglasses ..... 790  
Ez\_numbers\_engraver ..... 42

## F

\f ..... 131  
\featherDurations ..... 103, 847  
\fermata ..... 128, 775, 801  
\ff ..... 131  
\fff ..... 131  
\ffff ..... 131  
\ffffff ..... 131  
figuredBassAlterationDirection ..... 459  
figuredBassPlusDirection ..... 459  
\figuremode ..... 457, 660  
\figures ..... 457, 660  
\fill-line ..... 272, 752  
\fill-with-pattern ..... 538, 752  
\filled-box ..... 275, 769  
\finalis ..... 478  
\finger ..... 240, 741, 847  
fingeringOrientations ..... 240  
first-page-number ..... 577  
\first-visible ..... 790  
\fixed ..... 2, 847  
\flageolet ..... 128, 429, 801  
\flat ..... 775  
\flexa ..... 486  
followVoice ..... 358  
font-encoding ..... 280  
font-interface ..... 239, 280  
font-size ..... 235, 239  
\fontCaps ..... 741  
\fontsize ..... 266, 741  
fontSize ..... 235  
\footnote ..... 526, 790, 847  
footnote-separator-markup ..... 578  
Forbid\_line\_break\_engraver ..... 55

forget ..... 37  
four-string-banjo ..... 419  
\fp ..... 131  
\fraction ..... 790  
\freeBass ..... 784  
\frenchChords ..... 450  
\fret-diagram ..... 391, 780  
fret-diagram-interface ..... 397  
\fret-diagram-terse ..... 393, 781  
\fret-diagram-verbose ..... 395, 781  
FretBoards ..... 400  
\fromproperty ..... 790  
\funkHeads ..... 44  
\funkHeadsMinor ..... 45

## G

\general-align ..... 270, 753  
\germanChords ..... 450  
\glissando ..... 147  
\glissandoMap ..... 148  
\grace ..... 120, 847  
GregorianTranscriptionStaff ..... 201  
Grid\_line\_span\_engraver ..... 249  
Grid\_point\_engraver ..... 249  
gridInterval ..... 249  
\grobdescriptions ..... 847  
grow-direction ..... 103

## H

\halfopen ..... 128, 425, 801  
\halign ..... 269, 754  
\harmonic ..... 41, 366, 376  
\harmonicByFret ..... 376, 847  
\harmonicByRatio ..... 376, 847  
\harmonicNote ..... 847  
\harmonicsOff ..... 366  
\harmonicsOn ..... 366, 847  
\harp-pedal ..... 782  
\haydnturn ..... 128, 800  
\hbracket ..... 274, 769  
\hcenter-in ..... 755  
\header ..... 510  
\henzelongfermata ..... 128, 801  
\henzeshortfermata ..... 128, 801  
\hide ..... 670, 847  
\hideKeySignature ..... 431  
\hideNotes ..... 242  
\hideSplitTiedTabNotes ..... 375  
\hideStaffSwitch ..... 358  
horizontal-shift ..... 575  
Horizontal\_bracket\_engraver ..... 251  
HorizontalBracketText ..... 253  
\hspace ..... 270, 755  
\huge ..... 235, 268, 741

# I

<code>\ictus</code> .....	478, 802
<code>\iij</code> .....	481
<code>\IIJ</code> .....	481
<code>\ij</code> .....	481
<code>\IJ</code> .....	481
<code>\improvisationOff</code> .....	47, 86
<code>\improvisationOn</code> .....	47, 86
<code>\in</code> .....	663
<code>\incipit</code> .....	490, 848
<code>\inclinatum</code> .....	480, 486
<code>\include</code> .....	511, 538
<code>indent</code> .....	222, 575, 611
<code>\inherit-acceptability</code> .....	639, 848
<code>inner-margin</code> .....	574
<code>\inStaffSegno</code> .....	161, 848
<code>\instrumentSwitch</code> .....	848
<code>\inversion</code> .....	14, 848
<code>\invertChords</code> .....	447, 848
<code>\ionian</code> .....	23
<code>\italianChords</code> .....	450
<code>\italic</code> .....	266, 742

# J

<code>\justified-lines</code> .....	279, 796
<code>\justify</code> .....	273, 757
<code>\justify-field</code> .....	756
<code>\justify-line</code> .....	756
<code>\justify-string</code> .....	757

# K

<code>keepAliveInterfaces</code> .....	216
<code>\keepWithTag</code> .....	541, 848
<code>\key</code> .....	23, 45, 848
<code>\kievanOff</code> .....	488
<code>\kievanOn</code> .....	488
<code>KievanStaff</code> .....	486
<code>KievanVoice</code> .....	486
<code>\killCues</code> .....	231, 848

# L

<code>\label</code> .....	535, 848
<code>\laissezVibrer</code> .....	59
<code>\language</code> .....	848
<code>\languageRestore</code> .....	848
<code>\languageSaveAndChange</code> .....	848
<code>\large</code> .....	235, 268, 742
<code>\larger</code> .....	266, 268, 742
<code>last-bottom-spacing</code> .....	572
<code>\layout</code> .....	510, 579, 622, 633
<code>layout-set-staff-size</code> .....	581
<code>\left-align</code> .....	269, 758
<code>\left-brace</code> .....	791
<code>\left-column</code> .....	758
<code>left-margin</code> .....	573
<code>\lheel</code> .....	128, 801
<code>\line</code> .....	759
<code>line-width</code> .....	272, 573, 611
<code>\linea</code> .....	480, 486
<code>\lineprall</code> .....	128, 800
<code>\locrian</code> .....	23

<code>\longa</code> .....	49, 62
<code>\longfermata</code> .....	128, 801
<code>\lookup</code> .....	791
<code>\lower</code> .....	270, 759
<code>\ltoe</code> .....	128, 801
<code>ly: minimal-breaking</code> .....	587
<code>ly: one-line-auto-height-breaking</code> .....	588
<code>ly: one-line-breaking</code> .....	588
<code>ly: one-page-breaking</code> .....	588
<code>ly: optimal-breaking</code> .....	587
<code>ly: page-turn-breaking</code> .....	588
<code>\lydian</code> .....	23
<code>\lyricmode</code> .....	287, 288, 661
<code>\lyrics</code> .....	661
<code>\lyricsto</code> .....	288, 290, 291

# M

<code>m</code> .....	443
<code>magnification-&gt;font-size</code> .....	235, 581
<code>\magnify</code> .....	266, 742
<code>\magnifyMusic</code> .....	235, 848
<code>\magnifyStaff</code> .....	581, 848
<code>magstep</code> .....	235, 581, 663
<code>maj</code> .....	443
<code>\major</code> .....	23
<code>majorSevenSymbol</code> .....	451, 454
<code>make-dynamic-script</code> .....	138
<code>make-pango-font-tree</code> .....	284
<code>\makeClusters</code> .....	181, 849
<code>\makeDefaultStringTuning</code> .....	849
<code>\map-markup-commands</code> .....	796
<code>\marcato</code> .....	128, 800
<code>\mark</code> .....	118, 260, 849
<code>Mark_engraver</code> .....	262
<code>\markalphabet</code> .....	791
<code>\markLengthOff</code> .....	76, 261
<code>\markLengthOn</code> .....	76, 261
<code>\markletter</code> .....	792
<code>\markup</code> .....	254, 260, 262, 263, 264, 661
<code>markup-markup-spacing</code> .....	572
<code>markup-system-spacing</code> .....	572
<code>\markuplist</code> .....	263, 279
<code>\markupMap</code> .....	849
<code>max-systems-per-page</code> .....	575
<code>\maxima</code> .....	49, 62
<code>Measure_grouping_engraver</code> .....	97
<code>measureLength</code> .....	91, 126
<code>measurePosition</code> .....	78, 126
<code>\medium</code> .....	743
<code>\melisma</code> .....	295
<code>\melismaEnd</code> .....	295
<code>MensuralStaff</code> .....	201, 468
<code>MensuralVoice</code> .....	468
<code>\mergeDifferentlyDottedOff</code> .....	186
<code>\mergeDifferentlyDottedOn</code> .....	186
<code>\mergeDifferentlyHeadedOff</code> .....	186
<code>\mergeDifferentlyHeadedOn</code> .....	186
<code>\mf</code> .....	131
<code>middleCPosition</code> .....	27
<code>\midi</code> .....	510, 622
<code>midBalance</code> .....	563
<code>midChannelMapping</code> .....	561
<code>midChorusLevel</code> .....	563

midiDrumPitches.....	426
midiExpression.....	563
midiPanPosition.....	563
midiReverbLevel.....	563
min-systems-per-page.....	575
minimum-Y-extent.....	590
minimumFret.....	373, 413
minimumPageTurnLength.....	589
minimumRepeatLengthForPageTurn.....	589
\minor.....	23
minorChordModifier.....	452
mixed.....	362
\mixolydian.....	23
\mm.....	663
\modalInversion.....	17, 849
\modalTranspose.....	16, 849
mode.....	807
modern.....	32
modern-cautionary.....	32
modern-voice.....	33
modern-voice-cautionary.....	33
\mordent.....	128, 800
\mp.....	131
\multi-measure-rest-by-number.....	775
MultiMeasureRestScript.....	66
MultiMeasureRestText.....	66
\musicglyph.....	119, 776
\musicMap.....	849
musicQuotes.....	807

## N

\n.....	131
\name.....	638
\natural.....	776
neo-modern.....	34
neo-modern-cautionary.....	35
neo-modern-voice.....	35
neo-modern-voice-cautionary.....	35
\new.....	624
\newSpacingSection.....	608
no-reset.....	36
\noBeam.....	101
\noBreak.....	582
nonstaff-nonstaff-spacing.....	590
nonstaff-relatedstaff-spacing.....	590
nonstaff-unrelatedstaff-spacing.....	590
\noPageBreak.....	586, 849
\noPageTurn.....	589, 849
\normal-size-sub.....	743
\normal-size-super.....	267, 743
\normal-text.....	743
\normalsize.....	235, 268, 744
\note.....	777
\note-by-number.....	776
note-event.....	226
Note_heads_engraver.....	84
Note_name_engraver.....	247
\notemode.....	661
noteNameFunction.....	247
NoteNames.....	247
noteNameSeparator.....	247
\null.....	270, 792
NullVoice.....	317

\number.....	744
\numericTimeSignature.....	70

## O

\octaveCheck.....	10, 849
\offset.....	653, 849
\omit.....	670, 849
\on-the-fly.....	523, 792
\once.....	648, 650, 655, 689, 849
\oneVoice.....	181
\open.....	128, 366, 429, 801
\oriscus.....	480, 486
\ottava.....	25, 850
ottavation-numbers.....	25
ottavation-ordinals.....	25
ottavation-simple-ordinals.....	25
ottavationMarkups.....	25
outer-margin.....	575
output-count.....	807
output-def.....	807
output-suffix.....	807
outside-staff-horizontal-padding.....	605
outside-staff-padding.....	605
outside-staff-priority.....	605
\oval.....	770
\overlay.....	759
\override.....	648, 652, 792
\override-lines.....	796
\overrideProperty.....	652, 850
\overrideTimeSignatureSettings.....	71, 850
\overtie.....	744

## P

\p.....	131
\pad-around.....	274, 759
\pad-markup.....	274, 760
\pad-to-box.....	274, 760
\pad-x.....	274, 760
page-breaking.....	576
page-breaking-system-system-spacing.....	576
page-count.....	576
\page-link.....	792
page-number-type.....	577
\page-ref.....	535, 793
page-spacing-weight.....	578
\pageBreak.....	586, 587, 850
\pageTurn.....	589, 850
\palmMute.....	850
\palmMuteOn.....	850
\paper.....	510, 568
paper-height.....	570
paper-width.....	573
\parallelMusic.....	197, 850
\parenthesize.....	245, 770, 850
\partCombine.....	192, 317, 850
\partCombineApart.....	194
\partCombineAutomatic.....	194
\partCombineChords.....	194
\partCombineDown.....	851
\partCombineForce.....	851
partCombineListener.....	807
\partCombineSoloI.....	194

<code>\partCombineSoloII</code> .....	194
<code>\partCombineUnisono</code> .....	194
<code>\partCombineUp</code> .....	851
<code>\partial</code> .....	78, 159, 160, 851
<code>\path</code> .....	771
<code>\pattern</code> .....	793
<code>pedalSustainStyle</code> .....	362
<code>percent</code> .....	170
<code>\pes</code> .....	486
<code>\phrasingSlurDashed</code> .....	143
<code>\phrasingSlurDashPattern</code> .....	144, 851
<code>\phrasingSlurDotted</code> .....	143
<code>\phrasingSlurDown</code> .....	143
<code>\phrasingSlurHalfDashed</code> .....	143
<code>\phrasingSlurHalfSolid</code> .....	143
<code>\phrasingSlurNeutral</code> .....	143
<code>\phrasingSlurSolid</code> .....	143
<code>\phrasingSlurUp</code> .....	143
<code>\phrygian</code> .....	23
<code>piano</code> .....	33
<code>piano-cautionary</code> .....	34
<code>PianoStaff</code> .....	353, 356
<code>Pitch_squash_engraver</code> .....	86
<code>\pitchedTrill</code> .....	156, 851
<code>pitchnames</code> .....	807
<code>\pointAndClickOff</code> .....	851
<code>\pointAndClickOn</code> .....	851
<code>\pointAndClickTypes</code> .....	851
<code>\polygon</code> .....	772
<code>\portato</code> .....	128, 800
<code>\postscript</code> .....	275, 772
<code>\pp</code> .....	131
<code>\ppp</code> .....	131
<code>\pppp</code> .....	131
<code>\prall</code> .....	128, 800
<code>\pralldown</code> .....	128, 800
<code>\prallmordent</code> .....	128, 800
<code>\prallprall</code> .....	128, 800
<code>\prallup</code> .....	128, 800
<code>\preBend</code> .....	851
<code>\preBendHold</code> .....	851
<code>predefinedDiagramTable</code> .....	409
<code>\predefinedFretboardsOff</code> .....	411
<code>\predefinedFretboardsOn</code> .....	411
<code>print-all-headers</code> .....	578
<code>print-first-page-number</code> .....	577
<code>print-page-number</code> .....	577
<code>printAccidentalNames</code> .....	247
<code>printNotesLanguage</code> .....	247
<code>printOctaveNames</code> .....	247
<code>\property-recursive</code> .....	793
<code>\propertyOverride</code> .....	851
<code>\propertyRevert</code> .....	851
<code>\propertySet</code> .....	851
<code>\propertyTweak</code> .....	851
<code>\propertyUnset</code> .....	852
<code>\pt</code> .....	663
<code>\pushToTag</code> .....	545, 852
<code>\put-adjacent</code> .....	761

## Q

<code>\quilisma</code> .....	480, 486
<code>quotedCueEventTypes</code> .....	226
<code>quotedEventTypes</code> .....	226
<code>\quoteDuring</code> .....	224, 227, 852

## R

<code>r</code> .....	62
<code>R</code> .....	65
<code>ragged-bottom</code> .....	570
<code>ragged-last</code> .....	574, 611
<code>ragged-last-bottom</code> .....	570
<code>ragged-right</code> .....	574, 611
<code>\raise</code> .....	270, 761
<code>\raiseNote</code> .....	447, 852
<code>\reduceChords</code> .....	86, 852
<code>\relative</code> .....	2, 6, 14, 357, 852
<code>\remove</code> .....	631
<code>remove-empty</code> .....	216
<code>remove-first</code> .....	216
<code>remove-grace-property</code> .....	123
<code>remove-layer</code> .....	218
<code>\RemoveAllEmptyStaves</code> .....	215, 856
<code>\RemoveEmptyStaves</code> .....	215, 856
<code>\removeWithTag</code> .....	541, 852
<code>\repeat</code> .....	159
<code>\repeat percent</code> .....	170
<code>\repeat tremolo</code> .....	173
<code>repeatCommands</code> .....	166
<code>repeatCountVisibility</code> .....	172
<code>\repeatTie</code> .....	59, 161, 313
<code>\replace</code> .....	745
<code>\resetRelativeOctave</code> .....	6, 852
<code>\responsum</code> .....	481
<code>\rest</code> .....	62, 777
<code>\rest-by-number</code> .....	777
<code>rest-event</code> .....	226
<code>restNumberThreshold</code> .....	234
<code>restrainOpenStrings</code> .....	373
<code>\retrograde</code> .....	15, 852
<code>\reverseturn</code> .....	128, 800
<code>\revert</code> .....	649
<code>\revertTimeSignatureSettings</code> .....	72, 852
<code>\rfz</code> .....	131
<code>rgb-color</code> .....	244
<code>\rheel</code> .....	128, 801
<code>RhythmicStaff</code> .....	201
<code>\right-align</code> .....	269, 761
<code>\right-brace</code> .....	793
<code>\right-column</code> .....	761
<code>right-margin</code> .....	574
<code>\rightHandFinger</code> .....	414, 852
<code>\roman</code> .....	745
<code>\romanStringNumbers</code> .....	366, 370
<code>\rotate</code> .....	762
<code>\rounded-box</code> .....	274, 773
<code>\rtoe</code> .....	128, 801

# S

s	64
\sacredHarpHeads	44
\sacredHarpHeadsMinor	45
\sans	745
\scale	773
\scaleDurations	57, 81, 852
\score	506, 510, 778
\score-lines	796
score-markup-spacing	572
score-system-spacing	572
scoreTitleMarkup	520
\segno	128, 801
self-alignment-X	590
\semicirculus	478, 802
\semiflat	779
\semiGermanChords	450
\semisharp	779
\sesquiflat	779
\sesquisharp	780
\set	91, 646, 652
set-global-fonts	284
set-global-staff-size	581
\settingsFrom	852
\sf	131
\sff	131
\sfz	131
\shape	684, 852
\sharp	780
\shiftDurations	853
\shiftOff	186
\shiftOn	186
\shiftOnn	186
\shiftOnnn	186
short-indent	222, 575
\shortfermata	128, 801
show-available-fonts	283
showFirstLength	549, 807
\showKeySignature	431
showLastLength	549, 807
\showStaffSwitch	358
\signumcongruentiae	128, 802
\simple	745
\single	529, 655, 853
\skip	64, 312, 853
skipBars	234
skipTypesetting	549
slashChordSeparator	452
\slashed-digit	793
\slashedGrace	120, 853
\slashSeparator	578
\slashturn	128, 800
slur-event	226
\slurDashed	140
\slurDashPattern	141, 853
\slurDotted	140
\slurDown	140
\slurHalfDashed	140
\slurHalfSolid	140
\slurNeutral	140
\slurSolid	140
\slurUp	141
\small	235, 268, 746
\smallCaps	746

\smaller	266, 268, 746
\snappizzicato	128, 801
\sostenutoOff	361
\sostenutoOn	361
\sourcefileline	511
\sourcefilename	511
\southernHarmonyHeads	44
\southernHarmonyHeadsMinor	45
\sp	131
spacing	607
\spacingTweaks	853
Span_stem_engraver	359
\spp	131
\staccatissimo	128, 800
\staccato	128, 800
staff-affinity	590
staff-padding	241
\staff-space	663
staff-staff-spacing	590
Staff_midiInstrument	564
Staff_collecting_engraver	262
Staff_symbol_engraver	215
staffgroup-staff-spacing	590
start-repeat	166
startAcciaccaturaMusic	123
startAppoggiaturaMusic	123
startGraceMusic	123
\startGroup	251
\startStaff	208, 212
\startTrillSpan	155
\stdBass	785
\stdBassIV	786
\stdBassV	787
\stdBassVI	788
stem-spacing-correction	607
\stemDown	245
stemLeftBeamCount	101, 102
\stemNeutral	245
stemRightBeamCount	101, 102
\stemUp	245
\stencil	794
stopAcciaccaturaMusic	123
stopAppoggiaturaMusic	123
stopGraceMusic	123
\stopGroup	251
\stopped	128, 425, 429, 801
\stopStaff	208, 212, 215
\stopTrillSpan	155
\storePredefinedDiagram	404, 409, 853
strictBeatBeaming	97
\string-lines	796
stringNumberOrientations	240
\stringTuning	387, 853
stringTunings	386, 400
strokeFingerOrientations	240, 414
\stropho	480, 486
\strut	794
\styledNoteHeads	853
\sub	267, 746
subdivideBeams	96
suggestAccidentals	130, 473
\super	267, 747
sus	446
\sustainOff	361
\sustainOn	361



system-count .....	576
system-separator-markup .....	578
system-system-spacing .....	572
systems-per-page .....	576

## T

\tabChordRepeats .....	373, 853
\tabChordRepetition .....	853
\tabFullNotation .....	372
\table .....	797
\table-of-contents .....	538, 797
TabStaff .....	201, 371
TabVoice .....	371
\tag .....	541, 853
\tagGroup .....	544, 853
\taor .....	431
teaching .....	36
\teeny .....	235, 268, 747
\tempo .....	74
\temporary .....	655, 689, 853
\tenuto .....	128, 800
text .....	362, 747
\textLengthOff .....	67, 69, 257
\textLengthOn .....	67, 69, 135, 257
\textSpannerDown .....	258
\textSpannerNeutral .....	258
\textSpannerUp .....	258
\thumb .....	128, 240
\tie .....	747
TieColumn .....	61
\tied-lyric .....	780
\tieDashed .....	59
\tieDashPattern .....	59, 854
\tieDotted .....	59
\tieDown .....	59
\tieHalfDashed .....	59
\tieHalfSolid .....	59
\tieNeutral .....	59
\tieSolid .....	59
\tieUp .....	59
tieWaitForNote .....	61
\time .....	70, 91, 854
\times .....	854
timeSignatureFraction .....	81
\tiny .....	235, 268, 748
tocFormatMarkup .....	538
tocIndentMarkup .....	538
\tocItem .....	538, 854
tocItemMarkup .....	538
\tocItemWithDotsMarkup .....	537
tocTitleMarkup .....	538
top-margin .....	570
top-markup-spacing .....	572
top-system-spacing .....	572
toplevel-bookparts .....	807
toplevel-scores .....	807
\translate .....	270, 762
\translate-scaled .....	270, 762
\transparent .....	794
\transpose .....	6, 11, 14, 854
\transposedCueDuring .....	231, 854
\transposition .....	28, 224, 854
\treCorde .....	361

tremolo .....	173
\triangle .....	275, 773
\trill .....	128, 155, 800
\tuplet .....	51, 81, 854
tuplet-slur .....	52
\tupletDown .....	52
\tupletNeutral .....	52
TupletNumber .....	53
\tupletSpan .....	53, 854
tupletSpannerDuration .....	53
\tupletUp .....	52
\turn .....	128, 800
\tweak .....	650, 652, 854
two-sided .....	574
\type .....	638
\typewriter .....	748

## U

\unaCorda .....	361
\underline .....	266, 748
\undertie .....	749
\undo .....	655, 855
unfold .....	168
\unfolded .....	855
\unfoldRepeats .....	560, 855
\unHideNotes .....	242
\unset .....	647
\upbow .....	128, 366, 801
\upmordent .....	128, 800
\upprall .....	128, 800
\upright .....	749

## V

\varcoda .....	128, 801
VaticanaStaff .....	201, 476
VaticanaVoice .....	476
\vcenter .....	763
\verbatim-file .....	794
\version .....	511
\versus .....	481
VerticalAxisGroup .....	590
\verylongfermata .....	128, 801
\veryshortfermata .....	128, 801
\virga .....	480, 486
\virgula .....	478
Voice .....	181
voice .....	30, 32
\voiceFour .....	181
\voiceFourStyle .....	185
\voiceNeutralStyle .....	185
\voiceOne .....	181
\voiceOneStyle .....	185
\voices .....	184, 855
\voiceThree .....	181
\voiceThreeStyle .....	185
\voiceTwo .....	181
\voiceTwoStyle .....	185
\void .....	565, 855
\volta .....	855
Volta_engraver .....	164
\vshape .....	855
\vspace .....	271, 763

## W

<code>\walkerHeads</code> .....	44
<code>\walkerHeadsMinor</code> .....	45
<code>whichBar</code> .....	111
<code>\whiteout</code> .....	794
<code>\whiteTriangleMarkup</code> .....	451
<code>\with</code> .....	631, 635
<code>\with-color</code> .....	243, 795
<code>\with-dimensions</code> .....	795
<code>\with-dimensions-from</code> .....	795
<code>\with-link</code> .....	795
<code>\with-outline</code> .....	796
<code>\with-url</code> .....	774
<code>\withMusicProperty</code> .....	855

<code>\woodwind-diagram</code> .....	783
<code>\wordwrap</code> .....	273, 764
<code>\wordwrap-field</code> .....	763
<code>\wordwrap-internal</code> .....	798
<code>\wordwrap-lines</code> .....	279, 798
<code>\wordwrap-string</code> .....	764
<code>\wordwrap-string-internal</code> .....	798

## X

<code>x11-color</code> .....	243, 244
<code>X-offset</code> .....	590
<code>\xNote</code> .....	41, 855
<code>\xNotesOff</code> .....	41
<code>\xNotesOn</code> .....	41

## Apéndice E Índice de LilyPond

Además de todas las instrucciones y palabras clave de LilyPond, este índice es una lista de términos musicales y las palabras que tienen relación con cada uno de ellos, con enlaces a aquellas secciones del manual que describen o se ocupan de dicho término.

Las entradas en letra inclinada apuntan a ubicaciones (principalmente secciones del tipo ‘Véase también’s) que contienen enlaces externos a otros archivos de documentación de LilyPond como la Referencia de funcionamiento interno o el Glosario.

<b>!</b>	<b>:</b>
! ..... 7	: ..... 174
\! ..... 131	
<b>"</b>	<b>&lt;</b>
" " ..... 117	<...> ..... 176
	<> ..... 176, 354
	< ..... 176
<b>%</b>	\< ..... 131
% ..... 507, 511	<b>=</b>
%{ ... %} ..... 507, 511	\= ..... 140, 855
<b>,</b>	= ..... 10
' ..... 1	<b>&gt;</b>
<b>(</b>	> ..... 176
( ..... 140	\> ..... 131
\( ..... 143	<b>?</b>
<b>)</b>	? ..... 7
) ..... 140	<b>[</b>
\) ..... 143	[ ..... 100
	\[ ..... 466
<b>,</b>	<b>]</b>
, ..... 1	] ..... 100
<b>—</b>	\] ..... 466
- ..... 128	<b>^</b>
-! ..... 800	^ ..... 446
-+ ..... 801	<b>_</b>
-- ..... 800	_ ..... 294
- ..... 800	<b> </b>
-> ..... 800	..... 117
-^ ..... 800	<b>~</b>
-_ ..... 800	~ ..... 58
<b>•</b>	
..... 50	
<b>/</b>	
/ ..... 446	
/+ ..... 446	

ángulo, regulador en .....	677
árabe, ejemplo de música .....	502
árabe, improvisación .....	501
árabe, música .....	497
árabe, maqam .....	497
árabe, notas, nombres .....	498
árabe, plantilla de música .....	502
árabe, semi-bemol .....	499
árabes, armaduras .....	501
índice general .....	535
índice general, funciones personalizadas .....	537
ópera .....	331
órgano, indicación de pedal .....	801
órgano, marca de pedal .....	128

## 1

15ma .....	25
15mb .....	25

## 8

8va .....	25
8vb .....	25

## A

<i>a due</i> .....	197
<i>a due</i> , partes .....	192
<i>Añadir y eliminar grabadores</i> .....	85
abajo, arco .....	366
‘abierta’, articulación .....	128, 429, 801
abierto, high hat .....	128, 802
<code>\abs-fontsize</code> .....	266, 740
absoluta .....	1
absoluta, especificación, de la octava .....	1
absoluta, introducción, de la octava .....	1
<code>\absolute</code> .....	844
absolutos, matices .....	131
acampanado, regulador .....	135
<i>accent</i> .....	130
<code>\accent</code> .....	128, 800
<code>\accentus</code> .....	478, 802
‘accentus’, articulación gregoriana .....	478
‘accentus’, articulación, gregoriana .....	802
<code>\accepts</code> .....	638, 639, 640
<i>acciaccatura</i> .....	124
<i>acciaccatura</i> .....	120
<code>\acciaccatura</code> .....	120, 844
<i>acciaccatura</i> , de varias notas .....	124
<i>Accidental</i> .....	8, 37, 477
<i>accidental</i> .....	473, 478, 489
<i>accidental</i> .....	6
<code>\accidental</code> .....	774
<i>accidental-interface</i> .....	8
<i>accidental-suggestion-interface</i> .....	37
<i>Accidental engraver</i> .....	8, 37, 474
<i>AccidentalCautionary</i> .....	8
<i>AccidentalPlacement</i> .....	37
<code>\accidentalStyle</code> .....	30, 844
<i>AccidentalSuggestion</i> .....	37, 474
<i>AccidentalSuggestion</i> .....	130
<i>Accordion Registers</i> .....	363

acento .....	128
‘acento’, articulación .....	128, 800
<i>Acerca de la no anidabilidad de</i> <i>llaves y ligaduras</i> .....	142, 144
acorde .....	176, 442, 448
acorde alterado .....	444
acorde arpegiado .....	152
acorde de jazz .....	442, 450
acorde extendido .....	444
acorde quebrado .....	152
acorde, alteración de notas .....	446
acorde, alteraciones dentro de .....	37
acorde, de pentagrama cruzado .....	359
acorde, de potencia .....	417
acorde, de quinta vacía .....	417
acorde, diagrama automático de .....	411
acorde, diagramas de trastes con nombres de .....	402
acorde, digitación de .....	240
acorde, división entre pentagramas con <code>\autoChange</code> .....	358
acorde, eliminar repetidos .....	449
acorde, formas de, en instrumentos de trastes .....	404
acorde, glissando en tablatura .....	384
acorde, hoja guía de .....	449
acorde, introducción relativa de la octava en .....	5
acorde, inversión .....	447
acorde, mayor 7, disposición .....	454
acorde, modificación de una nota .....	650
acorde, modo de .....	442
acorde, nombre alternativo .....	450
acorde, nombre de .....	448
acorde, nombres de .....	442
acorde, nombres de, excepciones .....	453
acorde, realización .....	447
acorde, repetición de volta, por debajo .....	454
acorde, rotación .....	447
acorde, separador, modificación .....	455
acorde, supresión en .....	446
acorde, suprimir repetido .....	408
acorde, vacío .....	123, 176, 354
acorde, variantes de .....	443
acorde, y altura relativa .....	177
acorde, y ligadura .....	59
acordeón .....	362
acordeón, cambios en el .....	362
acordeón, símbolo de registro .....	362
acordeón, signo de cambio en el .....	362
acordeón, signo de discanto .....	362
acordes, bajo de .....	446
acordes, diagramas de .....	390
acordes, repetición de .....	178, 373
acortar, corchete de primera y segunda vez .....	163
‘acro arriba’, indicación de arco .....	128
<code>add-bar-glyph-print-procedure</code> .....	859
<code>add-grace-property</code> .....	859
<code>add-grace-property</code> .....	123
<code>add-music-fonts</code> .....	859
<code>add-new-clef</code> .....	859
<code>add-simple-time-signature-style</code> .....	859
<code>add-stem-support</code> .....	241
<code>add-stroke-glyph</code> .....	859
<code>add-stroke-straight</code> .....	860
<code>add-toc-item!</code> .....	537
<code>\addChordShape</code> .....	404, 844
adding white background, to text .....	794

<code>\addInstrumentDefinition</code> .....	844	alteración mensural .....	473
<code>additionalPitchPrefix</code> .....	451	alteración no restablecer, estilo de .....	36
<code>\addlyrics</code> .....	288, 290, 291	alteración recordatoria .....	7
<code>\addQuote</code> .....	224, 844	alteración, accidental .....	30
adición en acordes .....	445	alteración, automática .....	30
adicional, evitar becuadro .....	7	alteración, bajo cifrado, posición .....	459
adicional, interna, línea .....	208	alteración, cadencias .....	79
adicional, línea .....	208	alteración, coral con precaución .....	34
adicional, modificar línea .....	208	alteración, en acordes .....	37
adicional, voz, en música polifónica .....	189	alteración, en varias voces .....	33
adicional, voz, para gestionar saltos .....	585	alteración, estilo coral .....	34
adorno, letra y notas de .....	322	alteración, estilo de .....	30
adorno, nota de, seguimiento .....	121	alteración, estilo de piano .....	33
adorno, nota, trucaje .....	123	alteración, estilo de voz de .....	32
adorno, nota, y espaciado estricto .....	123	alteración, estilo de voz de precaución moderno ...	33
adorno, sincronización de notas de .....	124	alteración, estilo de, olvidar .....	37
<code>\aeolian</code> .....	23	alteración, estilo dodecafónico sin repetir .....	36
afinación del laúd .....	420	alteración, estilo dodecafónico-primero .....	36
afinación, del banjo .....	419	alteración, estilo moderno .....	32
afinación, no occidental .....	497	alteración, estilo moderno de .....	33
<code>\afterGrace</code> .....	121, 844	alteración, estilo moderno de precaución .....	32
<code>afterGraceFraction</code> .....	807	alteración, estilo predeterminado de .....	30
agógó .....	802	alteración, estilo, dodecafónico .....	35
agrupación de grupos especiales .....	52	alteración, música sin metro .....	79
agrupar, pulsos .....	97	alteración, moderna .....	33
Aiken, cabezas de nota con forma .....	44	alteración, ocultar, en notas ligadas al	
Aiken, cabezas de nota, variante fina .....	46	principio de un sistema .....	7
<code>\aikenHeads</code> .....	44	alteración, piano de precaución, estilo de .....	34
<code>\aikenHeadsMinor</code> .....	45	alteración, trino con .....	157
<code>\aikenThinHeads</code> .....	44	alteración, y notas simultáneas .....	37
<code>\aikenThinHeadsMinor</code> .....	45	alteraciones de la musica ficta .....	473
aislada, altura .....	50	alteraciones estilo moderno .....	32
aislada, duración .....	49	alteraciones gregorianas .....	477
ajustado a palabras, texto .....	273	alteraciones kievanas .....	488
ajuste de línea, texto con .....	273	alteraciones, estilo de, neo-moderno .....	34
ajuste del símbolo del pentagrama .....	664	alteraciones, estilo predeterminado .....	31
al aire, indicación de cuerda .....	366	alteraciones, estilo teaching (enseñanza) .....	36
<i>al niente</i> .....	137	alteraciones, estilo, moderno-precaución .....	32
al niente, regulador .....	135	alteraciones, estilo, neo-moderno de precaución ...	35
alfa, transparencia .....	243	alteraciones, estilo, neo-moderno de voz .....	35
<code>\alias</code> .....	638	alteraciones, estilo, neo-modern-voice-cautionary ...	35
<code>alignAboveContext</code> .....	214, 642	<code>alterations-in-key</code> .....	860
<code>alignBelowContext</code> .....	214, 310, 642	<code>\alterBroken</code> .....	688, 844
alineación a un objeto .....	681	alternativa, tabla de trastes .....	409
alineación a una cadenza (cadencia) .....	125	<code>\alternative</code> .....	159
<i>Alineación de la letra a una melodía</i> .....	290, 298	alternativo, estilo de la breve .....	50
alineación vertical de inscripciones de texto .....	135	alternativo, finales, en repeticiones explícitas .....	168
alineación vertical, matices .....	135	alternativo, nombre de acorde .....	450
alineación, compás, número .....	115	alto varC, clave .....	734
alineación, horizontal, letra .....	307	alto, clave .....	734
alineación, instrucciones de texto .....	273	alto, clave de .....	18
alineación, vertical, de texto .....	270	altura .....	1
alinear, elementos de marcado .....	269	altura, aislada .....	50
alinear, texto .....	269	altura, nombres de, otros idiomas .....	8
alinear, texto de marcado .....	269	altura, predeterminada .....	50
alist .....	805	altura, transporte de .....	11
<code>alist-&gt;hash-table</code> .....	860	altura, transposición 'inteligente' .....	13
<i>All layout objects</i> .....	237, 646, 653, 678, 806	<i>Alturas y armaduras</i> .....	6, 8, 23, 25, 501
<code>allow-volta-hook</code> .....	860	Amazing Grace, ejemplo de gaita .....	432
<code>\allowPageTurn</code> .....	589, 844	<i>Ambitus</i> .....	40
<code>\allowVoltaHook</code> .....	844	<i>ambitus</i> .....	40, 287
alteración con paréntesis .....	7	<i>ambitus</i> .....	38
alteración de cuartos de tono .....	8	<i>ambitus-interface</i> .....	40
alteración de precaución .....	7	<i>Ambitus engraver</i> .....	40
alteración en nota ligada .....	7	<i>AmbitusAccidental</i> .....	40

<code>\ambitusAfter</code> .....	844	<code>\arpeggioParenthesis</code> .....	152
<code>AmbitusLine</code> .....	40	<code>\arpeggioParenthesisDashed</code> .....	152
<code>AmbitusLine</code> .....	39	arpegiado, acorde, símbolos especiales .....	152
<code>AmbitusNoteHead</code> .....	40	arpegio.....	152
análisis musicológico.....	251	arpegio con paréntesis de pentagrama cruzado ...	155
análisis, corchete de, con etiqueta.....	253	arpegio, corchete de pentagrama cruzado .....	360
anacrusa.....	78	arpegio, corchete, señalar voces divididas .....	330
anacrusa, en repeticiones .....	160	arpegio, entre pautas.....	153
<i>anacrusis</i> .....	79	arpegio, entre voces .....	154
analizador léxico .....	806	arpegio, y ligadura .....	61
analizador sintáctico.....	807	<i>Arquitectura flexible</i> .....	255, 256
<i>Ancient notation</i> .....	468, 473, 479, 480	<i>Arreglar notación con superposiciones</i> .....	355, 356
<code>angle-0-2pi</code> .....	860	arriba, arco .....	366
<code>angle-0-360</code> .....	860	<code>\arrow-head</code> .....	275, 765
anidada, repetición.....	166	<code>arrow-stencil</code> .....	860
<i>Anidado de expresiones musicales</i> .....	212, 215, 642	<code>arrow-stencil-maker</code> .....	860
anidados, corchetes de pentagrama .....	205	articulación .....	128
anidamiento, de pentagramas.....	205	articulación, abierta.....	128, 429, 801
<code>annotate-spacing</code> .....	618	articulación, acento.....	128, 800
anthem .....	328	articulación, espressivo .....	128, 800
antiguas, claves.....	18	articulación, gregoriana, <i>accentus</i> .....	802
<i>Aparece un pentagrama de más</i> .....	160, 170, 642	articulación, gregoriana, <i>circulus</i> .....	802
<code>\appendToTag</code> .....	545, 845	articulación, gregoriana, <i>ictus</i> .....	802
<code>\applyContext</code> .....	627, 845	articulación, gregoriana, <i>semicirculus</i> .....	802
<code>\applyMusic</code> .....	845	articulación, gregoriano, <i>accentus</i> .....	478
<code>\applyOutput</code> .....	845	articulación, gregoriano, <i>circulus</i> .....	478
<i>appoggiatura</i> .....	124	articulación, gregoriano, <i>episemFinis</i> .....	478
appoggiatura.....	120	articulación, gregoriano, <i>episemInitium</i> .....	478
<code>\appoggiatura</code> .....	120, 845	articulación, gregoriano, <i>ictus</i> .....	478
Arabic key signatures.....	499	articulación, gregoriano, <i>semicirculus</i> .....	478
<code>\arabicStringNumbers</code> .....	370	articulación, marcato .....	128, 800
arco abajo.....	128, 801	articulación, nota tapada.....	128
‘arco abajo’, indicación de arco .....	801	articulación, portato .....	128, 800
arco arriba .....	128, 801	articulación, semiabierta .....	128, 801
‘arco arriba’, indicación de arco.....	801	articulación, <i>staccatissimo</i> .....	128, 800
arco, abajo, indicación.....	366	articulación, <i>staccato</i> .....	128, 800
arco, arriba, indicación .....	366	articulación, tapada .....	429, 801
arco, indicación .....	366	articulación, tenuto.....	128, 800
armónico, en instrumento de trastes .....	416	articulación, valores	
armónico, en tablaturas .....	379	predeterminados, modificación.....	129
armónico, flageolet .....	128	articulaciones .....	800
armónico, natural .....	366	articulaciones gregorianas.....	478
armónicos artificiales .....	367	<i>Articulaciones y matices dinámicos</i> .....	137
armónicos de flageolet.....	801	<code>\articulate</code> .....	564
armónicos en tablatura.....	376	articulate script .....	564
armónicos, cabezas de nota .....	41	articulate, script.....	564
Armadura de la tonalidad .....	23	articulate.ly.....	564
armadura gregoriana .....	477	<code>articulation-event</code> .....	226
armadura mensural.....	473	artificiales, armónicos.....	367
armadura, evitar becuadros.....	24	<code>\ascendens</code> .....	480, 486
armadura, no tradicional .....	24	aspas, cabezas de nota.....	41
armadura, visibilidad después de un		<code>\assertBeamQuant</code> .....	845
cambio explícito.....	673	<code>\assertBeamSlope</code> .....	845
arpa .....	364	<code>associatedVoice</code> .....	288, 290, 323
arpa sacra, cabezas de nota con forma.....	44	<code>\auctum</code> .....	480, 486
arpa, diagrama de pedal.....	364	<code>aug</code> .....	443
arpa, pedal.....	364	<code>\augmentum</code> .....	486
<i>Arpeggio</i> .....	155, 653	<code>auto-first-page-number</code> .....	577
<i>arpeggio</i> .....	155	<code>\auto-footnote</code> .....	789
<code>\arpeggio</code> .....	152	<i>Auto_beam_engraver</i> .....	91, 99
<code>arpeggio-direction</code> .....	152	autobarrado, propiedades para	
<code>\arpeggioArrowDown</code> .....	152	indicaciones de compás.....	71
<code>\arpeggioArrowUp</code> .....	152	<code>autoBeaming</code> .....	91, 623
<code>\arpeggioBracket</code> .....	152	<code>\autoBeamOff</code> .....	88, 359
<code>\arpeggioNormal</code> .....	152	<code>\autoBeamOn</code> .....	88

<code>\autoBreaksOff</code> .....	582
<code>\autoBreaksOn</code> .....	582
<code>\autoChange</code> .....	356, 845
<code>\autoChange</code> y el modo relativo .....	357
<code>AutoChangeMusic</code> .....	358
<code>\autoLineBreaksOff</code> .....	582
<code>\autoLineBreaksOn</code> .....	582
automática, alteración .....	30
automática, combinación de partes .....	192
automático, cambio de pentagrama .....	356
automático, diagrama de acordes .....	411
automático, diagrama de traste .....	411
<code>automaticBars</code> .....	675
<code>\autoPageBreaksOff</code> .....	586
<code>\autoPageBreaksOn</code> .....	586
<code>Axis_group_engraver</code> .....	220, 596
ayuda, globos de .....	248

## B

Bézier, curva de, puntos de control .....	684
<code>Backend</code> .....	643, 646, 650
backslashed digit .....	789
<code>\backslashed-digit</code> .....	789
bajo .....	802
bajo acústico .....	802
bajo cifrado .....	456
bajo cifrado, alineación .....	461
bajo cifrado, líneas de extensión .....	459
bajo de los acordes .....	446
bajo, clave .....	734
bajo, clave de .....	18
balance en MIDI .....	563
<code>balloon-interface</code> .....	249
<code>Balloon_engraver</code> .....	249
<code>Balloon_engraver</code> .....	248
<code>\balloonGrobText</code> .....	248, 845
<code>\balloonLengthOff</code> .....	248
<code>\balloonLengthOn</code> .....	248
<code>\balloonText</code> .....	248, 845
<code>BalloonTextItem</code> .....	249
banjo, afinación del .....	419
banjo, tablatura de .....	369, 418
banjo-c-tuning .....	419
banjo-modal-tuning .....	419
banjo-open-d-tuning .....	419
banjo-open-dm-tuning .....	419
<code>\bar</code> .....	104, 111, 845
<code>bar-line::calc-break-visibility</code> .....	861
<code>bar-line::calc-glyph-name</code> .....	861
<code>bar-line::calc-glyph-name-for-direction</code> .....	861
<code>bar-line::compound-bar-line</code> .....	861
<code>bar-line::draw-filled-box</code> .....	861
<code>bar-line::widen-bar-extent-on-span</code> .....	861
<code>Bar_engraver</code> .....	450
<code>Bar_number_engraver</code> .....	117
barítono varC, clave .....	734
barítono varF, clave .....	734
barítono, clave de .....	18
<code>barCheckSynchronize</code> .....	117
<code>BarLine</code> .....	112
<code>BarNumber</code> .....	117
<code>BarNumber</code> .....	112
<code>\barNumberCheck</code> .....	118, 845

<code>barNumberVisibility</code> .....	112
barra cruzada .....	108
barra de compás, música sin metro .....	79
barra horizontal .....	378
barra, cadencias .....	79
barra, de compás .....	104
barra, de pentagrama cruzado .....	354
barra, de trémolo .....	173
barra, doblada, cambiar .....	89
barra, doble .....	104
barra, en ángulo .....	89
barra, en saltos de línea .....	89
barra, en tablatura .....	378
barra, estricta .....	97
barra, final .....	104
barra, finales con varias voces .....	98
barra, finales en la partitura .....	98
barra, grupo especial, cortar línea .....	55
barra, música sin metro .....	79
barra, manual .....	88, 100
barra, manual, abreviatura de dirección .....	100
barra, manual, notas de adorno .....	100
barra, <code>\partCombine</code> con <code>\autoBeamOff</code> .....	90
barra, progresiva .....	103
barra, reglas personalizadas .....	88
barra, salto de línea .....	89
barra, secundaria, orientación de .....	97
barra, subdivisión .....	96
barra, y letra .....	91
barra, y melismas .....	89
barradas, cabezas de nota .....	47
barrado, propiedades predefinidas	
para el compás .....	71
barras de compás manuales .....	105
barras, en música polimétrica .....	81
Bartók pizzicato .....	367
<code>bartype</code> .....	111
<code>base-length</code> .....	861
<code>base-shortest-duration</code> .....	607
<code>baseMoment</code> .....	91, 96
<code>BassFigure</code> .....	460, 461
<code>BassFigureAlignment</code> .....	460, 461
<code>BassFigureBracket</code> .....	460, 461
<code>BassFigureContinuation</code> .....	460, 461
<code>\bassFigureExtendersOff</code> .....	458
<code>\bassFigureExtendersOn</code> .....	458
<code>BassFigureLine</code> .....	460, 461
<code>\bassFigureStaffAlignmentDown</code> .....	461
<code>\bassFigureStaffAlignmentNeutral</code> .....	461
<code>\bassFigureStaffAlignmentUp</code> .....	461
basso continuo .....	456
batería .....	420, 422
<i>bayati</i> .....	501
<i>Beam</i> .....	91, 99, 103, 356, 386
<code>\beam</code> .....	765
<code>beam-exceptions</code> .....	861
<code>beam-interface</code> .....	91, 99, 103
<code>Beam_engraver</code> .....	91, 103
<code>BeamEvent</code> .....	91, 103
<code>\beamExceptions</code> .....	91, 845
<code>BeamForbidEvent</code> .....	91, 99
<code>beat-structure</code> .....	861
<code>beatStructure</code> .....	91, 96
becuadro .....	6
becuadro, adicional, evitar .....	7

becuadro, evitar en armadura.....	24
bemol.....	6
bemol, doble.....	6
bend-spanner::print.....	862
bend::arrow-head-stencil.....	861
bend::calc-bend-x-begin.....	861
bend::calc-bend-x-end.....	862
bend::target-cautionary.....	862
bend::text-string.....	862
\bendAfter.....	146, 845
\bendHold.....	845
\bendStartLevel.....	845
binding-offset.....	575
bisbiglando.....	364
Bison.....	807
blackmensural, clave.....	469, 734
\blackTriangleMarkup.....	451
blank-after-score-page-penalty.....	577
blank-last-page-penalty.....	577
blank-page-penalty.....	576
BNF.....	807
\bold.....	266, 740
bongó.....	802
bongó abierto.....	802
bongó agudo.....	802
bongó grave.....	802
bongó tapado.....	802
\book.....	507, 510
book-first-page.....	862
\bookOutputName.....	509, 845
\bookOutputSuffix.....	509, 845
\bookpart.....	508, 510, 586
bookTitleMarkup.....	520
bottom-margin.....	570
\box.....	274, 740
box-grob-stencil.....	862
box-stencil.....	863
brace.....	205
bracket.....	205
\bracket.....	138, 766
bracket.....	362
\bracket.....	274
bracketify-stencil.....	863
\break.....	582
break-align-symbols.....	681
break-alignable-interface::self-alignment-of-anchor.....	863
break-alignable-interface::self-alignment-opposite-anchor.....	863
break-alignment-interface.....	824, 825, 839
break-alignment-list.....	863
break-visibility.....	671
breakable.....	89
breakbefore.....	518
\breathe.....	144, 846
Breathing_sign_engraver.....	146
BreathingEvent.....	146
BreathingSign.....	146, 480
BreathingSign.....	145
breve.....	51, 64
\breve.....	49, 62
breve, nota, estilo alternativo.....	50
breve, silencio de.....	62
buscar, fuentes tipográficas disponibles.....	283

## C

cántico.....	339, 345
Cómo funcionan los archivos de entrada de LilyPond.....	512, 519
caída de tono (fall).....	146
cabasa.....	802
cabezas de nota antiguas.....	471, 488
cadencia.....	79
cadencia, alteraciones.....	79
cadencia, barras.....	79
cadencia, barras de compás.....	79
cadencia, números de compás.....	79
cadencia, saltos de línea en.....	81
cadencia, saltos de página en.....	81
cadenza.....	81, 126
cadenza (cadencia).....	125
cadenza (cadencia), alinear a.....	125
\cadenzaOff.....	79
\cadenzaOn.....	79
caesura.....	146, 478
caesura.....	145
\caesura.....	478
caja.....	802
caja acústica.....	802
caja china.....	802
caja circundante.....	664
caja electrónica.....	802
calc-harmonic-pitch.....	863
calderón.....	119, 128
calderón corto.....	128
calderón de Henze.....	128, 801
calderón largo.....	128
calderón o fermata.....	801
calderón, corto.....	801
calderón, largo.....	801
calderón, sobre la línea divisoria.....	260
calderón, sobre un silencio multicompa.....	66
callback.....	805
cambiar una propiedad.....	646
cambiar, nombre de instrumento.....	222
cambio automático de pentagrama.....	356
cambio de dedo.....	240
cambio de la fuente tipográfica.....	266
cambio, de pentagrama, manual.....	354
cambio, de tempo, sin indicación metronómica.....	77
canción.....	802
canchales.....	328
Canciones.....	288, 327
cantante, nombre del.....	320
capa.....	671
capo.....	395
\caps.....	740
carácter, especial, en modo de marcado.....	265
caracteres especiales.....	546
caracteres no ASCII.....	546
\cavum.....	480, 486
cejilla.....	395
cejilla, indicación de.....	390
cencerro.....	802
\center-align.....	269, 750
\center-column.....	271, 750
centered-stencil.....	863
centered-text-interface::print.....	863
centering column of text.....	750



centrado, matices, en música de piano .....	353	clave, de barítono .....	18
centrar texto en la página .....	272	clave, de contrabajo .....	18
cerradura .....	805	clave, de Do .....	18
cesura .....	145	clave, de Fa .....	18
<code>\change</code> .....	354	clave, de guía .....	18
<code>change-pitches</code> .....	863	clave, de mezzosoprano .....	18
changing direction of text column .....	751	clave, de Sol .....	18
<code>\char</code> .....	790	clave, de soprano .....	18
chart, fingering .....	430	clave, de tenor .....	18
<code>check-consistency</code> .....	574	clave, de violín .....	18
<code>check-context-path</code> .....	863	clave, Do .....	734
<code>check-grob-path</code> .....	864	clave, estilo .....	734
<code>check-music-path</code> .....	864	clave, estilos .....	469
<i>ChoirStaff</i> .....	205, 207, 329	clave, Fa .....	734
choral .....	34	clave, francesa .....	18, 734
choral-cautionary .....	34	clave, G2 .....	734
<i>chord</i> .....	177, 443, 450	clave, GG .....	734
<i>Chord_name engraver</i> .....	450	clave, kievana .....	734
<code>chordChanges</code> .....	408, 449	clave, música antigua .....	734
<code>\chordmode</code> .....	6, 14, 402, 660	clave, mensural .....	734
<i>ChordName</i> .....	450	clave, mezzosoprano .....	734
<code>chordNameExceptions</code> .....	452, 453	clave, percusión .....	734
<code>chordNameLowercaseMinor</code> .....	451	clave, Petrucci .....	734
<i>ChordNames</i> .....	220, 450	clave, Sol .....	734
<i>ChordNames</i> .....	402	clave, soprano .....	734
<i>ChordNames</i> , con líneas de compás .....	454	clave, subbajo .....	734
<code>chordNameSeparator</code> .....	452, 455	clave, tablatura .....	734
<code>chordNoteNamer</code> .....	451	clave, tenor .....	734
<code>chordPrefixSpacer</code> .....	453	clave, tenor Sol .....	734
<code>\chordRepeats</code> .....	373, 846	clave, tenor varC .....	734
<code>chordRootNamer</code> .....	451	clave, treble .....	734
<i>Chords</i> ... 443, 444, 447, 450, 455, 457, 460, 461, 695		clave, trucaje de propiedades .....	20
<code>\chords</code> .....	449, 660	clave, varbaritone .....	734
chorus en MIDI .....	563	clave, varC .....	734
Christian Harmony, cabezas de nota con forma ... 44		clave, violín .....	734
<i>church mode</i> .....	25	clave, visibilidad de la transposición .....	676
cifrado americano .....	442	clave, visibilidad después de un	
cifrado, bajo .....	456	cambio explícito .....	673
cifrado, bajo, alteración, posición .....	459	claves .....	802
<code>\circle</code> .....	274, 766	<i>clef</i> .....	469, 477, 487
<code>circle-stencil</code> .....	864	<i>Clef</i> .....	22, 469
circling text .....	766	<i>clef</i> .....	6
<code>\circulus</code> .....	478, 802	<code>\clef</code> .....	18, 846
‘circulus’, articulación gregoriana .....	478	<i>clef-interface</i> .....	22, 469
‘circulus’, articulación, gregoriana .....	802	<code>clef-transposition-markup</code> .....	864
citar otras voces .....	224, 227	<i>Clef engraver</i> .....	22, 469
clave .....	18, 476, 487	<i>ClefModifier</i> .....	22, 469
clave blackmensural .....	469	<code>clip-regions</code> .....	548
clave de música antigua .....	469	<i>cluster</i> .....	181
clave de percusión .....	421	<i>cluster</i> .....	181
clave de Petrucci .....	469	<i>Cluster_spanner engraver</i> .....	181
clave de tablatura .....	388	<i>ClusterSpanner</i> .....	181
clave kievana .....	469	<i>ClusterSpannerBeacon</i> .....	181
clave mensural .....	469	<code>\cm</code> .....	663
clave moderntab .....	388	coda .....	128
clave transpositora .....	18	<code>\coda</code> .....	128, 801
clave, alto .....	734	‘coda de variación’, signo .....	801
clave, alto varC .....	734	coda de variante .....	128
clave, antigua .....	18	coda de variante, símbolo .....	128
clave, bajo .....	734	‘coda de variante’, signo .....	128
clave, barítono varC .....	734	‘coda’, signo .....	119, 128, 801
clave, baritone varF .....	734	‘coda’, sobre la línea divisoria .....	260
clave, blackmensural .....	734	colisión .....	186
clave, de alto .....	18	colisión, del número de compás .....	117
clave, de bajo .....	18	colisión, ignorar .....	180, 191

colisión, voces de pentagrama cruzado .....	355
<code>collect-book-music-for-book</code> .....	864
<code>collect-bookpart-for-book</code> .....	864
<code>collect-music-aux</code> .....	864
<code>collect-music-for-book</code> .....	864
<i>Colocación de los objetos</i> .....	129, 130, 257
colocación, de los silencios multicomás .....	67
color .....	243
<code>color</code> .....	243
color de <code>x11</code> .....	243
color en acordes .....	244
color <code>rgb</code> .....	244
color, códigos de CSS .....	243
color, voz .....	185
coloreada, nota .....	243
coloreada, nota de acorde .....	244
coloreado de un objeto .....	671
coloreado, objeto .....	243
colorear notas .....	243
colorear objetos .....	243
colores semitransparentes .....	243
colores, lista de .....	710
coloring text .....	795
<code>\column</code> .....	271, 750
<code>\column-lines</code> .....	796
columna, texto en .....	263
columnas, texto .....	271
coma, intervalos de .....	503
<i>Combinar notas para formar acordes</i> .....	177
combinar, partes, cambiar el texto .....	196
<code>\combine</code> .....	275, 750
comentario .....	507, 511
comentario de bloque .....	507, 511
comentario de una línea .....	507, 511
comentario de varias líneas .....	507, 511
comienzo de repetición .....	166
comienzo del sistema .....	202
comillas, en la letra .....	287, 294
<i>Common Practice Period</i> .....	10, 497
<code>common-shortest-duration</code> .....	607
compás de anacrusa .....	78
compás parcial .....	78
compás polimétrico .....	81
compás, agrupación .....	97
compás, barras, invisibles .....	104
compás, completo, silencios de .....	65
compás, comprobación .....	117
compás, comprobación de .....	117
compás, compuesto, indicación de .....	83
compás, definir línea de .....	108
compás, doble .....	81
compás, estilo de la indicación de .....	70
compás, indicación de, ajustes predeterminados ..	71
compás, indicación de, en medio de un compás ...	78
compás, indicación de, estilos de .....	470
compás, indicación de, imprimir solo el numerador .....	73
compás, indicación de, mensural .....	470
compás, línea de, dentro de <code>ChordNames</code> .....	454
compás, número de .....	126
compás, número de, formato .....	114
compás, número, alineación .....	115
compás, número, comprobación .....	117
compás, número, eliminar .....	116
compás, número, imprimir a intervalos regulares .....	113
compás, número, imprimir en el primer compás ..	112
compás, números de .....	112
compás, parcial, en himnos .....	350
compás, propiedades, restaurar los valores predeterminados .....	72
compás, repetición de .....	170
compás, repeticiones y números de .....	166
compás, sub-agrupar .....	97
compás, visibilidad de la indicación de .....	70
compases distintos al mismo tiempo .....	632
<i>Completion_heads_engraver</i> .....	85
<i>Completion_heads_engraver</i> .....	84
<i>Completion_rest_engraver</i> .....	85
<i>Completion_rest_engraver</i> .....	84
<code>\compound-meter</code> .....	774
<code>\compoundMeter</code> .....	83, 846
compresión de música .....	57
<code>\compressEmptyMeasures</code> .....	233
<code>\compressMMRests</code> .....	65, 67, 233, 846
comprobación de compás con repeticiones .....	160
compuesto, indicación de compás .....	83
<code>\concat</code> .....	751
concatenating text .....	751
<i>concert pitch</i> .....	30
condensar silencios .....	70
conformar ligaduras .....	684
conga .....	802
conga abierta .....	802
conga aguda .....	802
conga grave .....	802
conga tapada .....	802
<code>\consists</code> .....	631, 638
constante, regulador de tipo .....	135
<i>constante-hairpin</i> .....	864
<i>Construcción de elementos de marcado en Scheme</i> .....	139, 536
<code>construct-chord-elements</code> .....	865
contador, repetición de porcentaje .....	171
contador, visibilidad de repetición de porcentaje .....	172
contemporáneo, glissando .....	148
<code>\context</code> .....	624, 633
<code>\context</code> dentro de un bloque <code>\layout</code> .....	633
<code>context-spec-music</code> .....	866
<code>context-spec-music</code> .....	189
<i>ContextChange</i> .....	356
contexto implícito .....	641
contexto, cambiar propiedades predeterminadas de .....	633
contexto, definición de nuevos .....	638
contexto, definiciones con MIDI .....	558
contexto, orden de disposición .....	640
<i>Contextos y grabadores</i> .....	183, 622
contextos, creación y referenciación de .....	624
contextos, mantener vivos .....	628
contextos, tiempo de vida .....	628
<i>Contexts</i> .....	591, 593, 597, 622
<i>Contexts and engravers</i> .....	622
continua, ligadura .....	140
contrabajo, clave de .....	18
control, alturas de .....	10
control, trucar un punto de .....	652
controlling general text alignment .....	753

<b>controlpitch</b> .....	10
<i>Convenciones de nombres de</i>	
<i>objetos y propiedades</i> .....	806
<b>copy-repeat-chord</b> .....	866
copyright, símbolo de .....	547
coral, clave de tenor .....	18
coral, estilo de alteraciones accidentales .....	34
coral, estilo de alteraciones con precaución .....	34
corchete .....	245, 251
corchete de fraseo .....	251
corchete de fraseo con etiqueta .....	253
corchete de primera y segunda vez .....	166
corchete horizontal .....	251
corchete mensural .....	471
corchete vertical .....	202
corchete, al comienzo de un pentagrama único ...	204
corchete, cuadrado, al inicio de un grupo .....	203
corchete, de casilla de repetición, con texto .....	168
corchete, de pentagrama cruzado .....	360
corchete, recto .....	102
corchete, vertical, señalar voces divididas .....	330
corchetes, anidados .....	205
coro, sistema de .....	202
corto, calderón .....	128, 801
<b>count-list</b> .....	866
<b>countPercentRepeats</b> .....	171
<b>\cr</b> .....	131
<b>create-glyph-flag</b> .....	866
creating a table .....	797
creating empty text object .....	792
creating horizontal space, in text .....	755
creating text fraction .....	790
creating vertical space, in text .....	763, 794
<b>\cresc</b> .....	132
<i>crescendo</i> .....	137
<i>crescendo</i> .....	131
<b>crescendo-event</b> .....	226
<b>crescendoSpanner</b> .....	136
<b>crescendoText</b> .....	136
<b>\crescHairpin</b> .....	132
<b>\crescTextCresc</b> .....	132
<b>cross</b> .....	41
<b>cross-staff-connect</b> .....	866
<b>\crossStaff</b> .....	359, 846
cruce de pentagramas .....	358
cruzado, acordes de pentagrama .....	359
cruzado, línea de pentagrama .....	358
cruzado, notas de pentagrama .....	359
cruzado, pentagrama .....	358
cruzado, pentagrama, barras de .....	354
cruzado, pentagrama, colisión de voces .....	355
cruzado, pentagrama, corchete de arpeggio .....	360
cruzado, pentagrama, notas de .....	354
cruzado, plicas de pentagrama .....	359
<b>css-&gt;colorlist</b> .....	866
CSS, códigos de color .....	243
cuadrado, corchete, al inicio de un grupo .....	203
cuadrado, ligadura de neuma .....	480
cuartos de tono .....	6
cuartos de tono, alteraciones .....	8
cuartos de tono, tablatura .....	389
<i>cue-notes</i> .....	337
<b>cue-substitute</b> .....	866
<b>\cueClef</b> .....	227, 846
<b>\cueClefUnset</b> .....	227, 846

<b>\cueDuring</b> .....	227, 846
<b>\cueDuringWithClef</b> .....	227, 846
cuerda al aire, indicación de .....	366
cuerda, número .....	366
cuerda, número de la .....	369
cuerda, números de, orientación .....	240
cuerdas orquestales .....	365
cuerdas, escribir música para .....	365
<i>CueVoice</i> .....	232, 338
<i>CueVoice</i> .....	227
cuica .....	802
<b>currentBarNumber</b> .....	112, 126
custodes .....	466
<b>\customTabClef</b> .....	775
<i>custos</i> .....	464, 468
<i>Custos</i> .....	468
custos .....	466
<b>cyclic-base-value</b> .....	866

## D

dórico, modo .....	23
D.S. al Fine .....	119
<b>\dashBang</b> .....	129
<b>\dashDash</b> .....	129
<b>\dashDot</b> .....	129
<b>\dashHat</b> .....	129
<b>\dashLarger</b> .....	129
<b>\dashPlus</b> .....	129
<b>\dashUnderscore</b> .....	129
<b>\deadNote</b> .....	41, 846
<b>\deadNotesOff</b> .....	41
<b>\deadNotesOn</b> .....	41
decorar texto .....	274
<b>\decr</b> .....	131
<b>\decresc</b> .....	132
<i>decrescendo</i> .....	137
<i>decrescendo</i> .....	131
<b>decrescendoSpanner</b> .....	136
<b>decrescendoText</b> .....	136
<b>default-flag</b> .....	867
<b>default-staff-staff-spacing</b> .....	590
<i>Default_bar_line_engraver</i> .....	84
<b>default</b> .....	30, 31
<b>\default</b> .....	118, 526
<b>defaultBarType</b> .....	111
<b>\defaultchild</b> .....	641
<b>\defaultTimeSignature</b> .....	70
<b>define-bar-line</b> .....	867
<b>define-event-class</b> .....	867
<b>define-fonts</b> .....	867
<b>define-tag-group</b> .....	867
<b>\defineBarLine</b> .....	108, 846
<i>Definición de nuevas instrucciones de</i>	
<i>lista de marcado</i> .....	279
definición, traste predefinido .....	406
definir línea de compás .....	108
<b>degrees-&gt;radians</b> .....	867
<b>\deminutum</b> .....	480, 486
<b>\denies</b> .....	638, 639, 640
derecha, mano, digitación, posicionamiento .....	414
derecha, mano, digitaciones para	
instrumentos de traste .....	414
derecho, pedal .....	361

<b>descend-to-context</b> .....	867
<b>\descendens</b> .....	480, 486
deslizamiento en notación de tablatura .....	383
desplazamiento .....	64, 653
desplazamiento, de silencios, automático .....	186
desplazamiento, de voces .....	186
desplazamiento, silencio automático de .....	186
desplazar nota .....	186
desplegar, repetición .....	168
<b>determine-split-list</b> .....	867
<b>determine-string-fret-finger</b> .....	867
diagrama de acordes .....	400
diagrama de acordes para	
instrumentos de trastes .....	390
diagrama de posiciones .....	400
diagrama de posiciones, de mandolina .....	400
diagrama de posiciones, de ukelele .....	400
diagrama de traste, transporte de .....	402
diagrama de trastes .....	390
diagrama de trastes personalizado .....	397
diagrama de trastes personalizados, añadir .....	403
dibujar el símbolo del pentagrama .....	664
dibujar, objeto gráfico .....	274
digitación .....	240
digitación de acordes .....	240
digitación de pulgar .....	128
digitación frente a números de cuerda .....	369
digitación, añadir, a diagramas de trastes .....	412
digitación, de la mano derecha para	
instrumentos de trastes .....	414
digitación, dentro del pentagrama .....	241
digitación, instrucciones de, para acordes .....	240
digitación, orientación .....	240
digitación, posicionamiento, mano derecha .....	414
digitación, pulgar .....	240, 801
digitación, símbolos para	
instrumentos de viento .....	429
digitación, y plicas .....	241
digitación, y silencios multicompás .....	69
<b>dim</b> .....	443
<b>\dim</b> .....	132
dimensión .....	664
<b>\dimHairpin</b> .....	132
diminuendo .....	131
<b>\dimTextDecr</b> .....	132
<b>\dimTextDecresc</b> .....	132
<b>\dimTextDim</b> .....	132
dinámica .....	131
dinámica del MIDI .....	555
dinámica personalizada en MIDI .....	555
dinámica, indicaciones centradas en	
música de tecla .....	353
dinámica, posición vertical .....	133
dinámica, varias indicaciones sobre una nota .....	132
dinámicas indicaciones editoriales .....	138
dinámicas, indicaciones entre paréntesis .....	138
dinámicas, indicaciones, nuevas .....	137
<b>dir-basename</b> .....	868
<b>\dir-column</b> .....	751
dirección automática de la plica	
sobre la línea media .....	246
dirección predeterminada de la plica	
sobre la línea media .....	246
dirección, símbolos .....	97
<b>\discant</b> .....	784
discanto, signo de, del acordeón .....	362
discontinua, ligadura .....	59, 140
discontinuas, ligaduras de fraseo .....	143
discontinuidad, definir patrón de,	
ligadura de expresión .....	142
diseño «mensurstriche» .....	205
diseño de página .....	611
<b>display-lily-music</b> .....	868
<b>display-music</b> .....	868
<b>display-scheme-music</b> .....	868
<b>\displayLilyMusic</b> .....	565, 846
<b>\displayMusic</b> .....	846
<b>\displayScheme</b> .....	846
disponibles, listar las fuentes tipográficas .....	283
disposición, acorde mayor 7 .....	454
disposición, archivo de .....	581
distancia absoluta .....	663
distancia escalada .....	663
distancia, entre pentagramas .....	590
distancia, entre pentagramas en	
música de piano .....	359
dividida, letra .....	315
dividida, voz .....	330
dividido, modificación de un	
objeto de extensión .....	688
dividir en música etiquetada .....	545
divisi, pentagramas .....	218
división, de notas .....	84
división, de silencios .....	84
divisible, glissando .....	149
<i>divisio</i> .....	478
<i>divisio</i> .....	478
<b>\divisioMaior</b> .....	478
<b>\divisioMaxima</b> .....	478
<b>\divisioMinima</b> .....	478
divisiones .....	478
Do, clave .....	734
Do, clave de .....	18
doblada, barra .....	89
doblada, barra, cambiar .....	89
doble bemol .....	6
doble compás .....	81
doble ligadura de expresión para acordes legato ..	141
doble puntillo, nota con .....	50
doble sostenido .....	6
doble, barra .....	104
dodecafónico sin repetir, estilo de alteración .....	36
dodecafónico, estilo de alteraciones .....	35
dodecafónico-primero, estilo de alteración .....	36
<b>dodecaphonic</b> .....	35
<b>dodecaphonic-first</b> .....	36
<b>dodecaphonic-no-repeat</b> .....	36
<b>dodecaphonic-no-repeat-rule</b> .....	868
<i>doit</i> .....	147
<i>doit</i> (elevación de tono) .....	146
<b>\dorian</b> .....	23
<i>DotColumn</i> .....	51
<i>Dots</i> .....	51
<b>\dotsDown</b> .....	50
<b>\dotsNeutral</b> .....	50
<b>\dotsUp</b> .....	50
<i>double flat</i> .....	8
<i>double sharp</i> .....	8
<i>Double_percent_repeat_engraver</i> .....	173
<b>\doubleflat</b> .....	775

<i>DoublePercentEvent</i> .....	172
<i>DoublePercentRepeat</i> .....	173
<i>DoublePercentRepeatCounter</i> .....	173
<i>DoubleRepeatSlash</i> .....	172
<i>\doublesharp</i> .....	775
<i>doubleSlurs</i> .....	141
‘down pralltriller’, ornamento .....	128
‘down-bow’, indicación de arco .....	128
<i>\downbow</i> .....	128, 366, 801
<i>\downmordent</i> .....	128, 800
<i>\downprall</i> .....	128, 800
<i>\draw-circle</i> .....	275, 766
<i>\draw-dashed-line</i> .....	766
<i>\draw-dotted-line</i> .....	767
<i>\draw-hline</i> .....	767
<i>\draw-line</i> .....	275, 768
<i>\draw-squiggle-line</i> .....	768
drawing beam, within text .....	765
drawing box, with rounded corners .....	769
drawing box, with rounded corners, around text .....	773
drawing circle, within text .....	766
drawing dashed line, within text .....	766
drawing dotted line, within text .....	767
drawing ellipse, around text .....	768
drawing line, across a page .....	767
drawing line, within text .....	768
drawing oval, around text .....	770
drawing path .....	771
drawing polygon .....	772
drawing solid box, within text .....	769
drawing squiggled line, within text .....	768
drawing triangle, within text .....	773
<i>\dropNote</i> .....	447, 846
<i>\drummode</i> .....	201, 420, 660
<i>drumPitchNames</i> .....	425
<i>drumPitchTable</i> .....	426
<i>\drums</i> .....	420, 660
<i>DrumStaff</i> .....	202, 422, 427
<i>DrumStaff</i> .....	201
<i>drumStyleTable</i> .....	425
<i>DrumVoice</i> .....	422, 423, 427
duración aislada .....	49
duración predeterminada .....	49
duración, de las notas .....	49
duración, escalado de .....	56
<i>Duration names notes and rests</i> .....	51
<i>duration-dot-factor</i> .....	868
<i>duration-length</i> .....	868
<i>duration-line::calc</i> .....	869
<i>duration-line::print</i> .....	869
<i>duration-log-factor</i> .....	869
<i>duration-visual</i> .....	869
<i>duration-visual-length</i> .....	869
<i>\dwn</i> .....	499
<i>\dynamic</i> .....	138, 741
<i>dynamic-event</i> .....	226
<i>dynamic-text-spanner::before-line-breaking</i> ..	869
<i>Dynamic-performer</i> .....	556, 557, 559
<i>\dynamicDown</i> .....	133
<i>DynamicLineSpanner</i> .....	133, 137
<i>DynamicLineSpanner</i> .....	133, 135
<i>\dynamicNeutral</i> .....	133
<i>Dynamics</i> .....	137
<i>DynamicText</i> .....	137

<i>\dynamicUp</i> .....	133
-------------------------	-----

## E

<i>\easyHeadsOff</i> .....	42
<i>\easyHeadsOn</i> .....	42
ecualización predeterminada, instrumento MIDI, sustitución de la .....	557
<i>Editorial annotations</i> ...	239, 242, 243, 244, 245, 246, 249, 251
editoriales, indicaciones dinámicas .....	138
efectos en MIDI .....	563
<i>Ejemplos reales de música</i> .....	188, 191, 353, 354
<i>Elaborar canciones sencillas</i> .....	286, 287
elevación de tono (doit) .....	146
eliminación de acordes repetidos .....	449
eliminar, número de compás .....	116
eliminar, notas guía .....	231
<i>\ellipse</i> .....	768
<i>ellipse-stencil</i> .....	869
Emmentaler, tipografía .....	713
encabezamiento .....	512
enclosing text in box, with rounded corners ..	773
enclosing text within a box .....	740
encuadernación .....	574
<i>\endcr</i> .....	131
<i>\enddecr</i> .....	131
<i>\endSpanners</i> .....	669, 847
<i>Engravers and Performers</i> .....	622
enmarcar texto .....	274
ensayo, letra de, estilo .....	119
ensayo, letra de, formato .....	119
ensayo, letra de, manual .....	119
ensayo, letras de .....	118
ensayo, marca de, debajo del pentagrama .....	76
ensemble, música de .....	331
eolio, modo .....	23
eonio, modo .....	23
<i>Episema</i> .....	479
<i>Episema engraver</i> .....	479
<i>EpisemaEvent</i> .....	479
<i>\epistemFinis</i> .....	478
‘epistemFinis’, articulación gregoriana .....	478
<i>\epistemInitium</i> .....	478
‘epistemInitium’, articulación gregoriana .....	478
EPS, salida .....	550
<i>\epsfile</i> .....	275, 769
escala, duración .....	56
escalado, del pentagrama .....	212
escribir música en paralelo .....	197
<i>Espaciado óptico</i> .....	607, 608
espaciado horizontal .....	606
espaciado vertical .....	589
espaciado, de la letra .....	306
espaciado, nueva sección de .....	608
espaciado, presentación del .....	618
espacio, dentro de los sistemas .....	590
espacio, en la letra .....	287, 294
espacio, entre pentagramas .....	590
espacios en blanco .....	512
especial, símbolos de arpeggio .....	152
especiales, cabezas de figura .....	41
espressivo .....	128
<i>\espressivo</i> .....	128, 132, 800

‘espressivo’, articulación .....	128, 132, 800
estándar, tamaño de la tipografía (de notación) ..	239
estéreo, balance en MIDI .....	563
estilo de la ligadura .....	140
estilo, de cabeza de nota .....	41
estilo, de la letra de ensayo .....	119
estilo, de silencio .....	62
estilo, matiz de, de texto .....	136
estilo, número de compás .....	114
estilo, repetición doble para primera y segunda vez .....	164
estilo, silencios multicompas .....	234
estilo, voz .....	185
estricta, barra .....	97
estricto, espaciado, notas de adorno y .....	123
<i>Estructura de un elemento de nota</i> .....	800
\etc .....	693
etiqueta .....	541
eval-carefully .....	869
event-chord-notes .....	870
event-chord-pitches .....	870
event-chord-reduce .....	870
event-chord-wrap! .....	870
event-has-articulation? .....	870
\eventChords .....	847
evento, contenedor .....	176
excepción, nombres de acorde .....	453
expand-repeat-chords! .....	870
expand-repeat-notes! .....	870
\expandEmptyMeasures .....	233
expansión de música .....	57
explícita, repetición .....	168
explícita, trinos de nota .....	156
<i>Explicación de las expresiones musicales</i> .....	507
<i>Explicación de los grabadores</i> .....	85
explicitClefVisibility .....	673
explicitKeySignatureVisibility .....	673
expresión en MIDI .....	563
expresión, de marcado .....	264
<i>Expressive marks</i> .. 130, 137, 139, 142, 144, 146, 147, 152, 155, 158, 260, 416	
extensión, matiz, ocultar línea de .....	136
extensión, modificación de un objeto de .....	688
extensora, línea .....	298
extra-offset .....	590
extract-beam-exceptions .....	870
extract-music .....	870
extract-named-music .....	870
extract-typed-music .....	871
\eyeglasses .....	790
Ez_numbers_engraver .....	42

## F

\f	131
fácil, notación, cabezas de nota de	42
fácil, notación, con números	42
Fa, clave	734
Fa, clave de	18
fall	147
fall (caída de tono)	146
falsa, percusión, nota	427
fantasma, nota	245
fantasma, percusión , nota	427

\featherDurations.....	103, 847
\fermata.....	128, 775, 801
fermata o calderón.....	128, 801
Ferneyhough, regulador de.....	135
Feta, tipografía.....	713
\ff.....	131
\fff.....	131
\ffff.....	131
\ffffff.....	131
<i>fifth</i> .....	6
figura, cabeza de la.....	4
<i>figured bass</i> .....	45
<i>FiguredBass</i> .....	220, 460, 461
figuredBassAlterationDirection.....	459
figuredBassPlusDirection.....	459
\figuremode.....	457, 660
\figures.....	457, 660
\fill-line.....	272, 752
\fill-with-pattern.....	538, 752
\filled-box.....	275, 769
final de repetición.....	166
final, barra.....	104
finales alternativos.....	159
finales alternativos y letra.....	311
finalis.....	478
\finalis.....	478
find-named-props.....	871
find-pitch-entry.....	871
\finger.....	240, 741, 847
finger-glide::print.....	871
<i>finger-interface</i> .....	644
<i>Fingering</i> .....	242, 371, 642, 643, 644
fingering chart.....	430
<i>fingering-event</i> .....	242, 643
<i>Fingering_engraver</i> .....	242, 643, 645
<i>FingeringEvent</i> .....	242, 643
fingeringOrientations.....	240
first-assoc.....	871
first-member.....	871
first-page-number.....	577
\first-visible.....	790
\fixed.....	2, 847
<i>flag</i> .....	465, 472
flageolet.....	128
\flageolet.....	128, 429, 801
‘flageolet’, armónicos.....	128
‘flageolet’, armónicos de.....	801
flageolet, cambiar tamaño.....	429
flared-hairpin.....	871
<i>flat</i> .....	8
\flat.....	775
flat-flag.....	872
flatten-list.....	872
Flex.....	806
\flexa.....	486
flip-stencil.....	872
fold-some-music.....	872
followVoice.....	358
font-encoding.....	280
<i>font-interface</i> .....	239, 644, 792
font-interface.....	239, 280
font-name-split.....	873
font-size.....	235, 239
\fontCaps.....	741
fontSize.....	235

<code>\fontsize</code> .....	266, 741
<code>\footnote</code> .....	526, 790, 847
<code>footnote-separator-markup</code> .....	578
<code>Footnote_engraver</code> .....	534
<code>FootnoteEvent</code> .....	534
<code>FootnoteItem</code> .....	534
<code>FootnoteSpanner</code> .....	534
<code>for-some-music</code> .....	873
<code>Forbid_line_break_engraver</code> .....	85
<code>Forbid_line_break_engraver</code> .....	55
<code>forget</code> .....	37
formas, notas con.....	44
formato, de la letra.....	287
formato, de la letra de ensayo.....	119
formato, de los elementos de extensión de texto..	258
<i>Formatting text</i> .....	833, 842
forte, pedal.....	361
<code>four-string-banjo</code> .....	419
<code>\fp</code> .....	131
<code>\fraction</code> .....	790
fragmento.....	227
fragmento de música.....	548
fragmento, citar.....	224
francesa, clave.....	18, 734
francesa, partituras a la.....	215
francesa, pentagrama a la.....	212, 215
fraseo, corchetes de.....	251
fraseo, en la letra.....	295
fraseo, ligadura de.....	140, 143
fraseo, ligadura de, definir patrones de discontinuidad.....	144
fraseo, ligadura de, mitad continua y mitad discontinua.....	143
fraseo, ligadura de, simultáneas.....	143
fraseo, ligadura de, varias.....	143
fraseo, ligaduras de, discontinuas.....	143
fraseo, ligaduras de, punteadas.....	143
fraseo, marca de.....	143
<code>\freeBass</code> .....	784
<code>\frenchChords</code> .....	450
<i>Frenched score</i> .....	332
<i>Frenched staff</i> .....	215, 220
<i>Frenched staves</i> .....	332
<code>fret-&gt;pitch</code> .....	873
<code>\fret-diagram</code> .....	391, 780
fret-diagram, marcado con.....	391
<i>fret-diagram-interface</i> ....	397, 400, 405, 411, 413, 414
<code>fret-diagram-interface</code> .....	397
<code>\fret-diagram-terse</code> .....	393, 781
fret-diagram-terse, marcado con.....	393
<code>\fret-diagram-verbose</code> .....	395, 781
fret-diagram-verbose, marcado con.....	395
<code>fret-parse-terse-definition-string</code> .....	873
<code>FretBoards</code> .....	400
<i>Fretted strings</i> .....	371, 386, 390, 400, 411, 414, 415, 416, 417, 418, 419
frigio, modo.....	23
<code>\fromproperty</code> .....	790
fuelle tipográfica.....	805
fuelle tipográfica (de notación), tamaño.....	235
fuelle tipográfica (de notación), tamaño estándar de.....	239
fuelle tipográfica, buscar, disponibles.....	283
fuelle tipográfica, cambiar.....	266

fuelle tipográfica, cambiar para todo el documento.....	284
fuelle tipográfica, de símbolos en marcado.....	280
fuelle tipográfica, explicación.....	280
fuelle tipográfica, familias de.....	268
fuelle tipográfica, tamaño de.....	266
fuelle, establecer la familia de.....	284
fuelle, notación musical.....	280
fuentes externas.....	280
<i>Funciones de callback</i> .....	679
<i>Funciones de marcado</i> .....	266
<i>Funciones de sustitución intermedias</i> .....	655, 689
<i>Funciones musicales</i> .....	691, 692
<code>function-chain</code> .....	873
fundamental de un acorde.....	443
Funk, cabezas de nota con forma.....	44
<code>\funkHeads</code> .....	44
<code>\funkHeadsMinor</code> .....	45

## G

gaita.....	431
gaita escocesa de highland.....	431
gaita, ejemplo de.....	432
<code>\general-align</code> .....	270, 753
<code>\germanChords</code> .....	450
<code>get-bound-note-heads</code> .....	874
<code>get-chord-shape</code> .....	874
<code>get-postscript-bbox</code> .....	874
<code>get-tweakable-music</code> .....	874
glifo.....	805
glifo, musicales.....	119
<i>Glissando</i> .....	152, 669
<i>glissando</i> .....	151
glissando.....	147
<code>\glissando</code> .....	147
glissando contemporáneo.....	148
glissando, a través de repetición.....	150
glissando, divisible.....	149
glissando, en acordes de tablatura.....	384
glissando, marca de tiempo.....	149
glissando, y repeticiones.....	166
<code>\glissandoMap</code> .....	148
globo.....	248
globo de ayuda.....	248
golpeado, ritmo, mostrar.....	87
gráfica, notación.....	275
gráfico, dibujar objeto.....	274
gráfico, insertar.....	274, 275
gráfico, insertar objeto.....	274
grabador, incluir en contextos.....	638
<code>\grace</code> .....	120, 847
<i>grace notes</i> .....	124
<i>Grace_auto_beam_engraver</i> .....	124
<i>Grace_beam_engraver</i> .....	124
<i>Grace_engraver</i> .....	124
<i>Grace.spacing_engraver</i> .....	124
<i>GraceMusic</i> .....	124
gramática, de LilyPond.....	807
<i>grand staff</i> .....	205
<i>GrandStaff</i> .....	37, 205
<i>Graphical Object Interfaces</i> .....	806
gregoriano, alteración.....	477
gregoriano, armadura.....	477
gregoriano, articulación.....	478

gregoriano, articulación, <i>accentus</i> .....	478, 802
gregoriano, articulación, <i>circulus</i> .....	478, 802
gregoriano, articulación, <i>episemFinis</i> .....	478
gregoriano, articulación, <i>episemInitium</i> .....	478
gregoriano, articulación, <i>ictus</i> .....	478, 802
gregoriano, articulación, <i>semicirculus</i> .....	478, 802
gregoriano, canto, pauta de transcripción de .....	201
gregoriano, ligadura de neumas .....	480
gregoriano, transcripción moderna .....	345
<i>GregorianTranscriptionStaff</i> .....	202
<b>GregorianTranscriptionStaff</b> .....	201
<i>grid-line-interface</i> .....	251
<i>grid-point-interface</i> .....	251
<i>Grid_line_span_engraver</i> .....	251
<b>Grid_line_span_engraver</b> .....	249
<i>Grid_point_engraver</i> .....	251
<b>Grid_point_engraver</b> .....	249
<i>gridInterval</i> .....	249
<i>GridLine</i> .....	251
<i>GridPoint</i> .....	251
<i>grob</i> .....	643, 805
<i>grob</i> , propiedad de .....	648
<i>grob</i> , sobreescritura de .....	671
<i>grob</i> , visibilidad de .....	670
<i>grob-elts::X-extent</i> .....	875
<i>grob-interface</i> .....	643, 644, 806, 808
<i>grob-transformer</i> .....	877
<i>grob::all-objects</i> .....	874
<i>grob::compose-function</i> .....	874
<i>grob::display-objects</i> .....	874
<i>grob::name</i> .....	874
<i>grob::offset-function</i> .....	875
<i>grob::rhythmic-location</i> .....	875
<i>grob::unpure-Y-extent-from-stencil</i> .....	875
<i>grob::when</i> .....	875
<i>\grobdescriptions</i> .....	847
<i>grow-direction</i> .....	103
grupeto .....	128
‘grupeto circular barrado’, ornamento .....	128, 800
‘grupeto circular de Haydn’, ornamento .....	800
‘grupeto circular invertido’, ornamento .....	128, 800
‘grupeto circular’, ornamento .....	128, 800
‘grupeto de Haydn’, ornamento .....	128
grupeto de retardación, ornamento .....	130
grupo de pentagramas, corchete	
cuadrado al inicio .....	203
grupo de valoración especial, barra, cortar línea ...	55
grupo de valoración especial,	
colocación del corchete .....	52
grupo de valoración especial,	
corchete, visibilidad .....	54
grupo de valoración especial,	
modificación del número .....	53
grupo de valoración especial, número,	
no predeterminado .....	54
grupo especial .....	51
grupo especial, corchete, visibilidad .....	53
grupo especial, formato de .....	53
grupo especial, introducir muchos .....	53
grupo especial, ligadura .....	52
grupo especial, número, visibilidad .....	53
grupos de etiquetas .....	544
guía, en partitura vocal .....	336
guía, formateo de las notas .....	227
guía, musicales .....	334

guía, nota .....	227
guía, nota, eliminar .....	231
guía, nota, formato de .....	227
guía, notas, clave con .....	18
guía, voces .....	224
guión .....	128
guion .....	298
guiro .....	802
guitarra, cabezas de nota .....	41
guitarra, mostrar ritmos rasgueados .....	86
guitarra, ritmo golpeado, mostrar .....	87
guitarra, tablas de acordes .....	86
guitarra, tablatura de .....	369

## H

<i>hairpin</i> .....	137
<i>Hairpin</i> .....	137, 653, 654
Hal Leonard .....	42
half-open high hat .....	425
<i>\halfopen</i> .....	128, 425, 801
<i>\halign</i> .....	269, 754
hammer on .....	385
Harmonia Sacra, cabezas de nota con forma .....	44
<i>\harmonic</i> .....	41, 366, 376
<i>\harmonicByFret</i> .....	376, 847
<i>\harmonicByRatio</i> .....	376, 847
<i>\harmonicNote</i> .....	847
<i>harmonics</i> .....	367
<i>\harmonicsOff</i> .....	366
<i>\harmonicsOn</i> .....	366, 847
<i>\harp-pedal</i> .....	782
<i>\haydnturn</i> .....	128, 800
<i>\hbracket</i> .....	274, 769
<i>\hcenter-in</i> .....	755
<i>\header</i> .....	510
Henze, calderón de .....	128, 801
<i>\henzelongfermata</i> .....	128, 801
<i>\henzeshortfermata</i> .....	128, 801
<i>\hide</i> .....	670, 847
<i>\hideKeySignature</i> .....	431
<i>\hideNotes</i> .....	242
<i>\hideSplitTiedTabNotes</i> .....	375
<i>\hideStaffSwitch</i> .....	358
high hat .....	802
high hat, abierto .....	128, 802
high hat, de pedal .....	802
high hat, half open .....	425
high hat, pedal .....	128, 425
high hat, semi abierto .....	128
high hat, semiabierto .....	802
himno .....	339, 345
himno, y compás parcial .....	350
hoja guía de acordes .....	449
<i>Hojas de estilo</i> .....	546
<i>hook-stencil</i> .....	877
horizontal, alineación de texto .....	269
horizontal, alineación, letra .....	307
horizontal, barra .....	378
horizontal, corchete .....	251
horizontal, desplazamiento de notas .....	190
horizontal, espaciado .....	606
horizontal, relleno en marcado .....	270
horizontal, sobreescribir el espaciado .....	690



<i>horizontal-bracket-interface</i> .....	253
<i>horizontal-bracket-text-interface</i> .....	253
<i>horizontal-shift</i> .....	575
<i>Horizontal_bracket_engraver</i> .....	253
<i>Horizontal_bracket_engraver</i> .....	251
<i>HorizontalBracket</i> .....	253
<i>HorizontalBracketText</i> .....	253
<i>HorizontalBracketText</i> .....	253
horizontally centering text .....	750
<i>\hspace</i> .....	270, 755
hufnagel .....	464, 465
<i>\huge</i> .....	235, 268, 741

## I

<i>\ictus</i> .....	478, 802
'ictus', articulación gregoriana .....	478
'ictus', articulación, gregoriana .....	802
idioma, alturas en otros .....	8
idioma, nombres de nota en otros .....	8
<i>\iij</i> .....	481
<i>\IIJ</i> .....	481
<i>\ij</i> .....	481
<i>\IJ</i> .....	481
imagen, insertar .....	275
implícito, contexto .....	641
importing stencil, into text .....	794
impresión de los nombres de acorde .....	448
impresión, de caracteres especiales .....	265
impresión, orden de .....	671
improvisación .....	47
improvisación, árabe .....	501
<i>\improvisationOff</i> .....	47, 86
<i>\improvisationOn</i> .....	47, 86
<i>\in</i> .....	663
<i>\incipit</i> .....	490, 848
incipit, añadir .....	490
inclinación, modificación, extensor de octava .....	27
<i>\inclinatum</i> .....	480, 486
<i>\include</i> .....	511, 538
include settings .....	545
including files .....	538
<i>indent</i> .....	222, 575, 611
independiente, texto .....	262
Indicación de compás .....	70
indicación de pedal .....	128
indicación de pedalero de órgano .....	128
indicación de pulgar .....	128
indicación, de texto .....	260
indicaciones dinámicas nuevas .....	137
indicar No Chord (sin acorde) en <i>ChordNames</i> .....	448
<i>\inherit-acceptability</i> .....	639, 848
inlining an Encapsulated PostScript image .....	769
<i>inner-margin</i> .....	574
inscripción, ordenación vertical .....	129
inscripción, sobre un silencio multicompa .....	66
inscripciones textuales .....	800
insertar, gráfico .....	275
insertar, objeto gráfico .....	274
inserting music, into text .....	778
inserting PostScript directly, into text .....	772
inserting URL link, into text .....	774
<i>\inStaffSegno</i> .....	161, 848
<i>instrument-specific-markup-interface</i> .....	441, 792

<i>InstrumentName</i> .....	224
instrumento MIDI, sustituir la ecualización predeterminada .....	557
instrumento, nombre abreviado .....	220
instrumento, nombre del .....	220
instrumento, nombre, centrar .....	221
instrumento, nombre, complejo .....	221
instrumento, transpositor .....	12
instrumentos transpositores .....	12
instrumentos, nombre .....	564
<i>\instrumentSwitch</i> .....	848
<i>Interfaces para programadores</i> .....	678
interfaz .....	806
interfaz de la presentación .....	643
interfaz, de grob .....	806
interfaz, de objeto gráfico .....	806
<i>interval</i> .....	6
<i>interval-center</i> .....	878
<i>interval-index</i> .....	878
<i>interval-length</i> .....	878
introducir, la letra .....	287
<i>invalidate-alterations</i> .....	878
inversión .....	14
inversión modal .....	17
<i>\inversion</i> .....	14, 848
<i>\invertChords</i> .....	447, 848
invisible, nota .....	242
invisible, plica .....	245
invisible, silencio .....	64
<i>\ionian</i> .....	23
<i>iraq</i> .....	501
<i>\italianChords</i> .....	450
<i>\italic</i> .....	266, 742
<i>item-interface</i> .....	644

## J

jazz, acorde de .....	450
jazz, acordes de .....	442
justificado, texto .....	273
<i>\justified-lines</i> .....	279, 796
<i>\justify</i> .....	273, 757
<i>\justify-field</i> .....	756
<i>\justify-line</i> .....	756
<i>\justify-string</i> .....	757
justifying lines of text .....	796
justifying text .....	757

## K

<i>Keep_alive_together_engraver</i> .....	220
<i>keepAliveInterfaces</i> .....	216
<i>\keepWithTag</i> .....	541, 848
<i>\key</i> .....	23, 45, 848
<i>key signature</i> .....	473, 478
key signature .....	6
<i>key-signature-interface</i> .....	25
<i>Key_engraver</i> .....	25
<i>Key_performer</i> .....	25
<i>Keyboards</i> .....	354, 356, 358, 359, 361, 362, 363
<i>KeyCancellation</i> .....	25
<i>KeyChangeEvent</i> .....	25
<i>KeySignature</i> .....	25, 473, 477, 478, 501
<i>kievan notation</i> .....	487, 488, 489

kievana, clave .....	469, 734
kievano, alteraciones de .....	488
kievano, ligaduras de .....	489
<code>\kieveanOff</code> .....	488
<code>\kieveanOn</code> .....	488
<code>KieveanStaff</code> .....	486
<code>KieveanVoice</code> .....	486
<code>\killCues</code> .....	231, 848
kirchenpause .....	234
kurd .....	501

## L

línea de cambio de pentagrama .....	358
línea de extensión .....	298
línea de rejilla .....	249
línea de seguimiento de pentagrama .....	358
línea divisoria entre pentagramas .....	108
línea divisoria predeterminada, cambiar .....	111
línea divisoria, detener regulador .....	133
línea divisoria, eliminación .....	675
línea vertical entre pentagramas .....	249
línea, adicional .....	208
línea, barras en saltos de .....	89
línea, comentario de .....	507
línea, de compás, entre pautas .....	205
línea, del pentagrama .....	208
línea, divisoria, evitada por la letra lyrics .....	307
línea, divisoria, invisible .....	104
línea, divisoria, símbolos sobre las .....	260
línea, divisorias .....	104
línea, final, posicionar marcas .....	261
línea, salto .....	104
línea, salto de .....	582
línea, salto, barra .....	89
línea, salto, gestión con una voz adicional .....	585
líneas divisorias manuales .....	105
<i>La partitura es una (única) expresión</i>	
<i>musical compuesta</i> .....	507
<i>La propiedad outside-staff-priority (prioridad</i>	
<i>fuera del pentagrama)</i> .....	605
laúd, afinación del .....	420
laúd, tablaturas de .....	419
<code>\label</code> .....	535, 848
<i>laissez vibrer</i> .....	61
<i>laissez vibrer</i> .....	59
<code>\laissezVibrer</code> .....	59
<code>LaissezVibrerTie</code> .....	62
<code>LaissezVibrerTieColumn</code> .....	62
<code>\language</code> .....	848
<code>\languageRestore</code> .....	848
<code>\languageSaveAndChange</code> .....	848
<code>\large</code> .....	235, 268, 742
<code>\larger</code> .....	266, 268, 742
largo, calderón .....	128, 801
<i>Las voces contienen música</i> .....	185, 191
last-bottom-spacing .....	572
<code>\layout</code> .....	510, 579, 622, 633
layout-line-thickness .....	878
layout-set-absolute-staff-size .....	879
layout-set-staff-size .....	879
layout-set-staff-size .....	581
lead sheet .....	449
ledger line .....	211
ledger-line-spanner-interface .....	42

<i>Ledger_line_engraver</i> .....	42
<i>LedgerLineSpanner</i> .....	42
<code>\left-align</code> .....	269, 758
left-aligning text .....	758
<code>\left-brace</code> .....	791
<code>\left-column</code> .....	758
left-margin .....	573
letra .....	287
letra de ensayo, estilo .....	119
letra de ensayo, formato .....	119
letra dividida .....	315
letra, alineación a una melodía .....	288
letra, alineación con melodía esporádica .....	629
letra, alineación horizontal .....	307
letra, asignada a una voz .....	181
letra, aumentar la separación .....	306
letra, compartida entre voces .....	317
letra, desplazamiento de .....	64
letra, espaciado versión 2.12 .....	303
letra, evitar líneas divisorias .....	307
letra, formateado .....	287
letra, introducir .....	287
letra, mantener dentro del margen .....	257, 307
letra, posicionamiento .....	301
letra, puntuación en la .....	287
letra, repetición .....	307
letra, repetición con finales alternativos .....	311
letra, repetición con primera y segunda vez .....	311
letra, saltar notas .....	312
letra, sobre notas de adorno .....	322
letra, y barras .....	91
letra, y marcado .....	288
letra, y melodías .....	290
letra, y notas ligadas .....	313
letras de ensayo .....	118
lexer .....	806
<code>\lheel</code> .....	128, 801
lidio, modo .....	23
ligadura .....	466
ligadura continua .....	140
ligadura de expresión .....	140
ligadura de expresión y repeticiones .....	166
ligadura de expresión, debajo de las notas .....	140
ligadura de expresión, definición del patrón	
de discontinuidad .....	142
ligadura de expresión, definir patrones	
de discontinuidad .....	141
ligadura de expresión, doble, para	
acordes legato .....	141
ligadura de expresión, encima de las notas .....	140
ligadura de expresión, estilo .....	140
ligadura de expresión, marcado de	
texto por dentro .....	141
ligadura de expresión, mitad continua y	
mitad discontinua .....	140
ligadura de expresión, modificar .....	684
ligadura de expresión, posicionamiento manual .....	140
ligadura de expresión, simultáneas .....	140
ligadura de expresión, varias .....	140
ligadura de fraseo .....	140, 143
ligadura de fraseo discontinua .....	143
ligadura de fraseo mitad continua y	
mitad discontinua .....	143
ligadura de fraseo punteada .....	143

ligadura de fraseo, definir patrones		
de discontinuidad	144	
ligadura de fraseo, simultáneas	143	
ligadura de grupo especial	52	
ligadura de puntos	140	
ligadura de unión	58	
ligadura de unión, corchetes de		
primera y segunda vez	59	
ligadura de unión, modificar	684	
ligadura de unión, repetición y	59	
ligadura discontinua	140	
ligadura laissez vibrer	59	
ligadura mensural	474	
ligadura mensural blanca	474	
ligadura, alteraciones y	7	
ligadura, apariencia	59	
ligadura, colocación	59	
ligadura, con arpeggio	61	
ligadura, de puntos	59	
ligadura, discontinua	59	
ligadura, edición manual	61	
ligadura, en la letra	294	
ligadura, en primera y segunda vez	161	
ligadura, en repeticiones	161	
ligadura, neumas cuadrados	480	
ligadura, y acordes	59	
ligaduras de fraseo, varias	143	
ligaduras kievanas	489	
<i>ligature</i>	464, 466, 475, 476, 486, 490	
ligature, in text	751	
<i>LilyPond grammar</i>	807	
<i>lilypond-main</i>	879	
<i>line</i>	211	
<i>\line</i>	759	
<i>line-spanner-interface</i>	669	
<i>line-width</i>	272, 573, 611	
<i>\linea</i>	480, 486	
<i>LineBreakEvent</i>	586	
<i>\lineprall</i>	128, 800	
list of keys in woodwind diagrams	441	
list of woodwind diagrams	437	
<i>list-insert-separator</i>	879	
<i>list-join</i>	879	
lista de asociación	805	
lista de colores	710	
<i>Lista de referencias bibliográficas</i>	450, 455	
lista-A	805	
listar, fuentes tipográficas disponibles	283	
llave vertical	202	
llave, al comienzo de un pentagrama único	204	
llave, de distintos tamaños	280	
llaves, anidado de	205	
<i>\locrian</i>	23	
locrio, modo	23	
<i>longa</i>	51, 64	
<i>\longa</i>	49, 62	
longa, silencio de	62	
<i>\longfermata</i>	128, 801	
longitud mínima, regulador	134	
<i>Longitud y grosor de los objetos</i>	215, 663	
longitud, de las notas	49	
longitud, de un silencio multicompa	67	
<i>\lookup</i>	791	
<i>lookup-markup-command</i>	879	
<i>\lower</i>	270, 759	
lowering text	759	
<i>\ltoe</i>	128, 801	
<i>ly:add-context-mod</i>	859	
<i>ly:add-interface</i>	859	
<i>ly:add-listener</i>	859	
<i>ly:add-option</i>	859	
<i>ly:all-grob-interfaces</i>	860	
<i>ly:all-options</i>	860	
<i>ly:all-output-backend-commands</i>	860	
<i>ly:all-stencil-commands</i>	860	
<i>ly:all-stencil-expressions</i>	860	
<i>ly:angle</i>	860	
<i>ly:assoc-get</i>	860	
<i>ly:axis-group-interface::add-element</i>	861	
<i>ly:bar-line::calc-anchor</i>	861	
<i>ly:bar-line::print</i>	861	
<i>ly:basic-progress</i>	861	
<i>ly:book-add-bookpart!</i>	862	
<i>ly:book-add-score!</i>	862	
<i>ly:book-book-parts</i>	862	
<i>ly:book-header</i>	862	
<i>ly:book-paper</i>	862	
<i>ly:book-process</i>	862	
<i>ly:book-process-to-systems</i>	862	
<i>ly:book-scores</i>	862	
<i>ly:book-set-header!</i>	862	
<i>ly:book?</i>	862	
<i>ly:bp</i>	863	
<i>ly:bracket</i>	863	
<i>ly:broadcast</i>	863	
<i>ly:camel-case-&gt;lisp-identifier</i>	863	
<i>ly:chain-assoc-get</i>	863	
<i>ly:check-expected-warnings</i>	863	
<i>ly:cm</i>	864	
<i>ly:command-line-code</i>	864	
<i>ly:command-line-options</i>	864	
<i>ly:connect-dispatchers</i>	864	
<i>ly:context-current-moment</i>	865	
<i>ly:context-def-lookup</i>	865	
<i>ly:context-def-modify</i>	865	
<i>ly:context-def?</i>	865	
<i>ly:context-event-source</i>	865	
<i>ly:context-events-below</i>	865	
<i>ly:context-find</i>	865	
<i>ly:context-grob-definition</i>	865	
<i>ly:context-id</i>	865	
<i>ly:context-matched-pop-property</i>	865	
<i>ly:context-mod-apply!</i>	866	
<i>ly:context-mod?</i>	865	
<i>ly:context-name</i>	866	
<i>ly:context-now</i>	866	
<i>ly:context-parent</i>	866	
<i>ly:context-property</i>	866	
<i>ly:context-property-where-defined</i>	866	
<i>ly:context-pushpop-property</i>	866	
<i>ly:context-set-property!</i>	866	
<i>ly:context-unset-property</i>	866	
<i>ly:context?</i>	865	
<i>ly:debug</i>	867	
<i>ly:default-scale</i>	867	
<i>ly:dimension?</i>	867	
<i>ly:dir?</i>	868	
<i>ly:directed</i>	868	
<i>ly:disconnect-dispatchers</i>	868	
<i>ly:dispatcher?</i>	868	

ly:duration->string	868	ly:grob-property-data	876
ly:duration-dot-count	868	ly:grob-pure-height	876
ly:duration-factor	868	ly:grob-pure-property	876
ly:duration-length	868	ly:grob-relative-coordinate	876
ly:duration-log	869	ly:grob-robust-relative-extent	876
ly:duration-scale	869	ly:grob-script-priority-less	876
ly:duration<?	868	ly:grob-set-nested-property!	876
ly:duration?	868	ly:grob-set-object!	876
ly:effective-prefix	869	ly:grob-set-parent!	877
ly:encode-string-for-pdf	869	ly:grob-set-property!	877
ly:engraver-announce-end-grob	869	ly:grob-spanned-rank-interval	877
ly:engraver-make-grob	869	ly:grob-staff-position	877
ly:error	869	ly:grob-suicide!	877
ly:event-deep-copy	870	ly:grob-system	877
ly:event-property	870	ly:grob-translate-axis!	877
ly:event-set-property!	870	ly:grob-vertical<?	877
ly:event?	870	ly:grob?	874
ly:expect-warning	870	ly:gulp-file	877
ly:extract-subfont-from-collection	871	ly:gulp-file-utf8	877
ly:find-file	871	ly:has-glyph-names?	877
ly:font-config-add-directory	872	ly:hash-table-keys	877
ly:font-config-add-font	872	ly:in-event-class?	877
ly:font-config-display-fonts	872	ly:inch	877
ly:font-config-get-font-file	872	ly:input-both-locations	878
ly:font-design-size	872	ly:input-file-line-char-column	878
ly:font-file-name	872	ly:input-location?	878
ly:font-get-glyph	872	ly:input-message	878
ly:font-glyph-name-to-charcode	872	ly:input-warning	878
ly:font-glyph-name-to-index	872	ly:interpret-music-expression	878
ly:font-index-to-charcode	873	ly:intlog2	878
ly:font-magnification	873	ly:item-break-dir	878
ly:font-metric?	873	ly:item-get-column	878
ly:font-name	873	ly:item?	878
ly:font-sub-fonts	873	ly:iterator?	878
ly:format	873	ly:length	879
ly:format-output	873	ly:lily-lexer?	879
ly:generic-bound-extent	873	ly:lily-parser?	879
ly:get-all-function-documentation	873	ly:line-interface::line	879
ly:get-all-translators	873	ly:listened-event-class?	879
ly:get-cff-offset	874	ly:listened-event-types	879
ly:get-context-mods	874	ly:listener?	879
ly:get-font-format	874	ly:make-book	879
ly:get-option	874	ly:make-book-part	879
ly:get-spacing-spec	874	ly:make-context-mod	880
ly:gettext	874	ly:make-dispatcher	880
ly:grob-alist-chain	875	ly:make-duration	880
ly:grob-array->list	875	ly:make-global-context	880
ly:grob-array-length	875	ly:make-global-translator	881
ly:grob-array-ref	875	ly:make-grob-properties	881
ly:grob-array?	875	ly:make-listener	881
ly:grob-basic-properties	875	ly:make-moment	881
ly:grob-chain-callback	875	ly:make-music	881
ly:grob-common-refpoint	875	ly:make-music-function	882
ly:grob-common-refpoint-of-array	875	ly:make-music-relative!	882
ly:grob-default-font	875	ly:make-output-def	882
ly:grob-extent	875	ly:make-page-label-marker	882
ly:grob-get-vertical-axis-group-index	876	ly:make-page-permission-marker	882
ly:grob-interfaces	876	ly:make-pango-description-string	882
ly:grob-layout	876	ly:make-paper-outputter	882
ly:grob-object	876	ly:make-pitch	883
ly:grob-original	876	ly:make-prob	883
ly:grob-parent	876	ly:make-rotation	883
ly:grob-pq<?	876	ly:make-scale	883
ly:grob-properties?	876	ly:make-scaling	883
ly:grob-property	876	ly:make-score	883

ly:make-spring.....	883	ly:output-def-set-variable!.....	889
ly:make-stencil.....	883	ly:output-def?.....	889
ly:make-stream-event.....	884	ly:output-description.....	889
ly:make-transform.....	884	ly:output-find-context-def.....	889
ly:make-translation.....	884	ly:output-formats.....	890
ly:make-unpure-pure-container.....	884	ly:outputter-close.....	890
ly:message.....	885	ly:outputter-dump-stencil.....	890
ly:minimal-breaking.....	885	ly:outputter-dump-string.....	890
ly:minimal-breaking.....	587	ly:outputter-output-scheme.....	890
ly:mm.....	885	ly:outputter-port.....	890
ly:module->alist.....	885	ly:page-marker?.....	890
ly:module-copy.....	885	ly:page-turn-breaking.....	890
ly:modules-lookup.....	885	ly:page-turn-breaking.....	588
ly:moment-add.....	885	ly:pango-font-physical-fonts.....	890
ly:moment-div.....	885	ly:pango-font?.....	890
ly:moment-grace.....	885	ly:paper-book-header.....	891
ly:moment-grace-denominator.....	885	ly:paper-book-pages.....	891
ly:moment-grace-numerator.....	886	ly:paper-book-paper.....	891
ly:moment-main.....	886	ly:paper-book-performances.....	891
ly:moment-main-denominator.....	886	ly:paper-book-scopes.....	891
ly:moment-main-numerator.....	886	ly:paper-book-systems.....	891
ly:moment-mod.....	886	ly:paper-book?.....	890
ly:moment-mul.....	886	ly:paper-column::break-align-width.....	891
ly:moment-sub.....	886	ly:paper-column::print.....	891
ly:moment<?.....	885	ly:paper-fonts.....	891
ly:moment?.....	885	ly:paper-get-font.....	891
ly:music-compress.....	886	ly:paper-get-number.....	891
ly:music-deep-copy.....	886	ly:paper-outputscales.....	891
ly:music-duration-compress.....	886	ly:paper-score-paper-systems.....	891
ly:music-duration-length.....	886	ly:paper-system-minimum-distance.....	891
ly:music-function-extract.....	886	ly:paper-system?.....	891
ly:music-function-signature.....	887	ly:parse-file.....	891
ly:music-function?.....	886	ly:parse-init.....	892
ly:music-length.....	887	ly:parse-string-expression.....	892
ly:music-list?.....	887	ly:parsed-undead-list!.....	892
ly:music-mutable-properties.....	887	ly:parser-clear-error.....	892
ly:music-output?.....	887	ly:parser-clone.....	892
ly:music-property.....	887	ly:parser-define!.....	892
ly:music-set-property!.....	887	ly:parser-error.....	892
ly:music-start.....	887	ly:parser-has-error?.....	892
ly:music-transpose.....	887	ly:parser-include-string.....	892
ly:music?.....	886	ly:parser-lookup.....	892
ly:note-column-accidentals.....	888	ly:parser-output-name.....	892
ly:note-column-dot-column.....	888	ly:parser-parse-string.....	892
ly:note-head::stem-attachment.....	888	ly:parser-set-note-names.....	892
ly:number->string.....	888	ly:performance-headers.....	892
ly:one-line-auto-height-breaking.....	888	ly:performance-write.....	893
ly:one-line-auto-height-breaking.....	588	ly:pitch-alteration.....	893
ly:one-line-breaking.....	888	ly:pitch-diff.....	893
ly:one-line-breaking.....	588	ly:pitch-negate.....	893
ly:one-page-breaking.....	889	ly:pitch-notename.....	893
ly:one-page-breaking.....	588	ly:pitch-octave.....	893
ly:optimal-breaking.....	889	ly:pitch-quartertones.....	893
ly:optimal-breaking.....	587	ly:pitch-semitones.....	893
ly:option-usage.....	889	ly:pitch-steps.....	893
ly:otf->cff.....	889	ly:pitch-tones.....	893
ly:otf-font-glyph-info.....	889	ly:pitch-transpose.....	893
ly:otf-font-table-data.....	889	ly:pitch<?.....	893
ly:otf-font?.....	889	ly:pitch?.....	893
ly:otf-glyph-count.....	889	ly:pointer-group-interface::add-grob.....	893
ly:otf-glyph-list.....	889	ly:position-on-line?.....	893
ly:output-def-clone.....	889	ly:prob-immutable-properties.....	893
ly:output-def-lookup.....	889	ly:prob-mutable-properties.....	894
ly:output-def-parent.....	889	ly:prob-property.....	894
ly:output-def-scope.....	889	ly:prob-property?.....	894

ly:prob-set-property!	894	ly:stencil-rotate	901
ly:prob-type?	894	ly:stencil-rotate-absolute	901
ly:prob?	893	ly:stencil-scale	901
ly:programming-error	894	ly:stencil-stack	901
ly:progress	894	ly:stencil-translate	901
ly:property-lookup-stats	894	ly:stencil-translate-axis	901
ly:protects	894	ly:stencil?	900
ly:pt	894	ly:stream-event?	902
ly:pure-call	894	ly:string-percent-encode	902
ly:randomize-rand-seed	894	ly:string-substitute	902
ly:register-stencil-expression	895	ly:system-font-load	902
ly:register-translator	895	ly:text-interface::interpret-markup	902
ly:relative-group-extent	895	ly:time-signature::print	903
ly:rename-file	895	ly:transform->list	903
ly:reset-all-fonts	895	ly:transform?	903
ly:round-filled-box	895	ly:translate-cpp-warning-scheme	903
ly:round-polygon	896	ly:translator-context	903
ly:run-translator	896	ly:translator-description	903
ly:score-add-output-def!	896	ly:translator-group?	903
ly:score-embedded-format	896	ly:translator-name	903
ly:score-error?	896	ly:translator?	903
ly:score-header	896	ly:transpose-key-alist	903
ly:score-music	896	ly:ttf->pfa	903
ly:score-output-defs	897	ly:ttf-ps-name	903
ly:score-set-header!	897	ly:type1->pfa	903
ly:score?	896	ly:unit	903
ly:separation-item::print	897	ly:unpure-call	904
ly:set-default-scale	897	ly:unpure-pure-container-pure-part	904
ly:set-grob-creation-callback	897	ly:unpure-pure-container-unpure-part	904
ly:set-grob-modification-callback	897	ly:unpure-pure-container?	904
ly:set-middle-C!	898	ly:usage	904
ly:set-option	898	ly:verbose-output?	904
ly:set-origin!	898	ly:version	904
ly:set-property-cache-callback	898	ly:version?	904
ly:skyline-empty?	898	ly:volta-bracket::calc-shorten-pair	904
ly:skyline-pair?	898	ly:warning	904
ly:skyline?	898	ly:warning-located	904
ly:smob-protects	898	ly:wide-char->utf-8	904
ly:solve-spring-rod-problem	899	\lydian	23
ly:source-file?	899	lyric-text::print	879
ly:source-files	899	<i>LyricCombineMusic</i>	295, 301
ly:span-bar::before-line-breaking	899	<i>LyricExtender</i>	299
ly:span-bar::calc-glyph-name	899	<i>LyricHyphen</i>	299
ly:span-bar::print	899	\lyricmode	287, 288, 661
ly:span-bar::width	899	<i>Lyrics</i>	220, 290, 294, 301, 329, 814, 815
ly:spanner-bound	899	\lyrics	661
ly:spanner-broken-into	899	lyrics, posicionamiento	214
ly:spanner-set-bound!	899	lyrics, using variables	299
ly:spanner?	899	\lyricsto	288, 290, 291
ly:spawn	899	<i>LyricText</i>	288, 326, 339
ly:spring-set-inverse-compress-strength!	900		
ly:spring-set-inverse-stretch-strength!	900		
ly:spring?	900		
ly:staff-symbol-line-thickness	900		
ly:staff-symbol-staff-radius	900		
ly:staff-symbol-staff-space	900		
ly:stderr-redirect	900		
ly:stencil-add	900		
ly:stencil-aligned-to	900		
ly:stencil-combine-at-edge	900		
ly:stencil-empty?	900		
ly:stencil-expr	900		
ly:stencil-extent	901		
ly:stencil-in-color	901		
ly:stencil-outline	901		

## M

m	443
música antigua, claves	469, 734
música de cuatro compases por línea	584
música intercalada	197
música para principiantes	42
música religiosa	339
música sin medida	126
música sin metro	79
música sin metro, alteraciones	79
música sin metro, barras	79
música sin metro, barras de compás	79

música sin metro, números de compás .....	79	<b>map-selected-alist-keys</b> .....	884
música, dentro de marcado .....	276	<b>map-some-music</b> .....	884
música, fragmento .....	548	<i>maqam</i> .....	501
<i>Métodos de trucaje</i> .....	56, 650, 652	<i>maqam</i> .....	497
<b>magnification-&gt;font-size</b> .....	235, 581	maracas .....	802
<b>\magnify</b> .....	266, 742	marca de ensayo, estilo .....	119
magnifying text .....	742	marca de ensayo, formato .....	119
<b>\magnifyMusic</b> .....	235, 848	marca de ensayo, manual .....	119
<b>\magnifyStaff</b> .....	581, 848	marca de fraseo .....	143
<b>magstep</b> .....	235, 581, 663	marca de pulso .....	145
<b>maj</b> .....	443	marca, al final de la línea .....	261
<b>\major</b> .....	23	marca, de ensayo, debajo del pentagrama .....	76
<b>majorSevenSymbol</b> .....	451, 454	marca, en todos los pentagramas .....	262
<i>makam</i> .....	503	marcado .....	264
<i>makam</i> .....	497, 503	marcado condicional .....	523
<i>makam</i> , ejemplo .....	504	marcado de texto .....	264
<i>makam</i> , turco, ejemplo .....	504	marcado, caracteres especiales en modo de .....	265
<i>makamlar</i> .....	497, 503	marcado, centrar en la página .....	272
<i>makamlar</i> .....	497, 503	marcado, de varias líneas .....	271
<b>make-bow-stencil</b> .....	879	marcado, de varias páginas .....	279
<b>make-c-time-signature-markup</b> .....	880	marcado, dos columnas .....	263
<b>make-circle-stencil</b> .....	880	marcado, elementos, alinear .....	269
<b>make-clef-set</b> .....	880	marcado, en letra .....	288
<b>make-connected-line</b> .....	880	marcado, expresión de .....	264
<b>make-connected-path-stencil</b> .....	880	marcado, notación musical dentro de .....	276
<b>make-cue-clef-set</b> .....	880	marcado, objeto .....	254
<b>make-cue-clef-unset</b> .....	880	marcado, partitura dentro de .....	278
<b>make-duration-of-length</b> .....	880	marcado, relleno de textos de .....	274
<b>make-dynamic-script</b> .....	138	marcado, relleno horizontal .....	270
<b>make-ellipse-stencil</b> .....	880	marcado, silencio multicompa .....	69
<b>make-filled-box-stencil</b> .....	880	marcado, sintaxis .....	264
<b>make-glyph-time-signature-markup</b> .....	881	marcado, sobre silencios multicompa .....	66
<b>make-grob-property-override</b> .....	881	marcado, texto de, varias páginas .....	279
<b>make-grob-property-revert</b> .....	881	marcado, texto, ajustado a palabras .....	273
<b>make-grob-property-set</b> .....	881	marcado, texto, alinear .....	269
<b>make-harmonic</b> .....	881	marcado, texto, decorar .....	274
<b>make-line-stencil</b> .....	881	marcado, texto, dentro de las ligaduras .....	141
<b>make-modal-inverter</b> .....	881	marcado, texto, enmarcar .....	274
<b>make-modal-transposer</b> .....	881	marcado, texto, instrucciones de alineación .....	273
<b>make-music</b> .....	882	marcadores .....	534, 535
<b>make-oval-stencil</b> .....	882	<i>marcato</i> .....	128
<b>make-pango-font-tree</b> .....	284	<b>\marcato</b> .....	128, 800
<b>make-part-combine-context-changes</b> .....	882	'marcato', articulación .....	128, 800
<b>make-part-combine-marks</b> .....	882	margen de encuadernación .....	574
<b>make-partial-ellipse-stencil</b> .....	882	margen, texto fuera del .....	257
<b>make-path-stencil</b> .....	883	<b>\mark</b> .....	118, 260, 849
<b>make-repeat</b> .....	883	<i>Mark_engraver</i> .....	120, 262
<b>make-semitone-&gt;pitch</b> .....	883	<i>Mark_engraver</i> .....	262
<b>make-stencil-boxer</b> .....	884	<b>\markalphabet</b> .....	791
<b>make-stencil-circler</b> .....	884	<i>MarkEvent</i> .....	120, 262
<b>make-tmpfile</b> .....	884	<b>\markLengthOff</b> .....	76, 261
<b>make-transparent-box-stencil</b> .....	884	<b>\markLengthOn</b> .....	76, 261
<b>\makeClusters</b> .....	181, 849	<b>\markLetter</b> .....	792
<b>\makeDefaultStringTuning</b> .....	849	<b>\markup</b> .....	254, 260, 262, 263, 264, 661
mantener música etiquetada .....	541	<i>Markup functions</i> .....	266
mantenimiento, pedal, estilo de .....	362	markup text, line width .....	272
manual, barra .....	88, 100	<b>markup-command-list?</b> .....	884
manual, cambio de pentagrama .....	354	<b>markup-list?</b> .....	884
manual, edición de ligadura .....	61	<b>markup-markup-spacing</b> .....	572
manual, marca de ensayo .....	119	<b>markup-system-spacing</b> .....	572
manual, marca de repetición .....	166	<b>\markuplist</b> .....	263, 279
<i>Manuales</i> .....	1	<b>\markupMap</b> .....	849
manuales, barras de compás .....	105	matices .....	131
manuales, líneas divisorias .....	105	matices absolutos .....	131
<b>\map-markup-commands</b> .....	796	matices, posición vertical .....	133

matices, varios sobre una nota .....	132	microtono .....	9
matiz, alineación vertical .....	135	microtonos, tablatura .....	389
matiz, estilo de texto .....	136	middleCPosition .....	27
matiz, extensión de texto, personalizar .....	258	\midi .....	510, 622
matiz, línea de extensión, ocultar .....	136	midi-program .....	885
max-systems-per-page .....	575	MIDI .....	556, 559
maxima .....	51, 64	MIDI .....	28, 552
\maxima .....	49, 62	MIDI con repeticiones .....	560
maxima, silencio de .....	62	MIDI, balance .....	563
mayor 7, acorde, disposición .....	454	MIDI, balance estéreo .....	563
mayor, modo .....	23	MIDI, bloque .....	554
mayor, séptima, símbolo de .....	453	MIDI, canales .....	561
measure-counter::text .....	884	MIDI, chorus .....	563
Measure_grouping_engraver .....	97	MIDI, definiciones de contexto .....	558
measureLength .....	91, 126	MIDI, dinámicas personalizadas .....	555
measurePosition .....	78, 126	MIDI, ecualización .....	555
Medicaea, Editio .....	464, 465	MIDI, efectos .....	563
medida .....	70	MIDI, expresión .....	563
medida, música sin .....	126	MIDI, instrumento .....	564
medio, intervalo .....	497	MIDI, matices dinámicos .....	555
\medium .....	743	MIDI, metadatos de .....	525
melisma .....	295, 298	MIDI, notación contemplada .....	553
melisma .....	295, 298	MIDI, notación no contemplada .....	553
\melisma .....	295	MIDI, panorama .....	563
melisma, barras en .....	89	MIDI, pistas .....	561
\melismaEnd .....	295	MIDI, reverberación .....	563
melismata .....	295	MIDI, sustituir la ecualización de instrumento predeterminada .....	557
melodía alternativa, cambio a una .....	323	MIDI, transposición .....	28
melodía, mostrar el ritmo .....	86	MIDI, un canal por cada voz .....	562
menor, modo .....	23	MIDI, volumen .....	555
mensuración, símbolo de .....	470	midBalance .....	563
mensural .....	464, 465	midChannelMapping .....	561
mensural notation .. 464, 465, 466, 468, 469, 470, 471, 472, 473		midChorusLevel .....	563
mensural, clave .....	469, 734	midDrumPitches .....	426
mensural, corchete .....	471	midExpression .....	563
mensural, ligadura .....	474	midPanPosition .....	563
mensural, música, transcripción de .....	205	midReverbLevel .....	563
mensural, notación, signum congruentiae .....	802	min-systems-per-page .....	575
mensural-flag .....	885	minimum-Y-extent .....	590
MensuralStaff .....	202	minimumFret .....	373, 413
MensuralStaff .....	201, 468	minimumPageTurnLength .....	589
MensuralVoice .....	468	minimumRepeatLengthForPageTurn .....	589
Mensurstriche .....	491	\minor .....	23
mensurstriche, diseño .....	205	minorChordModifier .....	452
\mergeDifferentlyDottedOff .....	186	mirroring markup .....	773
\mergeDifferentlyDottedOn .....	186	mixed .....	362
\mergeDifferentlyHeadedOff .....	186	mixolidio, modo .....	23
\mergeDifferentlyHeadedOn .....	186	\mixolydian .....	23
merging text .....	750, 759	\mm .....	663
metadatos de PDF .....	525	mmrest-of-length .....	885
meter .....	83	modal, inversion .....	17
metrónomo, indicación de .....	74	modal, transformación .....	16
metrónomo, indicación, debajo del pentagrama .....	76	modal, transposición .....	16
metro polimétrico .....	81	\modalInversion .....	17, 849
metro, estilo de .....	70	\modalTranspose .....	16, 849
metronómica, marca de, personalizada .....	77	mode .....	807
metronome .....	78	modern .....	32
metronome mark .....	78	modern-cautionary .....	32
MetronomeMark .....	78	modern-straight-flag .....	885
metronomic indication .....	78	modern-voice .....	33
mezclar notas .....	186	modern-voice-cautionary .....	33
mezzosoprano, clave .....	734	moderna, transcripción del canto gregoriano .....	345
mezzosoprano, clave de .....	18	moderno de precaución, alteraciones de estilo .....	32
\mf .....	131	moderno de precaución, voz, estilo de alteración .....	33



moderno, alteraciones de estilo .....	32, 33
moderno, estilo de alteraciones .....	32
moderno-precaución, estilo de alteraciones .....	32
moderntab, clave .....	388
modificación del separador de acordes .....	455
modificador de acorde .....	443
<i>Modificar las propiedades de los contextos</i> .....	638
modificar símbolo de marca de respiración .....	145
modificar, flageolet, tamaño .....	429
modificar, número de puntillos .....	50
modo .....	23
modo eclesiástico .....	23
<code>\mordent</code> .....	128, 800
mordente .....	120, 128
‘mordente ascendente’, ornamento .....	128
‘mordente descendente’, ornamento .....	128, 800
mordente, ascendente .....	128
‘mordente, ascendente’, ornamento .....	800
mordente, circular .....	128
mordente, circular, cruzado .....	128
mordente, circular, de Haydn .....	128
mordente, circular, inverso .....	128
mordente, descendente .....	128
‘mordente, descendente’, ornamento .....	800
‘mordente’, ornamento .....	128, 800
mordentes .....	431
<i>Mover objetos</i> .....	269, 273
movimientos, varios .....	507
<code>\mp</code> .....	131
muerta, percusión, nota .....	427
<i>multi-measure rest</i> .....	69
multi-measure rest, within text, by duration .....	777
multi-measure rest, within text, by <i>duration-scale</i> .....	775
<code>\multi-measure-rest-by-number</code> .....	775
multicompás, notas, contraer .....	233
multicompás, notas, expandir .....	233
multicompás, posicionamiento de los silencios .....	67
multicompás, silencio, adjuntar texto .....	66
multicompás, silencio, aplicar un calderón a .....	66
multicompás, silencio, con marcados .....	66
multicompás, silencio, estilo .....	234
multicompás, silencio, longitud .....	67
multicompás, silencio, numerar .....	234
multicompás, silencios .....	65
multicompás, silencios, contraer .....	233
multicompás, silencios, digitaciones y .....	69
multicompás, silencios, expandir .....	233
multicompás, silencios, inscripción en .....	66
<i>MultiMeasureRest</i> .....	69, 235
<i>MultiMeasureRestNumber</i> .....	69, 235
<i>MultiMeasureRestScript</i> .....	69, 235
<i>MultiMeasureRestScript</i> .....	66
<i>MultiMeasureRestText</i> .....	69, 235
<i>MultiMeasureRestText</i> .....	66
<i>Music classes</i> .....	227
<code>music-&gt;make-music</code> .....	886
<code>music-clone</code> .....	886
<code>music-filter</code> .....	886
<code>music-is-of-type?</code> .....	887
<code>music-map</code> .....	887
<code>music-pitches</code> .....	887
<code>music-selective-filter</code> .....	887
<code>music-selective-map</code> .....	887
<code>music-separator?</code> .....	887

<code>music-type-predicate</code> .....	887
musica ficta .....	473
musical, glifo .....	119
musical, guía .....	334
musical, teatro .....	331
<code>\musicglyph</code> .....	119, 776
<code>\musicMap</code> .....	849
musicológico, análisis .....	251
<code>musicQuotes</code> .....	807

## N

<code>\n</code> .....	131
número de compás .....	126
número de compás y repeticiones .....	166
número de compás, alternativo, en repetición .....	164
número de compás, cadencias .....	79
número de compás, colisión .....	117
número de compás, comprobación de .....	117
número de compás, formato de .....	114
número de compás, música sin metro .....	79
número de cuerda frente a digitaciones .....	369
número, de compás .....	112
número, de compás, espaciado regular .....	112
número, de página, automático .....	577
número, de página, especificar el primero .....	577
número, de página, suprimir .....	577
número, grupo de valoración especial, modificación .....	53
números de página, romanos .....	577
números, en notación simplificada .....	42
N.C., símbolo .....	448
<code>\name</code> .....	638
<code>\natural</code> .....	776
natural, armónico .....	366
natural, nota .....	6
neo-modern .....	34
neo-modern-accidental-rule .....	887
neo-modern-cautionary .....	35
neo-modern-voice .....	35
neo-modern-voice, estilo de alteración .....	35
neo-modern-voice-cautionary .....	35
neo-modern-voice-cautionary, estilo de alteraciones .....	35
neo-moderno de precaución, estilo de alteraciones .....	35
neo-moderno, estilo de alteraciones .....	34
neomensural .....	465
<code>\new</code> .....	624
<i>New_fingering_engraver</i> .....	242, 643
<code>\newSpacingSection</code> .....	608
niente, al, regulador .....	135
nivel superior, texto del .....	262
no musicales, símbolos .....	275
no restablecer, estilo de alteraciones .....	36
no vacíos, textos .....	257
<code>no-flag</code> .....	888
no-puro: contenedor de Scheme .....	690
<code>no-reset</code> .....	36
<code>\noBeam</code> .....	101
<code>\noBreak</code> .....	582
nombre de nota .....	1
nombre del cantante .....	320

nombre, de instrumento, añadir a		
otros contextos.....	222	
nombre, de instrumento, cambiar .....	222	
nombres, de personaje.....	332	
<i>NonMusicalPaperColumn</i> .....	608	
<i>nonstaff-nonstaff-spacing</i> .....	590	
<i>nonstaff-relatedstaff-spacing</i> .....	590	
<i>nonstaff-unrelatedstaff-spacing</i> .....	590	
<i>\noPageBreak</i> .....	586, 849	
<i>\noPageTurn</i> .....	589, 849	
<i>normal-flag</i> .....	888	
<i>\normal-size-sub</i> .....	743	
<i>\normal-size-super</i> .....	267, 743	
<i>\normal-text</i> .....	743	
<i>\normalsize</i> .....	235, 268, 744	
nota al pie.....	526	
nota al pie dentro de expresiones musicales.....	526	
nota al pie en texto independiente.....	531	
Nota al pie, basada en eventos.....	527	
Nota al pie, basada en tiempo.....	528	
nota coloreada.....	243	
nota coloreada en acordes.....	244	
nota de adorno.....	431	
nota de adorno, cambiar los		
ajustes de disposición.....	122	
nota de adorno, trucar.....	122	
nota de espaciado.....	64	
nota entre paréntesis.....	245	
nota fantasma.....	245	
nota invisible.....	242	
nota oculta.....	242	
‘nota tapada’, articulación.....	128	
nota transparente.....	242	
nota, abierta.....	128	
nota, altura predeterminada.....	50	
nota, cabeza de, armónico.....	41	
nota, cabeza de, en aspás.....	41	
nota, cabeza de, en parlato.....	41	
nota, cabeza de, en rombo.....	41	
nota, cabeza de, estilos.....	41	
nota, cabeza de, forma de rombo.....	366	
nota, cabeza de, guitarra.....	41	
nota, cabeza especial de.....	41	
nota, cabeza, Aiken, variante fina.....	46	
nota, cabezas de, Aiken.....	44	
nota, cabezas de, arpa sacra.....	44	
nota, cabezas de, barradas.....	47	
nota, cabezas de, Christian Harmony.....	44	
nota, cabezas de, estudio.....	42	
nota, cabezas de, Funk.....	44	
nota, cabezas de, Harmonia Sacra.....	44	
nota, cabezas de, improvisación.....	47	
nota, cabezas de, notación simplificada.....	42	
nota, cabezas de, para practicar.....	42	
nota, cabezas de, Southern Harmony.....	44	
nota, cabezas de, Walker.....	44	
nota, colisiones de.....	186	
nota, con doble puntillo.....	50	
nota, con puntillo.....	50	
nota, con puntillo, cambiar número.....	50	
nota, con puntillo, mover horizontalmente.....	190	
nota, corchetes de agrupación de.....	251	
nota, de adorno, cambiar ajustes		
de presentación.....	123	
nota, de pentagrama cruzado.....	354, 359	
nota, desplazamiento horizontal.....	190	
nota, división de.....	84	
nota, duración de.....	49	
nota, duración predeterminada.....	49	
nota, espaciar horizontalmente.....	608	
nota, estilo de cabeza.....	734	
nota, guía.....	224, 227	
nota, longitud de.....	49	
nota, nombres de, en otros idiomas.....	8	
nota, nombres holandeses de.....	6	
nota, nombres predeterminados.....	6	
nota, nombres, árabes.....	498	
nota, nombres, Hel-arabic.....	498	
nota, pequeña.....	227	
nota, tapada.....	128	
nota, transporte de.....	11	
<i>Notación de los pentagramas</i> .....	208	
notación gráfica.....	275	
notación, dentro de marcado.....	276	
notación, explicación de.....	248	
notación, tamaño de la fuente.....	235	
notas, cabeza de.....	235	
notas, cabezas de, con formas.....	44	
notas, nombre, imprimir.....	247	
<i>\note</i> .....	777	
<i>note head</i> .....	471, 488	
<i>note value</i> .....	51	
note, within text, by duration.....	777	
note, within text, by log and dot-count.....	776	
<i>\note-by-number</i> .....	776	
<i>note-collision-interface</i> .....	828, 831, 833	
<i>note-column::main-extent</i> .....	888	
<i>note-event</i> .....	42, 44, 47	
<i>note-event</i> .....	226	
<i>note-head-interface</i> .....	42, 44, 47	
<i>note-name-&gt;markup</i> .....	888	
<i>note-name-&gt;string</i> .....	888	
<i>note-to-cluster</i> .....	888	
<i>Note_head_line_engraver</i> .....	359	
<i>Note_heads_engraver</i> .....	42, 44, 47, 85, 640	
<i>Note_heads_engraver</i> .....	84	
<i>Note_name_engraver</i> .....	248	
<i>Note_name_engraver</i> .....	247	
<i>Note_spacing_engraver</i> .....	243	
<i>NoteCollision</i> .....	191	
<i>NoteColumn</i> .....	191	
<i>NoteHead</i> .....	42, 44, 47	
<i>\notemode</i> .....	661	
<i>NoteName</i> .....	248	
<i>noteNameFunction</i> .....	247	
<i>NoteNames</i> .....	248	
<i>NoteNames</i> .....	247	
<i>noteNameSeparator</i> .....	247	
<i>NoteSpacing</i> .....	243, 607, 608	
nuevo pentagrama.....	201	
nuevos contextos.....	624	
<i>\null</i> .....	270, 792	
<i>NullVoice</i> .....	317	
<i>\number</i> .....	744	
<i>number-format</i> .....	888	
<i>\numericTimeSignature</i> .....	70	

## O

- objeto, marcado ..... 254
- objeto de presentación ..... 805
- objeto de propiedades ..... 807
- objeto de Scheme ..... 807
- objeto gráfico ..... 805
- objeto inmutable ..... 806
- objeto, coloreado de ..... 671
- objeto, colorear ..... 243
- objeto, rotación ..... 677
- objeto, sobreescritura de ..... 671
- objeto, visibilidad de ..... 670
- Objetos e interfaces* ..... 534, 806
- Objetos interiores al pentagrama* ..... 662, 663
- objetos mutables ..... 806
- octava, comprobación de ..... 10
- octava, corrección de ..... 10
- octava, elemento de extensión,
  - modificar inclinación ..... 27
- octava, especificación absoluta ..... 1
- octava, especificación relativa ..... 2
- octava, introducción absoluta ..... 1
- octava, introducción relativa ..... 2
- octava, marca de cambio de ..... 1
- octava, para una voz ..... 27
- octava, transposición ..... 18
- octava, transposición, opcional ..... 18
- octavación ..... 25, 26
- octavación, texto ..... 26
- octavation* ..... 28
- `\octaveCheck` ..... 10, 849
- oculta, nota ..... 242
- ocultar, alteración, en notas ligadas al
  - principio de un sistema ..... 7
- ocultar, línea de extensión de matiz ..... 136
- ocultar, pentagrama ..... 215
- `\offset` ..... 653, 849
- `offset-fret` ..... 888
- `offsetter` ..... 888
- Oigo voces* ..... 185, 423, 425
- `old-straight-flag` ..... 888
- olvidar, estilo de alteraciones ..... 37
- `\omit` ..... 670, 849
- `\on-the-fly` ..... 523, 792
- `\once` ..... 648, 650, 655, 689, 849
- `\oneVoice` ..... 181
- Opciones avanzadas de línea de*
  - órdenes para LilyPond ..... 551
- Opciones básicas de la línea de*
  - órdenes para LilyPond ..... 550
- `\open` ..... 128, 366, 429, 801
- operación de inversión ..... 14
- operación de inversión modal ..... 17
- operación de retrogradación ..... 15
- operación de transposición ..... 16
- operación modal ..... 16
- oratorio ..... 328
- ordenación vertical de las inscripciones ..... 129
- Organizar las piezas mediante variables* ..... 200, 334, 540, 545, 546, 627, 628
- orientación de las digitaciones ..... 240
- orientación, de las digitaciones ..... 240
- orientación, de los números de cuerda ..... 240
- orientación, modificación de,
  - diagrama de trastes ..... 398
- `\oriscus` ..... 480, 486
- ornamentación ..... 128
- ornamento ..... 120, 128
- ornamento, grupeto circular ..... 128, 800
- ornamento, grupeto circular barrado ..... 128
- ornamento, grupeto circular de Haydn ..... 128, 800
- ornamento, grupeto circular invertido ..... 128, 800
- ornamento, grupeto de retardación ..... 130
- ornamento, mordente ..... 128, 800
- ornamento, mordente, ascendente ..... 128, 800
- ornamento, mordente, descendente ..... 128, 800
- ornamento, semitrino ..... 128, 800
- ornamento, semitrino, ascendente ..... 128, 800
- ornamento, semitrino, descendente ..... 128, 800
- ornamento, semitrino, largo ..... 128, 800
- ornamento, signum congruentiae ..... 128
- ornamento, trino ..... 128, 800
- orquesta, notación para ..... 331
- orquestal, música ..... 331
- orquestales, cuerdas ..... 365
- ossia* ..... 215
- ossia* ..... 212, 220
- ossia*, posicionamiento ..... 214
- otomana, música ..... 503
- Otomana, música clásica ..... 497
- Otras aplicaciones de los trucos* ..... 354
- Otras fuentes de información* ..... 185, 497, 539, 540, 551, 552, 559, 560, 565, 644, 678
- ottava ..... 25
- `\ottava` ..... 25, 850
- ottava-bracket-interface* ..... 28
- Ottava-spanner-engraver* ..... 28
- OttavaBracket* ..... 28
- ottavation-numbers* ..... 25
- ottavation-ordinals* ..... 25
- ottavation-simple-ordinals* ..... 25
- ottavationMarkups* ..... 25
- `outer-margin` ..... 575
- `output-count` ..... 807
- `output-def` ..... 807
- `output-module?` ..... 890
- `output-suffix` ..... 807
- `outside-staff-horizontal-padding` ..... 605
- `outside-staff-padding` ..... 605
- `outside-staff-priority` ..... 605
- `\oval` ..... 770
- `oval-stencil` ..... 890
- `\overlay` ..... 759
- `\override` ..... 648, 652, 792
- `override-head-style` ..... 890
- `\override-lines` ..... 796
- `override-time-signature-setting` ..... 890
- OverrideProperty* ..... 646
- \overrideProperty* ..... 652, 850
- \overrideTimeSignatureSettings* ..... 71, 850
- overriding property within text markup ..... 792
- `\overtie` ..... 744
- overtie-ing text ..... 744

## P

<code>\p</code> .....	131
página, disposición de la .....	611
página, marcado de varias .....	279
página, orientación de la .....	569
página, referenciar números de .....	534
página, saltos de .....	611
página, saltos manuales .....	586
<code>\pad-around</code> .....	274, 759
<code>\pad-markup</code> .....	274, 760
<code>\pad-to-box</code> .....	274, 760
<code>\pad-x</code> .....	274, 760
padding (relleno) .....	645
padding in markup, vertical .....	271
padding text .....	760
padding text horizontally .....	760
<code>page-breaking</code> .....	576
<code>page-breaking-system-system-spacing</code> .....	576
<code>page-count</code> .....	576
<code>\page-link</code> .....	792
<code>page-number-type</code> .....	577
<code>\page-ref</code> .....	535, 793
<code>page-spacing-weight</code> .....	578
<code>\pageBreak</code> .....	586, 587, 850
<code>\pageTurn</code> .....	589, 850
palmas .....	802
<code>\palmMute</code> .....	850
<code>\palmMuteOn</code> .....	850
pandereta .....	802
Pango .....	280
<code>pango-pf-file-name</code> .....	890
<code>pango-pf-font-name</code> .....	890
<code>pango-pf-fontindex</code> .....	890
panorama en MIDI .....	563
papel, tamaño, apaisado .....	569
papel, tamaño, orientación .....	569
<code>\paper</code> .....	510, 568
<code>paper-height</code> .....	570
<code>paper-width</code> .....	573
paréntesis .....	245
paréntesis en ángulo .....	176
paréntesis, alteración con .....	7
paralelo, música en .....	197
<code>\parallelMusic</code> .....	197, 850
parcial, compás .....	78
parcial, compás, en himnos .....	350
<code>parentheses-interface</code> .....	245
<code>ParenthesesItem</code> .....	245
<code>Parenthesis-engraver</code> .....	245
<code>\parenthesize</code> .....	245, 770, 850
<code>parenthesize-stencil</code> .....	891
parlato .....	338
parlato, cabezas de nota .....	41
Parmesan, tipografía .....	713
<code>parse-terse-string</code> .....	892
parser .....	807
<code>part</code> .....	197
<code>\partCombine</code> .....	192, 317, 850
<code>\partCombine</code> y letra .....	195, 317
<code>\partCombineApart</code> .....	194
<code>\partCombineAutomatic</code> .....	194
<code>\partCombineChords</code> .....	194
<code>\partCombineDown</code> .....	851
<code>\partCombineForce</code> .....	851
<code>partCombineListener</code> .....	807
<code>PartCombineMusic</code> .....	197
<code>\partCombineSoloI</code> .....	194
<code>\partCombineSoloII</code> .....	194
<code>\partCombineUnisono</code> .....	194
<code>\partCombineUp</code> .....	851
parte al alzar .....	78
partes de sección .....	220
partes individuales .....	220
partes, combinación automática .....	192
partes, combinador de .....	192
partes, combinador, cambiar textos .....	196
partes, de atril .....	220
partes, de sección .....	220
partes, individuales .....	220
<code>\partial</code> .....	78, 159, 160, 851
particellas .....	220
partitura general .....	331
<i>Partitura vocal a cuatro voces SATB</i> .....	328
partitura, dentro de un marcado .....	278
partitura, general .....	331
partitura, vocal, añadir guías .....	336
<i>Partituras y particellas</i> .....	540
<code>\path</code> .....	771
path, drawing .....	771
patrón de discontinuidad, ligadura de expresión ..	141
<code>\pattern</code> .....	793
pausa, marcas de .....	144
pauta .....	201
pauta, de coro .....	202
pauta, de percusión .....	201
pauta, escalado de .....	212
pautas, arpegio entre .....	153
pedal de órgano .....	128
pedal de mantenimiento, estilo de .....	362
pedal high hat .....	425
pedal, de piano .....	361
pedal, del arpa .....	364
pedal, del arpa, diagrama .....	364
pedal, derecho .....	361
pedal, estilos de indicación .....	362
pedal, forte .....	361
pedal, high hat .....	128, 802
pedal, indicación mixta .....	362
pedal, indicación por corchete .....	362
pedal, indicación textual .....	362
pedal, indicación, órgano .....	128, 801
pedal, indicación, punta .....	128, 801
pedal, indicación, talón .....	128, 801
pedal, sostenuto .....	361
<code>pedalSustainStyle</code> .....	362
pentagrama único, polifonía de .....	181
pentagrama a la francesa .....	212
pentagrama cruzado, arpeggios con paréntesis .....	155
pentagrama cruzado, plica de .....	359
pentagrama, único .....	201
pentagrama, único, con cohete o llave .....	204
pentagrama, cambio .....	358
pentagrama, cambio automático de .....	356
pentagrama, cambio de .....	358
pentagrama, cambios manuales de .....	354
pentagrama, cruzado, trémolo de .....	174
pentagrama, de instrumentos de teclado .....	353
pentagrama, de percusión .....	201
pentagrama, de piano .....	202, 353

pentagrama, detener e iniciar líneas del .....	208	piano, sistema de .....	202
pentagrama, distancia entre los .....	590	<b>piano-cautionary</b> .....	34
pentagrama, establecer el símbolo del .....	664	<i>Piano_pedal_engraver</i> .....	362
pentagrama, fijar el tamaño de .....	581	<i>PianoPedalBracket</i> .....	362
pentagrama, grupo de .....	202	<i>PianoStaff</i> .....	37, 155, 205, 224, 329, 354
pentagrama, indicación metronómica debajo .....	76	<b>PianoStaff</b> .....	353, 356
pentagrama, iniciar .....	201	picado .....	128
pentagrama, instanciar .....	201	pie de página .....	512
pentagrama, línea de cambio de .....	358	pie, indicación de .....	128
pentagrama, modificar las líneas del .....	208	pie, marca de .....	801
pentagrama, nuevo .....	201	<i>Pitch names</i> .....	2, 6, 8, 10, 473
pentagrama, ocultación de .....	215	<i>Pitch_squash_engraver</i> .....	48, 88, 640, 819
pentagrama, símbolo del .....	208	<b>Pitch_squash_engraver</b> .....	86
pentagrama, vacío .....	215	<i>\pitchedTrill</i> .....	156, 851
pentagramas, anidados .....	205	<i>Pitches</i> ..	2, 6, 8, 10, 11, 14, 22, 25, 28, 30, 37, 40, 42, 44, 47, 48, 469, 501
pentagramas, divisi .....	218	<b>pitchnames</b> .....	807
pentagramas, varios .....	202	pizzicato, Bartók .....	367
pequeña, nota .....	227	pizzicato, snap .....	367
<b>percent</b> .....	170	placing horizontal brackets, around text .....	769
<i>percent repeat</i> .....	172	placing parentheses, around text .....	770
<i>PercentRepeat</i> .....	172	placing vertical brackets, around text .....	766
<i>PercentRepeatCounter</i> .....	172	plantilla, música árabe .....	502
<i>PercentRepeatedMusic</i> .....	172	<i>Plantillas de conjuntos vocales</i> ....	303, 305, 328, 346, 349, 626
percusión .....	420, 422, 802	<i>Plantillas de cuarteto de cuerda</i> .....	366
percusión, clave .....	421, 734	<i>Plantillas incorporadas</i> .....	328
percusión, nota falsa .....	427	platillo .....	802
percusión, nota fantasma .....	427	platillo, chino .....	802
percusión, nota muerta .....	427	platillo, ride .....	802
percusión, nota silenciada .....	427	platillo, splash .....	802
percusión, nota tapada .....	427	platillos, varios .....	802
percusión, pauta de .....	201	plica .....	245
percusión, personalizada .....	425	plica abajo .....	245
<i>Percussion</i> .....	420, 421, 422, 425, 427	plica arriba .....	245
<b>percussion?</b> .....	892	plica de pentagrama cruzado .....	359
persa, makam .....	497	plica invisible .....	245
personaje, nombre .....	332	plica neutral .....	245
personalización de los nombres de acorde .....	450	plica, de pentagrama cruzado .....	359
personalizada, marca de ensayo .....	119	plica, dirección .....	245
personalizada, marca metronómica .....	77	plica, dirección automática sobre la línea media ..	246
personalizadas, afinaciones de cuerdas .....	387	plica, dirección predeterminada	
personalizado, añadir diagramas de traste .....	403	sobre la línea media .....	246
personalizado, diagrama de traste .....	405	plica, en tablatura .....	378
personalizado, diagrama de trastes .....	390, 398	plica, tachada .....	122
personalizados, diagramas de trastes .....	397	<i>\pointAndClickOff</i> .....	851
<i>\pes</i> .....	486	<i>\pointAndClickOn</i> .....	851
Petrucchi .....	464, 465	<i>\pointAndClickTypes</i> .....	851
Petrucchi, clave .....	469, 734	<b>polar-&gt;rectangular</b> .....	893
<i>PhrasingSlur</i> .....	144	polifónica, música .....	186
<i>\phrasingSlurDashed</i> .....	143	polifonía, en tablaturas .....	378
<i>\phrasingSlurDashPattern</i> .....	144, 851	polifonía, en un solo pentagrama .....	181
<i>\phrasingSlurDotted</i> .....	143	polifonía, letra compartida .....	317
<i>\phrasingSlurDown</i> .....	143	polifonía, voces adicionales .....	189
<i>\phrasingSlurHalfDashed</i> .....	143	polimétrica, partitura .....	632
<i>\phrasingSlurHalfSolid</i> .....	143	polimétrico, compás .....	81
<i>\phrasingSlurNeutral</i> .....	143	polimétrico, metro, barras en .....	81
<i>\phrasingSlurSolid</i> .....	143	<i>\polygon</i> .....	772
<i>\phrasingSlurUp</i> .....	143	<i>polymetric</i> .....	56, 83
<i>\phrygian</i> .....	23	<i>polymetric time signature</i> .....	83
piano .....	33	<i>polyphony</i> .....	190
piano de precaución, estilo de alteraciones .....	34	porcentaje, contador de repetición .....	171
piano, estilo de alteraciones de .....	33	porcentaje, repetición de .....	170
piano, música de, matices centrados .....	353	porcentaje, repetición, aislada .....	172
piano, pedales de .....	361	porcentaje, repetición, visibilidad del contador ...	172
piano, pentagramas de .....	202, 353		

<i>portato</i> .....	130
portato .....	128
<code>\portato</code> .....	128, 800
‘portato’, articulación .....	128, 800
posición y cejilla, indicación para instrumentos de trastes .....	415
posición, alteración de bajo cifrado .....	459
posicionamiento, de la letra .....	301
posicionamiento, de letra .....	214
posicionamiento, de ossia .....	214
posicionamiento, digitación, mano derecha .....	414
post-eventos .....	800
PostScript .....	275
<code>\postscript</code> .....	275, 772
PostScript encapsulado, salida de .....	550
potencia, acorde de .....	417
<i>power chord</i> .....	418
<code>\pp</code> .....	131
<code>\ppp</code> .....	131
<code>\pppp</code> .....	131
<code>\prall</code> .....	128, 800
<code>\pralldown</code> .....	128, 800
<code>\prallmordent</code> .....	128, 800
<code>\prallprall</code> .....	128, 800
‘pralltriller, ascendente’, ornamento .....	128
‘pralltriller, descendente’, ornamento .....	128
‘pralltriller, largo’, ornamento .....	128
‘pralltriller’, ornamento .....	128
<code>\prallup</code> .....	128, 800
<code>\preBend</code> .....	851
<code>\preBendHold</code> .....	851
precaución, alteración de .....	7
<code>predefinedDiagramTable</code> .....	409
<code>\predefinedFretboardsOff</code> .....	411
<code>\predefinedFretboardsOn</code> .....	411
predeterminada, altura .....	50
predeterminada, duración de la nota .....	49
predeterminada, línea divisoria, cambiar .....	111
predeterminadas, cambio de propiedades de contexto .....	633
predeterminado, estilo de las alteraciones .....	30
predeterminado, nombres de nota .....	6
<i>Presentación de las expresiones musicales</i> ...	565, 652
presentación, interfaz de la .....	643
prima volta .....	159
primera y segunda vez, estilo de repetición doble .....	164
primera y segunda vez .....	159
primera y segunda vez, acortar corchete .....	163
primera y segunda vez, con ligaduras .....	161
primera y segunda vez, corchete en pautas adicionales .....	164
primera y segunda vez, corchetes y ligaduras en ...	59
primera y segunda vez, repetición .....	168
primera y segunda vez, y letra .....	311
principiantes, música para .....	42
<code>print-all-headers</code> .....	578
<code>print-first-page-number</code> .....	577
<code>print-page-number</code> .....	577
<code>printAccidentalNames</code> .....	247
<code>printNotesLanguage</code> .....	247
<code>printOctaveNames</code> .....	247
prob .....	807
progresiva, barras .....	103
<code>\property-recursive</code> .....	793

<code>\propertyOverride</code> .....	851
<code>\propertyRevert</code> .....	851
<i>PropertySet</i> .....	646
<code>\propertySet</code> .....	851
<code>\propertyTweak</code> .....	851
<code>\propertyUnset</code> .....	852
propiedad .....	646
propiedad de grob .....	648
propiedad, compartida .....	806
propiedad, inmutable .....	806
<i>Propiedades de los interfaces</i> .....	806
<i>Propiedades de los objetos de presentación</i> .....	806
propiedades mutables .....	806
<code>\pt</code> .....	663
‘pulgár’, digitación .....	128, 240, 801
pull off .....	385
pulso, agrupación .....	97
pulsos por minuto .....	74
‘punta’, indicación de pedal .....	801
‘punta’, marca de pedal .....	128
punteada, ligadura de fraseo .....	143
puntillo, cambiar número de .....	50
puntillo, mover horizontalmente .....	190
puntillo, nota con .....	50
puntos de control en curvas de Bézier .....	684
puntos, ligadura de .....	59, 140
puntuación .....	287
puntuación, en la letra .....	287
<code>pure-chain-offset-callback</code> .....	894
puro, contenedor de Scheme .....	690
<code>\pushToTag</code> .....	545, 852
<code>\put-adjacent</code> .....	761
putting space around text .....	760

## Q

q, repetición de acordes .....	178, 373
<i>quarter tone</i> .....	8
quebrado, acorde .....	152
<i>\quilisma</i> .....	480, 486
quinta vacía, acorde de .....	417
quitar música etiquetada .....	541
<code>quotedCueEventTypes</code> .....	226
<code>quotedEventTypes</code> .....	226
<code>\quoteDuring</code> .....	224, 227, 852
<i>QuoteMusic</i> .....	227

## R

<i>r</i> .....	62
rítmica, pauta .....	201
<i>R</i> .....	65
racimo (cluster) .....	181
<code>ragged-bottom</code> .....	570
<code>ragged-last</code> .....	574, 611
<code>ragged-last-bottom</code> .....	570
<code>ragged-right</code> .....	574, 611
<code>\raise</code> .....	270, 761
<code>\raiseNote</code> .....	447, 852
raising text .....	761
rango de alturas .....	38
rasgueado, mostrar ritmos .....	86
<i>rast</i> .....	501
<code>ratio-&gt;fret</code> .....	894

<b>ratio-&gt;pitch</b> .....	894	repetición, barra de .....	104
<b>Ratisbona, Editio</b> .....	465	repetición, cambiar finales alternativos .....	166
<b>read-lily-expression</b> .....	894	repetición, cambiar número de .....	166
recordatoria, alteración .....	7	repetición, comienzo de .....	166
<b>recording-group-emulate</b> .....	895	repetición, con anacrusa .....	160
recto, corchete .....	102	repetición, con comprobaciones de compás .....	160
redonda, silencios de, para un compás completo ..	65	repetición, con finales alternativos .....	159
<b>\reduceChords</b> .....	86, 852	repetición, con ligaduras .....	161
Referencia de funcionamiento interno .....	622	repetición, con primera y segunda vez .....	168
referenciar contextos .....	624	repetición, con <b>q</b> .....	178
referencing page label, in text .....	795	repetición, con segno .....	161
referencing page number, in text .....	792, 793	repetición, contador de tiempo de la .....	166
registro, símbolo, acordeón .....	362	repetición, contador, porcentaje .....	171
regulador .....	131	repetición, corta .....	170
regulador acampanados .....	135	repetición, de compás .....	170
regulador de Ferneyhough .....	135	repetición, de la letra con finales alternativos ....	311
regulador de tipo constante .....	135	repetición, de la letra con primera	
regulador en ángulo .....	677	y segunda vez .....	311
regulador, al niente .....	135	repetición, de porcentaje .....	170
regulador, desplazar extremos .....	134	repetición, de trémolo .....	173
regulador, longitud mínima .....	134	repetición, desplegada .....	168
reguladores sobre líneas divisorias .....	133	repetición, desplegada con finales alternativos ....	168
<b>RehearsalMark</b> .....	120, 262	repetición, doble, estilo para	
rejilla, líneas de .....	249	primera y segunda vez .....	164
rejilla, líneas de, modificación .....	250	repetición, explícita .....	168
relativa, especificación de la octava .....	2	repetición, final de .....	166
relativa, introducción de la octava, y acordes .....	5	repetición, letras y números de compás .....	164
relativas, especificación de octavas .....	2	repetición, ligaduras de unión en .....	59
<b>\relative</b> .....	2, 6, 14, 357, 852	repetición, números de compás alternativos .....	164
<b>RelativeOctaveCheck</b> .....	11	repetición, números de compás de	
<b>RelativeOctaveMusic</b> .....	6	primera y segunda vez .....	164
relativo .....	2	repetición, normal .....	159
relativo, acordes en altura .....	177	repetición, porcentaje, aislada .....	172
relativo, modo, <b>\autoChange</b> y .....	357	repetición, porcentaje, visibilidad del contador ...	172
relativo, modo, transposición y .....	6	repetición, y glissando .....	166
relleno .....	645	repetición, y glissandos .....	150
relleno, alrededor de texto .....	274	repetición, y letra .....	307
<b>\remove</b> .....	631	repetición, y ligadura de expresión .....	166
<b>remove-empty</b> .....	216	repetición, y números de compás .....	166
<b>remove-first</b> .....	216	repeticiones en MIDI .....	560
<b>remove-grace-property</b> .....	895	repetido, acorde, eliminación .....	449
<b>remove-grace-property</b> .....	123	repetido, acorde, suprimir .....	408
<b>remove-layer</b> .....	218	<b>\replace</b> .....	745
<b>remove-whitespace</b> .....	895	reservados, impresión de caracteres .....	265
<b>\RemoveAllEmptyStaves</b> .....	215, 856	<b>\resetRelativeOctave</b> .....	6, 852
<b>\RemoveEmptyStaves</b> .....	215, 856	respiración, cambiar símbolo de marca de .....	145
<b>\removeWithTag</b> .....	541, 852	respiraciones .....	144
renacimiento, música del .....	205	<b>\responsum</b> .....	481
<b>repeat</b> .....	166	<b>Rest</b> .....	64
<b>\repeat</b> .....	159	<b>\rest</b> .....	62, 777
<b>\repeat percent</b> .....	170	rest, within text, by duration .....	777
<b>\repeat tremolo</b> .....	173	rest, within text, by <b>log</b> and <b>dot-count</b> .....	777
<b>repeatCommands</b> .....	166	<b>\rest-by-number</b> .....	777
<b>repeatCountVisibility</b> .....	172	<b>rest-event</b> .....	226
<b>RepeatedMusic</b> .....	166, 168, 170	<b>Rest-engraver</b> .....	85
<b>Repeats</b> .....	166, 168, 170, 172, 175	restaurar, las propiedades predeterminadas	
<b>RepeatSlash</b> .....	172	del tipo de compás .....	72
<b>RepeatSlashEvent</b> .....	172	<b>RestCollision</b> .....	191
<b>\repeatTie</b> .....	59, 161, 313	<b>restNumberThreshold</b> .....	234
repetición .....	107	<b>restrainOpenStrings</b> .....	373
repetición con <b>q</b> .....	373	retardación, grupeto de, ornamento .....	130
repetición manual .....	166	<b>retrieve-glyph-flag</b> .....	895
repetición, alternativas .....	168	retrogradación, transformación de .....	15
repetición, ambigua .....	166	<b>\retrograde</b> .....	15, 852
repetición, anidada .....	166	<b>retrograde-music</b> .....	895

reverberación en MIDI .....	563
<code>\reverseturn</code> .....	128, 800
reversión de una sobreescritura .....	649
<code>\revert</code> .....	649
<code>revert-fontSize</code> .....	895
<code>revert-head-style</code> .....	895
<code>revert-props</code> .....	895
<i>RevertProperty</i> .....	646
<code>\revertTimeSignatureSettings</code> .....	72, 852
<code>\rfz</code> .....	131
rgb, color .....	244
rgb-color .....	244
<code>\rheel</code> .....	128, 801
<i>Rhythmic.column-engraver</i> .....	640
<i>RhythmicStaff</i> .....	48, 88, 202
<i>RhythmicStaff</i> .....	201
<i>Rhythms</i> .. 51, 56, 58, 62, 64, 65, 69, 74, 79, 81, 84, 85, 88, 91, 99, 103, 104, 112, 117, 118, 120, 124, 126, 127	
<code>\right-align</code> .....	269, 761
right-aligning text .....	761
<code>\right-brace</code> .....	793
<code>\right-column</code> .....	761
right-margin .....	574
<code>\rightHandFinger</code> .....	414, 852
ritmo, de la melodía, mostrar .....	86
<code>\roman</code> .....	745
<code>\romanStringNumbers</code> .....	366, 370
rombo, cabezas de nota .....	41
rombo, cabezas en forma de .....	366
rotar un objeto .....	677
<code>\rotate</code> .....	762
rotating text .....	762
<code>\rounded-box</code> .....	274, 773
<code>rounded-box-stencil</code> .....	896
<code>\rtoe</code> .....	128, 801

## S

<i>s</i> .....	64
séptima, acorde de .....	443
sílaba, duración automática de .....	290
símbolo de séptima mayor .....	453
símbolo, cambiar, marca de respiración .....	145
símbolo, coda .....	128
símbolo, coda variante .....	128
símbolo, de dirección .....	97
símbolo, fuentes tipográficas en el marcado .....	280
símbolo, segno .....	128
símbolos no musicales .....	275
<code>\sacredHarpHeads</code> .....	44
<code>\sacredHarpHeadsMinor</code> .....	45
salida de gráficos de vector escalables .....	550
salida de PostScript encapsulado .....	550
salida, definiciones de .....	622
salmo .....	339, 345
<i>Salmos</i> .....	349
saltar el tipografiado .....	549
saltar notas en la letra .....	312
salto, de línea .....	582
salto, de línea, a intervalos .....	584
salto, de línea, cadencias .....	81
salto, de línea, música sin medida .....	81
salto, de página .....	611
salto, de página, cadencias .....	81

salto, de página, gestión con una voz adicional ...	585
salto, de página, música sin medida .....	81
salto, en música sin medida .....	81
<code>\sans</code> .....	745
SATB .....	328
<code>\scale</code> .....	773
<code>scale-beam-thickness</code> .....	896
<code>scale-fontSize</code> .....	896
<code>scale-layout</code> .....	896
<code>scale-props</code> .....	896
<code>\scaleDurations</code> .....	57, 81, 852
scaling markup .....	773
scaling text .....	762
Scheme, contenedor no-puro .....	690
Scheme, contenedor puro .....	690
<i>scordatura</i> .....	25
<i>Score</i> .....	127, 809, 812
<code>\score</code> .....	506, 510, 778
<code>\score-lines</code> .....	796
<code>score-markup-spacing</code> .....	572
<code>score-system-spacing</code> .....	572
<code>scoreTitleMarkup</code> .....	520
<i>scorify-music</i> .....	897
<i>Script</i> .....	128, 130, 479
<i>Script-engraver</i> .....	479
<i>ScriptEvent</i> .....	479
sección de espaciado nueva .....	608
seconda volta .....	159
<code>seconds-&gt;moment</code> .....	897
segno .....	128
<code>\segno</code> .....	128, 801
'segno', con repeticiones .....	161
'segno', signo .....	119, 128, 801
'segno', signo de .....	106
'segno', sobre la línea divisoria .....	260
seguimiento de voz .....	358
selección del tamaño de la fuente tipográfica (de notación) .....	235
<code>select-head-glyph</code> .....	897
<i>self-alignment-interface</i> .....	644, 678
<code>self-alignment-interface::self-aligned-on-breakable</code> ..	897
<code>self-alignment-X</code> .....	590
sello .....	808
sello, eliminar .....	670
<i>semal</i> .....	502
Semal, forma .....	501
'semi abierta', articulación .....	128
semi abierto, high hat .....	128
semi-bemol .....	9
semi-bemol, apariencia .....	499
semi-bemoles .....	6
semi-sostenido .....	9
semi-sostenidos .....	6
'semiabierta', articulación .....	801
semiabierto, high hat .....	802
<code>\semicirculus</code> .....	478, 802
'semicirculus', articulación gregoriana .....	478
'semicirculus', articulación, gregoriana .....	802
<code>\semiflat</code> .....	779
<code>\semiGermanChords</code> .....	450
<code>\semisharp</code> .....	779
semitrino .....	128
'semitrino ascendente', ornamento .....	128, 800
'semitrino descendente', ornamento .....	800
semitrino, ascendente .....	128



'semitrino, ascendente', ornamento.....	800	silencio, de redonda para un compás completo.....	65
semitrino, descendente.....	128	silencio, división de.....	84
'semitrino, descendente', ornamento.....	800	silencio, eclesiástico.....	234
semitrino, doble.....	128	silencio, especificar la posición vertical.....	62
'semitrino, largo', ornamento.....	800	silencio, estilo de.....	62
'semitrino', ornamento.....	128, 800	silencio, multicompas.....	65
separación de sistemas, marca de.....	207	silencio, multicompas, marcado.....	69
separado, texto.....	262	silencios de compás completo.....	62
<b>sequential-music-to-chord-exceptions</b> .....	897	silencios multicompas.....	62
sesqui-bemol.....	9	silencios, indicar duraciones.....	62
sesqui-sostenido.....	9	<i>simile</i> .....	172
<code>\sesquiflat</code> .....	779	<code>\simple</code> .....	745
<code>\sesquisharp</code> .....	780	simple text string.....	745
<code>session-save</code> .....	897	simple text string, with tie characters.....	780
<code>\set</code> .....	91, 646, 652	simplificada, notación.....	42
<code>set-accidental-style</code> .....	897	simultáneas, ligaduras de expresión.....	140
<code>set-global-fonts</code> .....	284	simultáneas, ligaduras de fraseo.....	143
<code>set-global-staff-size</code> .....	897	simultáneas, notas y alteraciones.....	37
<code>set-global-staff-size</code> .....	581	<i>Simultaneous notes</i> .....	177, 181, 185, 191, 197, 200
<code>set-mus-properties!</code> .....	898	sin acorde, símbolo.....	448
setting extent of text object.....	795	sin medida, música, saltos de línea en.....	81
setting horizontal text alignment.....	754	sin medida, música, saltos de página en.....	81
setting subscript, in standard font size.....	743	<code>\single</code> .....	529, 655, 853
setting superscript, in standard font size.....	743	sintaxis del marcado.....	264
<code>\settingsFrom</code> .....	852	sistema.....	202
<code>\sf</code> .....	131	sistema de coro.....	202
<code>\sff</code> .....	131	sistema de piano.....	202
<code>\sfz</code> .....	131	sistema, delimitador de comienzo.....	202
<code>\shape</code> .....	684, 852	sistema, delimitadores de inicio anidados.....	205
<i>sharp</i> .....	8	<code>\skip</code> .....	64, 312, 853
<code>\sharp</code> .....	780	<code>skip-&gt;rest</code> .....	898
<code>shift-one-duration-log</code> .....	898	<code>skip-of-length</code> .....	898
<code>shift-right-at-line-begin</code> .....	898	<code>skipBars</code> .....	234
<code>\shiftDurations</code> .....	853	<i>SkipMusic</i> .....	65
<code>\shiftOff</code> .....	186	<code>skipTypesetting</code> .....	549
<code>\shiftOn</code> .....	186	<i>Slash-repeat-engraver</i> .....	173
<code>\shiftOnn</code> .....	186	<code>slashChordSeparator</code> .....	452
<code>\shiftOnnn</code> .....	186	slashed digit.....	793
<code>short-indent</code> .....	222, 575	<code>\slashed-digit</code> .....	793
<code>\shortfermata</code> .....	128, 801	<code>\slashedGrace</code> .....	120, 853
<code>show-available-fonts</code> .....	283	<code>\slashSeparator</code> .....	578
<code>showFirstLength</code> .....	549, 807	<code>\slashturn</code> .....	128, 800
<code>\showKeySignature</code> .....	431	<i>Slur</i> .....	142, 155
<code>showLastLength</code> .....	549, 807	<i>slur</i> .....	142
<code>\showStaffSwitch</code> .....	358	<code>slur-event</code> .....	226
<i>side-position-interface</i> .....	644, 678	<code>\slurDashed</code> .....	140
sidestick.....	802	<code>\slurDashPattern</code> .....	141, 853
signo, coda.....	119, 128, 801	<code>\slurDotted</code> .....	140
signo, coda de variación.....	128, 801	<code>\slurDown</code> .....	140
signo, 'segno'.....	106, 119, 128, 801	<code>\slurHalfDashed</code> .....	140
signo, snappizzicato.....	128, 801	<code>\slurHalfSolid</code> .....	140
'signum congruentiae', notación mensural.....	802	<code>\slurNeutral</code> .....	140
'signum congruentiae', ornamento.....	128	<i>slurs</i> .....	62
<code>\signumcongruentiae</code> .....	128, 802	<code>\slurSolid</code> .....	140
<i>sikah</i> .....	501	<code>\slurUp</code> .....	141
silbato.....	802	<code>\small</code> .....	235, 268, 746
silenciada, percusión, nota.....	427	<code>\smallCaps</code> .....	746
silencio.....	62	<code>\smaller</code> .....	266, 268, 746
silencio antiguo.....	472	<i>smob</i> .....	807
silencio de separación.....	64	<code>snap pizzicato</code> .....	367
silencio invisible.....	64	<code>\snappizzicato</code> .....	128, 801
silencio multicompas, colocación.....	67	'snappizzicato', sign.....	128
silencio normal, condensar.....	70	'snappizzicato', signo de.....	801
silencio, colisiones de.....	70	sobreescritura de un objeto.....	671
silencio, de compás completo.....	65	sobreescritura para un solo momento.....	650

sobreescritura, reversión .....	649	<i>StanzaNumber</i> .....	326
Sol, clave de .....	18	<b>start-repeat</b> .....	166
Solesmes .....	465	<b>startAcciaccaturaMusic</b> .....	123
solo, partes .....	192	<b>startAppoggiaturaMusic</b> .....	123
sonido .....	552	<b>startGraceMusic</b> .....	123
soprano, clave .....	734	<b>\startGroup</b> .....	251
soprano, clave de .....	18	<b>\startStaff</b> .....	208, 212
sos. ....	361	<b>\startTrillSpan</b> .....	155
sostenido .....	6	<i>staves</i> .....	202
sostenido, doble .....	6	<b>\stdBass</b> .....	785
sostenuto, pedal .....	361	<b>\stdBassIV</b> .....	786
<i>SostenutoEvent</i> .....	362	<b>\stdBassV</b> .....	787
<b>\sostenutoOff</b> .....	361	<b>\stdBassVI</b> .....	788
<b>\sostenutoOn</b> .....	361	<i>Stem</i> .....	247, 361, 654
<i>SostenutoPedal</i> .....	362	<i>stem-interface</i> .....	247
<i>SostenutoPedalLineSpanner</i> .....	362	<b>stem-spacing-correction</b> .....	607
<b>\sourcefileline</b> .....	511	<i>Stem_engraver</i> .....	103, 247
<b>\sourcefilename</b> .....	511	<b>\stemDown</b> .....	245
Southern Harmony, cabezas de nota con forma ....	44	<b>stemLeftBeamCount</b> .....	101, 102
<b>\southernHarmonyHeads</b> .....	44	<b>\stemNeutral</b> .....	245
<b>\southernHarmonyHeadsMinor</b> .....	45	<b>stemRightBeamCount</b> .....	101, 102
<b>\sp</b> .....	131	<b>\stemUp</b> .....	245
<i>Spacing</i> .. 570, 573, 575, 579, 581, 582, 586, 587, 588,		<i>stencil</i> .....	808
589, 594, 595, 597, 605, 606, 608, 609, 611, 618, 619, 621		<b>\stencil</b> .....	794
<b>spacing</b> .....	607	<b>stencil-whiteout</b> .....	901
<i>spacing-spanner-interface</i> .....	838, 840	<b>stencil-whiteout-box</b> .....	901
<i>SpacingSpanner</i> .....	606, 607, 608, 609	<b>stencil-whiteout-outline</b> .....	901
<b>\spacingTweaks</b> .....	853	<b>stopAcciaccaturaMusic</b> .....	123
<b>span-bar::compound-bar-line</b> .....	899	<b>stopAppoggiaturaMusic</b> .....	123
<i>Span_stem_engraver</i> .....	899	<b>stopGraceMusic</b> .....	123
<i>Span_stem_engraver</i> .....	359	<b>\stopGroup</b> .....	251
<i>SpanBar</i> .....	112	<b>\stopped</b> .....	128, 425, 429, 801
<b>split-list-by-separator</b> .....	899	<b>\stopStaff</b> .....	208, 212, 215
<b>\spp</b> .....	131	<b>\stopTrillSpan</b> .....	155
Sprechgesang .....	338	<b>\storePredefinedDiagram</b> .....	404, 409, 853
staccatissimo .....	128	<b>straight-flag</b> .....	902
<b>\staccatissimo</b> .....	128, 800	<b>strictBeatBeaming</b> .....	97
'staccatissimo', articulación .....	128, 800	<b>\string-lines</b> .....	796
<i>staccato</i> .....	130	<i>StringNumber</i> .....	371
staccato .....	128	<b>stringNumberOrientations</b> .....	240
<b>\staccato</b> .....	128, 800	<b>\stringTuning</b> .....	387, 853
'staccato', articulación .....	128, 800	<b>stringTunings</b> .....	386, 400
<b>stack-lines</b> .....	900	<i>StrokeFinger</i> .....	415
<b>stack-stencil-line</b> .....	900	<b>strokeFingerOrientations</b> .....	240, 414
<b>stack-stencils</b> .....	900	<b>\stroph</b> .....	480, 486
<b>stack-stencils-padding-list</b> .....	900	<b>\strut</b> .....	794
stacking text in a column .....	750	<b>style-note-heads</b> .....	902
<i>Staff</i> .... 37, 40, 84, 202, 205, 220, 224, 253, 607, 809		<b>\styledNoteHeads</b> .....	853
<i>staff</i> .....	202, 211, 215	<b>\sub</b> .....	267, 746
<i>Staff notation</i> .. 78, 202, 205, 207, 211, 215, 220, 224,		subíndice .....	267
227, 232		subbajo, clave .....	734
<b>staff-affinity</b> .....	590	<b>subdivideBeams</b> .....	96
<b>staff-padding</b> .....	241	subdivisión, de barra .....	96
<b>\staff-space</b> .....	663	subscript text .....	746
<b>staff-staff-spacing</b> .....	590	<b>suggestAccidentals</b> .....	130, 473
<i>staff-symbol-interface</i> .....	211	<b>\super</b> .....	267, 747
<i>Staff.midiInstrument</i> .....	564	superíndice .....	267
<i>Staff_collecting_engraver</i> .....	262	superscript text .....	747
<i>Staff_symbol_engraver</i> .....	220	supresión, acorde repetido .....	408
<b>Staff_symbol_engraver</b> .....	215	<b>sus</b> .....	446
<i>StaffGroup</i> .....	117, 205, 207	<i>SustainEvent</i> .....	362
<b>staffgroup-staff-spacing</b> .....	590	<b>\sustainOff</b> .....	361
<i>StaffGrouper</i> .....	330, 591, 593, 595, 658	<b>\sustainOn</b> .....	361
<i>StaffSpacing</i> .....	608	<i>SustainPedal</i> .....	362
<i>StaffSymbol</i> .....	202, 211, 215	<i>SustainPedalLineSpanner</i> .....	362

SVG, salida .....	550
symbol-concatenate .....	902
system-count .....	576
system-separator-markup .....	578
system-system-spacing .....	572
systems-per-page .....	576
SystemStartBar .....	205, 207
SystemStartBrace .....	205, 207
SystemStartBracket .....	205, 207
SystemStartSquare .....	205, 207

## T

título .....	512
Tab_note_heads_engraver .....	390
\tabChordRepeats .....	373, 853
\tabChordRepetition .....	853
\tabFullNotation .....	372
tabla de contenidos .....	535
tablatura .....	369
tablatura de banjo .....	369, 418
tablatura de cuartos de tono .....	389
tablatura de microtonos .....	389
tablatura predeterminada .....	371
tablatura y armónicos .....	376
tablatura y deslizamientos .....	383
tablatura, afinaciones personalizadas .....	387
tablatura, afinaciones predefinidas de .....	386
tablatura, armónicos y .....	379
tablatura, barras y .....	378
tablatura, clave .....	734
tablatura, clave de .....	388
tablatura, de bajo .....	386
tablatura, de banjo .....	386
tablatura, de contrabajo .....	386
tablatura, de guitarra .....	386
tablatura, de mandolina .....	386
tablatura, de ukelele .....	386
tablatura, de violín .....	386
tablatura, de viola .....	386
tablatura, de violoncello .....	386
tablatura, glisando en acorde de .....	384
tablatura, hammer on .....	385
tablatura, laúd .....	419
tablatura, pauta de .....	201
tablatura, personalizada .....	386
tablatura, plicas y .....	378
tablatura, principios básicos .....	371
tablatura, pull off .....	385
tablatura, y polifonía .....	378
\table .....	797
\table-of-contents .....	538, 797
TabNoteHead .....	386
TabStaff .....	202, 386
TabStaff .....	201, 371
TabVoice .....	386
TabVoice .....	371
tachada, plica .....	122
\tag .....	541, 853
tag-group-get .....	902
\tagGroup .....	544, 853
tags-keep-predicate .....	902
tags-remove-predicate .....	902
‘talón’, indicación de pedal .....	801

‘talón’, marca de pedal .....	128
tam tam .....	802
tamaño de la página .....	568
Tamaño de los objetos .....	215
tamaño del papel .....	568
\taor .....	431
‘tapada’, articulación .....	429, 801
tapada, nota, en instrumento de trastes .....	416
tapada, percusión, nota .....	427
taqasim .....	502
taqasim .....	501
teaching .....	36
teaching (enseñanza), estilo de alteraciones .....	36
teaching-accidental-rule .....	902
tecla, pentagramas para instrumentos de .....	353
teclado, matices centrados .....	353
teclado, pentagramas para instrumentos de .....	353
\teeny .....	235, 268, 747
tempo .....	74
\tempo .....	74
tempo indication .....	78
tempo, cambio, sin indicación metronómica .....	77
\temporary .....	655, 689, 853
tenor Sol, clave .....	734
tenor varC, clave .....	734
tenor, clave .....	734
tenor, clave de .....	18
tenor, clave de, coral .....	18
tenuto .....	130
tenuto .....	128
\tenuto .....	128, 800
‘tenuto’, articulación .....	128, 800
tesitura .....	38
tesitura, línea de separación .....	39
tesitura, por voz .....	38
tesitura, varias voces .....	39
Text .....	256, 257, 260, 262, 264, 266, 269, 274, 276, 278, 279, 283, 288
text .....	362, 747
text column, left-aligned .....	758
text column, right-aligned .....	761
text, line width .....	272
text-interface .....	644, 792
text-script-interface .....	644
Text_engraver .....	640
\textLengthOff .....	67, 69, 257
\textLengthOn .....	67, 69, 135, 257
texto ajustado a palabras .....	273
texto de marcado ajustado a palabras .....	273
texto del nivel superior .....	262
texto destacado .....	257
texto independiente .....	262
texto justificado .....	273
texto separado .....	262
texto, alineación de .....	269
texto, alineación horizontal .....	269
texto, alineación vertical .....	270
texto, alineación vertical de inscripciones de .....	135
texto, centrar en la página .....	272
texto, de octava .....	26
texto, de varias líneas .....	271
texto, decorar .....	274
texto, elemento de extensión de .....	258
texto, elementos no vacíos de .....	257
texto, en casilla de repetición .....	168

texto, en columnas .....	263, 271	tipografía, Emmentaler .....	713
texto, en otros idiomas .....	254	tipografía, Feta .....	713
texto, enmarcar .....	274	tipografía, fijar el tamaño de .....	581
texto, entrecomillado en modo de marcado .....	265	tipografía, notación musical .....	280
texto, extensión de, matiz, personalizar .....	258	tipografiado, saltar .....	549
texto, formateado de los		tipografiar texto .....	264
elementos de extensión .....	258	<b>tocFormatMarkup</b> .....	538
texto, fuera del margen .....	257	<b>tocIndentMarkup</b> .....	538
texto, indicaciones de .....	260	<b>\tocItem</b> .....	538, 854
texto, inscripciones de .....	257	<b>tocItemMarkup</b> .....	538
texto, instrucciones de alineación .....	273	<b>\tocItemWithDotsMarkup</b> .....	537
texto, mantener dentro del margen .....	257	<b>tocTitleMarkup</b> .....	538
texto, marcado de .....	264	tom tom .....	802
texto, marcado de, dentro de las ligaduras .....	141	tom tom agudo .....	802
texto, matiz de estilo de .....	136	tom tom de suelo .....	802
texto, objeto .....	254	tom tom grave .....	802
texto, que ocupa varias páginas .....	279	tom tom medio .....	802
texto, relleno de .....	274	<i>Top</i> .....	1, 622
texto, sobre la barra de compás .....	260	<b>top-margin</b> .....	570
texto, sobre un silencio multicompa .....	66	<b>top-markup-spacing</b> .....	572
texto, tamaño de .....	266	<b>top-system-spacing</b> .....	572
<i>TextScript</i> .. 130, 257, 264, 269, 274, 276, 278, 279, 441		<b>toplevel-bookparts</b> .....	807
<i>TextSpanner</i> .....	260, 669	<b>toplevel-scores</b> .....	807
<b>\textSpannerDown</b> .....	258	trémolo .....	173
<b>\textSpannerNeutral</b> .....	258	trémolo, indicación de .....	174
<b>\textSpannerUp</b> .....	258	trémolo, pentagrama cruzado .....	174
<i>The Emmentaler font</i> .....	776	tríada .....	443
<b>\thumb</b> .....	128, 240	<i>Trabajar sobre los archivos de entrada</i> .....	507
<i>Tie</i> .....	62	transcripción de música mensural .....	205
<i>tie</i> .....	61, 85	transcripción de mensural a moderno .....	495
<b>\tie</b> .....	747	transcripción, moderna del canto gregoriano .....	345
tie-ing text .....	747	transformación de retrogradación .....	15
<i>TieColumn</i> .....	62, 688	transformación, modal .....	16
<b>TieColumn</b> .....	61	<b>\translate</b> .....	270, 762
<b>\tied-lyric</b> .....	780	<b>\translate-scaled</b> .....	270, 762
<b>\tieDashed</b> .....	59	translating text .....	762
<b>\tieDashPattern</b> .....	59, 854	<i>Translation</i> .....	643
<b>\tieDotted</b> .....	59	transparencia, semi .....	243
<b>\tieDown</b> .....	59	<b>\transparent</b> .....	794
<b>\tieHalfDashed</b> .....	59	transparente, nota .....	242
<b>\tieHalfSolid</b> .....	59	transparentes, hacer los objetos .....	670
tiempo, administración .....	126	transportada, visibilidad de una clave .....	676
tiempo, control del (dentro de la partitura) .....	126	transporte .....	11, 16
tiempo, marca de, para glissando .....	149	transporte de diagramas de trastes .....	402
tiempo, repeticiones y contador de .....	166	transporte modal .....	16
<b>\tieNeutral</b> .....	59	<b>\transpose</b> .....	6, 11, 14, 854
<b>\tieSolid</b> .....	59	<b>\transposedCueDuring</b> .....	231, 854
<b>\tieUp</b> .....	59	<i>TransposedMusic</i> .....	14
<b>tieWaitForNote</b> .....	61	transposición .....	11
timbala .....	802	transposición de MIDI .....	28
timbala abierta .....	802	transposición instrumental .....	28
timbala aguda .....	802	transposición modal .....	16
timbala grave .....	802	transposición, alturas, ‘inteligente’ .....	13
timbala tapada .....	802	transposición, de alturas .....	11
<b>\time</b> .....	70, 91, 854	transposición, de notas .....	11
<i>time signature</i> .....	74	transposición, de octava, opcional .....	18
<b>\times</b> .....	854	transposición, modo relativo y .....	6
<i>TimeScaledMusic</i> .....	56	<i>transposing instrument</i> .....	30, 332
<i>TimeSignature</i> .....	74, 84	<b>\transposition</b> .....	28, 224, 854
<b>timeSignatureFraction</b> .....	81	transpositor, instrumento .....	12, 28
<i>Timing_translator</i> .....	74, 79, 84, 112, 127, 812	transpositoras, claves .....	18
<b>\tiny</b> .....	235, 268, 748	transpositores, instrumentos .....	12
tipografía .....	805	traste .....	373
tipografía Parmesan .....	713	traste, definir predeterminado .....	406
tipografía, archivos externos .....	280	traste, diagrama personalizado de .....	405

traste, diagramas de, personalizados .....	390
trastes personalizados, añadir diagramas de .....	403
trastes, añadir digitaciones a diagramas de .....	412
trastes, armónicos en instrumento de .....	416
trastes, diagrama automático de .....	411
trastes, diagrama de, cambiar la orientación .....	398
trastes, diagrama de, con nombres de acorde .....	402
trastes, diagrama de, personalizado .....	398
trastes, diagrama de, transporte .....	402
trastes, diagrama, zurdo .....	393
trastes, diagramas de .....	390
trastes, diagramas personalizados de .....	397
trastes, instrumentos con, afinaciones predefinidas .....	386
trastes, instrumentos de, digitaciones de la mano derecha .....	414
trastes, instrumentos de, formas de acordes .....	404
trastes, instrumentos de, indicación de la posición y cejilla .....	415
trastes, notas tapadas en instrumento de .....	416
trastes, tabla alternativa .....	409
tre corde .....	361
treble, clave .....	734
<code>\treCorde</code> .....	361
tremolo .....	173
tresillo .....	51
tresillo, formato de .....	53
triángulo .....	802
<code>\triangle</code> .....	275, 773
<i>trill</i> .....	158
<code>\trill</code> .....	128, 155, 800
<i>TrillSpanner</i> .....	158, 669
trino .....	128, 155
trino con nota y alteración .....	157
‘trino’, ornamento .....	800
trinos de nota explícita .....	156
<i>triplet</i> .....	56
trucaje, de las notas de adorno .....	122
trucaje, nota de adorno .....	123
trucar .....	650
<i>Trucar la salida</i> .....	622, 678
trucar un punto de control .....	652
<i>Trucos difíciles</i> .....	690
<i>Tunable context properties</i> .....	297, 298, 648
<i>tuplet</i> .....	56
<code>\tuplet</code> .....	51, 81, 854
<i>tuplet-slur</i> .....	52
<i>TupletBracket</i> .....	56
<code>\tupletDown</code> .....	52
<code>\tupletNeutral</code> .....	52
<i>TupletNumber</i> .....	56
<i>TupletNumber</i> .....	53
<code>\tupletSpan</code> .....	53, 854
<i>tupletSpannerDuration</i> .....	53
<code>\tupletUp</code> .....	52
turco, makam .....	497
<code>\turn</code> .....	128, 800
Turquía, música clásica .....	497
Turquía, música de .....	503
Turquía, nombres de notas en .....	503
<i>Tutorial de Scheme</i> .....	622
<code>\tweak</code> .....	650, 652, 854
<i>tweak</i> , relación con <code>\override</code> .....	652
<i>Tweaks and overrides</i> .....	678
<i>two-sided</i> .....	574

<code>\type</code> .....	638
<code>\typewriter</code> .....	748

## U

U.C. ....	361
ukelele .....	391
una corda .....	361
<code>\unaCorda</code> .....	361
<i>UnaCordaEvent</i> .....	362
<i>UnaCordaPedal</i> .....	362
<i>UnaCordaPedalLineSpanner</i> .....	362
<i>unbreakable-spanner-interface</i> .....	91
<code>\underline</code> .....	266, 748
underlining text .....	748
<code>\undertie</code> .....	749
undertie-ing text .....	749
<code>\undo</code> .....	655, 855
<i>unfold</i> .....	168
<i>unfold-repeats</i> .....	903
<i>unfold-repeats-fully</i> .....	903
<i>\unfolded</i> .....	855
<i>UnfoldedRepeatedMusic</i> .....	166, 170
<i>\unfoldRepeats</i> .....	560, 855
<i>Unfretted strings</i> .....	366
<code>\unHideNotes</code> .....	242
Unicode .....	546
<i>uniq-list</i> .....	903
<i>unity-if-multimeasure</i> .....	904
<code>\unset</code> .....	647
<code>\upbow</code> .....	128, 366, 801
<code>\upmordent</code> .....	128, 800
<code>\upprall</code> .....	128, 800
<code>\upright</code> .....	749
UTF-8 .....	546
<i>Utilización desde la línea de órdenes</i> .....	549

## V

vacía, acorde de quinta .....	417
vacío, acorde .....	123, 354
vacío, pentagrama .....	215
valores predeterminados, desplazar .....	653
varbaritone, clave .....	734
varC, clave .....	734
varcoda .....	128
<code>\varcoda</code> .....	128, 801
variable de Scheme .....	807
variable del analizador sintáctico .....	807
variable global .....	807
variables .....	511
variables, uso de las .....	540
varias ligaduras de expresión .....	140
varias ligaduras de fraseo .....	143
<i>Varias notas a la vez</i> .....	191
varias voces .....	186
varias, líneas, marcado .....	271
varias, líneas, texto .....	271
varios matices sobre una nota .....	132
Vaticana, Editio .....	464, 465
<i>VaticanaStaff</i> .....	202
<i>VaticanaStaff</i> .....	201, 476
<i>VaticanaVoice</i> .....	476
<code>\vcenter</code> .....	763

<code>\verbatim-file</code> .....	794
<code>\version</code> .....	511
verso, número de .....	319
<code>\versus</code> .....	481
vertical padding in markup.....	271
vertical, alineación de texto.....	270
vertical, espaciado .....	589, 611
vertical, inscripciones, alineación.....	135
vertical, línea, entre pentagramas .....	249
vertical, matices, alineación.....	135
vertical, orden de las inscripciones .....	129
vertical, posición, de las indicaciones dinámicas ..	133
<code>VerticalAxisGroup</code> .....	220, 330, 591, 593, 594, 595, 597, 856
<code>VerticalAxisGroup</code> .....	590
vertically centering text .....	763
<code>\verylongfermata</code> .....	128, 801
<code>\veryshortfermata</code> .....	128, 801
vibraslap.....	802
viento, instrumento de.....	428
viento, instrumento, diagrama, lista.....	437
viento, instrumento, diagrama, lista de claves....	441
viento, instrumento, diagrama, modificar.....	439
viento, instrumentos, símbolos de digitación.....	429
violín, clave .....	734
violín, clave de .....	18
<code>\virga</code> .....	480, 486
<code>\virgula</code> .....	478
visibilidad de los grupos especiales.....	53
visibilidad de un objeto.....	670
visibilidad de una clave transportada .....	676
<i>Visibilidad y color de los objetos</i> ... 65, 220, 243, 346, 631, 670, 672, 676	
visibilidad, corchete de grupo especial.....	54
<i>Vocal music</i> .....	287, 327, 329, 332, 338, 339
vocal, partitura .....	331
vocal, partitura, añadir guías .....	336
vocal, transición de.....	299
<i>Voces explícitas</i> .....	183, 185
voces, alteraciones en varias .....	33
voces, arpeggio entre .....	154
<i>Voice</i> .....	40, 48, 194, 197, 227, 232, 294, 607, 645
<i>voice</i> .....	30, 32
<i>Voice</i> .....	181
<i>VoiceFollower</i> .....	359, 669
<code>\voiceFour</code> .....	181
<code>\voiceFourStyle</code> .....	185
<code>\voiceNeutralStyle</code> .....	185
<code>\voiceOne</code> .....	181
<code>\voiceOneStyle</code> .....	185
<code>\voices</code> .....	184, 855
<code>\voiceThree</code> .....	181
<code>\voiceThreeStyle</code> .....	185
<code>\voiceTwo</code> .....	181
<code>\voiceTwoStyle</code> .....	185
<i>voicify-music</i> .....	904
<code>\void</code> .....	565, 855
<i>volta</i> .....	166
<i>volta</i> .....	159
<code>\volta</code> .....	855
volta, prima.....	159
volta, repetición de, debajo de los acordes.....	454
volta, seconda .....	159

<code>volta-bracket::calc-hook-visibility</code> .....	904
<code>volta-spec-music</code> .....	904
<i>Volta_engraver</i> .....	450
<i>Volta_engraver</i> .....	164
<i>VoltaBracket</i> .....	166, 168
<i>VoltaRepeatedMusic</i> .....	166, 168
<i>vowel transition</i> .....	299
<i>VowelTransition</i> .....	299
voz.....	181
voz, adicional, en música polifónica .....	189
voz, adicional, para la gestión de los saltos.....	585
voz, citar.....	227
voz, con octavación.....	27
voz, dividida .....	330
voz, estilo .....	185
voz, estilo de alteraciones.....	32
voz, <code>\partCombine</code> con <code>\autoBeamOff</code> .....	90
voz, seguimiento de.....	358
voz, tesitura.....	38
voz, varias .....	186
<code>\vshape</code> .....	855
<code>\vspace</code> .....	271, 763

## W

Walker, cabezas de nota con forma.....	44
<code>\walkerHeads</code> .....	44
<code>\walkerHeadsMinor</code> .....	45
<code>whichBar</code> .....	111
<code>\whiteout</code> .....	794
<code>\whiteTriangleMarkup</code> .....	451
<i>Winds</i> .....	429, 431, 432, 433, 441
<code>\with</code> .....	631, 635
<i>with-color</i> .....	243
<code>\with-color</code> .....	243, 795
<code>\with-dimensions</code> .....	795
<code>\with-dimensions-from</code> .....	795
<code>\with-link</code> .....	795
<code>\with-outline</code> .....	796
<code>\with-url</code> .....	774
<code>\withMusicProperty</code> .....	855
<code>\woodwind-diagram</code> .....	783
<code>\wordwrap</code> .....	273, 764
<code>\wordwrap-field</code> .....	763
<code>\wordwrap-internal</code> .....	798
<code>\wordwrap-lines</code> .....	279, 798
<code>\wordwrap-string</code> .....	764
<code>\wordwrap-string-internal</code> .....	798
<i>World music</i> .....	498, 499, 501, 502
<i>write-me</i> .....	904

## X

<i>x11</i> , color de .....	244
<i>x11-color</i> .....	243, 244
<i>X-offset</i> .....	590
<code>\xNote</code> .....	41, 855
<code>\xNotesOff</code> .....	41
<code>\xNotesOn</code> .....	41

## Z

zurdo, diagrama de trastes.....	393
---------------------------------	-----