

# LilyPond

---

El tipografiador de música

## Manual del usuario

### El equipo de desarrollo de LilyPond

Copyright © 1999–2007 por los autores

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

*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.*

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 LilyPond versión 2.11.57

---

# 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	4
	Nombres de las notas en otros idiomas	6
1.1.2	Modificación de varias notas a la vez	7
	Comprobación de octava	7
	Transposición	8
1.1.3	Imprimir las alturas	11
	Clave	11
	Armadura de la tonalidad	14
	Corchetes de octava	16
	Transposición de los instrumentos	17
	Alteraciones accidentales automáticas	18
	Tesitura	24
1.1.4	Cabeza de las notas	26
	Cabezas de nota especiales	26
	Cabezas de notas de Notación Fácil	26
	Cabezas de notas con formas diversas	27
	Improvisación	28
1.2	Duraciones	29
1.2.1	Escritura de las duraciones (valores rítmicos)	30
1.2.1.1	Duraciones	30
1.2.1.2	Grupos especiales	31
1.2.1.3	Escalar las duraciones	34
1.2.1.4	Ligaduras de unión	35
1.2.2	Escritura de silencios	37
1.2.2.1	Silencios	37
1.2.2.2	Silencios invisibles	39
1.2.2.3	Silencios de compás completo	40
1.2.3	Impresión de las duraciones	43
1.2.3.1	Indicación de compás	43
1.2.3.2	Anacrusas	45
1.2.3.3	Música sin compasear	46
1.2.3.4	Notación polimétrica	46
1.2.3.5	División automática de las notas	46
1.2.3.6	Mostrar los ritmos de la melodía	47
1.2.4	Barras	47
1.2.4.1	Barras automáticas	47
1.2.4.2	Establecer el comportamiento de las barras automáticas	49
1.2.4.3	Barras manuales	52
1.2.4.4	Barras progresivas	53
1.2.5	Barras de compás	53
1.2.5.1	Barras de compás	54
1.2.5.2	Numeración de compases	55
1.2.5.3	Comprobación de compás y de número de compás	56

1.2.5.4	Llamadas de ensayo.....	57
1.2.6	Asuntos rítmicos especiales.....	59
1.2.6.1	Notas de adorno.....	59
1.2.6.2	Alinear con una cadenza.....	62
1.2.6.3	Gestión del tiempo.....	63
1.3	Expresiones.....	64
1.3.1	Adosado a las notas.....	64
	Articulaciones y ornamentos.....	64
	Matices dinámicos.....	66
	Indicaciones dinámicas contemporáneas.....	70
1.3.2	Curvas.....	70
	Ligaduras de expresión.....	70
	Ligaduras de fraseo.....	71
	Marcas de respiración.....	72
	Caídas y elevaciones.....	73
1.3.3	Líneas.....	73
	Glissando.....	73
	Arpeggio.....	74
	Trinos.....	77
1.4	Repeticiones.....	78
1.4.1	Escribir repeticiones largas.....	78
1.4.1.1	Repeticiones normales.....	79
1.4.1.2	Marcas de repetición manual.....	81
1.4.1.3	Repeticiones explícitas.....	82
1.4.2	Repeticiones cortas.....	82
	Repeticiones de compás o parte de ellos.....	82
	Repeticiones de trémolo.....	83
1.5	Notas simultáneas.....	84
1.5.1	Una voz única.....	85
	Notas en acorde.....	85
	Racimos (clusters).....	85
1.5.2	Varias voces.....	86
	Polifonía en un solo pentagrama.....	86
	Estilos de voz.....	87
	Resolución de las colisiones.....	88
	Combinación automática de las partes.....	91
	Escribir música en paralelo.....	95
1.6	Notación de los pentagramas.....	97
1.6.1	Impresión de los pentagramas.....	98
1.6.1.1	Delimitadores del comienzo de un sistema.....	98
1.6.1.2	El símbolo del pentagrama.....	100
1.6.1.3	Ocultar pentagramas.....	100
1.6.2	Escritura de las particellas.....	101
	Indicaciones metronómicas.....	101
	Nombres de instrumentos.....	102
	Citar otras voces.....	105
	Formateo de las notas de aviso.....	106
1.7	Anotaciones editoriales.....	108
1.7.1	Dentro del pentagrama.....	108
	Seleccionar el tamaño de la tipografía para la notación.....	108
	Indicaciones de digitación.....	109
	Notas ocultas.....	111
	Colorear objetos.....	111
	Paréntesis.....	113

Plicas .....	113
1.7.2 Fuera del pentagrama .....	114
Globos de ayuda .....	114
Líneas de rejilla .....	115
Corchetes de análisis .....	117
1.8 Texto .....	118
1.8.1 Escritura del texto .....	118
1.8.1.1 Guiones de texto .....	118
1.8.1.2 Extensiones de texto .....	119
1.8.1.3 Indicaciones de texto .....	120
1.8.1.4 Texto separado .....	124
1.8.2 Formatear el texto .....	125
1.8.2.1 Introducción al marcado de texto .....	125
1.8.2.2 Instrucciones de marcado de texto más usuales .....	126
1.8.2.3 Alineación de texto .....	126
1.8.2.4 Notación gráfica dentro de elementos de marcado .....	127
1.8.2.5 Notación musical dentro de elementos de marcado .....	127
1.8.2.6 Elementos de marcado de varias páginas .....	127
1.8.3 Tipografías .....	127
1.8.3.1 Fuentes tipográficas del documento completo .....	127
1.8.3.2 Fuentes de un solo elemento .....	128

## 2 Notación especializada ..... 130

2.1 Música vocal .....	130
2.1.1 Notación usual para voces .....	130
2.1.1.1 Referencias para música vocal .....	130
2.1.1.2 Elaborar canciones sencillas .....	130
2.1.1.3 Introducir la letra .....	131
2.1.1.4 Trabajar con el texto y las variables .....	133
2.1.2 Alineación de la letra a una melodía .....	134
2.1.2.1 Duración automática de las sílabas .....	134
2.1.2.2 Duración manual de las sílabas .....	135
2.1.2.3 Varias sílabas sobre una nota .....	135
2.1.2.4 Varias notas sobre una sílaba .....	136
2.1.2.5 Saltar notas .....	137
2.1.2.6 Líneas de extensión y guiones .....	137
2.1.2.7 Letras y repeticiones .....	137
2.1.3 Colocación de la letra .....	137
2.1.3.1 Letras en divisi .....	137
2.1.3.2 Letra independiente de las notas .....	139
2.1.3.3 Cantos .....	139
2.1.3.4 Espaciado de las sílabas .....	139
2.1.3.5 Centrado de la letra entre pentagramas .....	140
2.1.4 Versos .....	140
2.1.4.1 Adding stanza numbers .....	140
2.1.4.2 Añadir expresiones dinámicas a los versos .....	141
2.1.4.3 Añadir el nombre de los cantantes a los versos .....	141
2.1.4.4 Versos con ritmos distintos .....	142
2.1.4.5 Imprimir los versos al final .....	143
2.1.4.6 Imprimir los versos al final en varias columnas .....	144
2.2 Instrumentos de teclado .....	146
2.2.1 Notación usual para instrumentos de teclado .....	146
Referencias para teclados .....	146
Cambiar de pentagrama manualmente .....	146



Cambiar de pentagrama automáticamente .....	146
Líneas de cambio de pentagrama .....	147
Plicas de pentagrama cruzado .....	148
2.2.2 Piano .....	148
Pedales de piano .....	148
2.2.3 Acordeón .....	150
Símbolos de Discanto .....	150
2.3 Instrumentos de cuerda sin trastes .....	150
2.3.1 Notación usual para cuerdas sin trastes .....	150
2.3.1.1 Referencias para cuerdas sin trastes .....	150
2.3.2 Instrumentos de arco .....	150
2.3.2.1 Referencias para instrumentos de cuerda frotada .....	150
2.3.3 Instrumentos de cuerda pulsada .....	150
2.3.3.1 Arpa .....	150
2.4 Instrumentos de cuerda con trastes .....	150
2.4.1 Notación usual para cuerdas con trastes .....	150
Referencias para cuerdas con trastes .....	150
Indicación de los números de cuerda .....	150
Tablaturas predeterminadas .....	151
Tablaturas personalizadas .....	152
Marcas de diagramas de trastes .....	153
Digitaciones de la mano derecha .....	154
2.4.2 Guitarra .....	155
Indicar la posición y la cejilla .....	156
Indicar armónicos y notas tapadas .....	156
2.4.3 Banjo .....	156
Tablaturas de banjo .....	156
2.5 Percusión .....	157
2.5.1 Notación usual para percusión .....	157
2.5.1.1 Referencias para percusión .....	157
2.5.1.2 Notación básica de percusión .....	157
2.5.1.3 Pautas de percusión .....	158
2.5.1.4 Notas fantasma .....	161
2.6 Instrumentos de viento .....	161
2.6.1 Notación usual para instrumentos de viento .....	161
Referencias para instrumentos de viento .....	161
Digitaciones .....	161
2.6.2 Gaita .....	161
2.6.2.1 Definiciones para la gaita .....	161
2.6.2.2 Ejemplo de música de gaita .....	162
2.7 Notación de acordes .....	163
2.7.1 Modo de acordes .....	163
Panorámica del modo de acordes .....	163
Acordes más usuales .....	165
Acordes extendidos y alterados .....	165
2.7.2 Imprimir los acordes .....	168
Impresión de los nombres de acorde .....	168
Nombres de acorde personalizados .....	171
2.7.3 Bajo cifrado .....	174
Introducción al bajo cifrado .....	174
Introducir el bajo cifrado .....	175
Imprimir el bajo cifrado .....	177
2.8 Notación antigua .....	180
2.8.1 Antiguos .....	181

2.8.1.1	Cabezas de nota antiguas .....	181
2.8.1.2	Alteraciones accidentales antiguos .....	181
2.8.1.3	Silencios antiguos .....	182
2.8.1.4	Claves antiguas .....	182
2.8.1.5	Corchetes antiguos .....	184
2.8.1.6	Indicaciones antiguas de compás .....	185
2.8.1.7	Articulaciones antiguas .....	186
2.8.1.8	Custos .....	187
2.8.1.9	Divisiones .....	188
2.8.1.10	Ligaduras .....	188
2.8.1.11	Ligaduras mensurales blancas .....	189
2.8.1.12	Ligaduras de neumas cuadrados gregorianos .....	190
2.8.1.13	Contextos del Canto Gregoriano .....	196
2.8.1.14	Contextos de la música mensural .....	197
2.8.1.15	Alteraciones de la Musica Ficta .....	197
2.9	Músicas del mundo .....	198
2.9.1	Música árabe .....	198
	References for Arabic music .....	198
	Arabic note names .....	198
	Arabic key signatures .....	198
	Arabic time signatures .....	198
	Arabic music example .....	198
	Lecturas adicionales .....	198
<b>3</b>	<b>Entrada y salida generales .....</b>	<b>199</b>
3.1	Estructura del código de entrada .....	199
3.1.1	Estructura de una partitura .....	199
3.1.2	Varias partituras en un libro .....	199
3.1.3	Estructura del archivo .....	199
3.2	Títulos y cabeceras .....	199
3.2.1	Crear títulos .....	199
3.2.2	Títulos personalizados .....	202
3.2.3	Referencias a números de página .....	203
3.2.4	Índice general .....	204
3.3	Trabajar sobre los archivos de entrada .....	206
3.3.1	Inclusión de archivos de LilyPond .....	206
3.3.2	Distintas ediciones a partir de una misma fuente .....	206
	Uso de las variables .....	206
	Uso de etiquetas .....	206
3.3.3	Codificación del texto .....	206
3.3.4	Presentar la notación de LilyPond .....	206
3.4	Controlar la salida .....	206
3.4.1	Extracción de fragmentos de música .....	206
3.4.2	Saltar la música corregida .....	206
3.5	Salida MIDI .....	206
3.5.1	Crear archivos MIDI .....	207
	Nombres de instrumentos .....	208
3.5.2	El bloque MIDI .....	208
3.5.3	¿Qué sale por el MIDI? .....	209
	Contemplado en el MIDI .....	209
	No contemplado en el MIDI .....	209
3.5.4	Repeticiones y MIDI .....	209
3.5.5	Control de los matices en el MIDI .....	209
	Indicaciones dinámicas .....	209

Volumen maestro MIDI .....	209
Balance entre instrumentos (i) .....	210
Balance entre instrumentos (i) .....	210
<b>4 Problemas de espaciado .....</b>	<b>211</b>
4.1 El papel y las páginas .....	211
4.1.1 Tamaño del papel .....	211
4.1.2 Formateo de las páginas .....	211
4.2 Disposición de la música .....	211
4.2.1 Establecer el tamaño del pentagrama .....	211
4.2.2 Disposición de la partitura .....	211
4.3 Saltos .....	211
4.3.1 Saltos de línea .....	211
4.3.2 Saltos de página .....	211
4.3.3 Saltos de página óptimos .....	211
4.3.4 Paso de página óptimo .....	211
4.3.5 Saltos de página mínimos .....	211
4.3.6 Saltos de línea explícitos .....	211
4.3.7 Utilizar una voz adicional para los saltos de línea .....	211
4.4 Espaciado vertical .....	211
4.4.1 Espaciado vertical dentro de un sistema .....	211
4.4.2 Espaciado vertical entre sistemas .....	211
4.4.3 Posicionamiento explícito de los pentagramas y los sistemas .....	211
4.4.4 Espaciado vertical en dos fases .....	211
4.4.5 Evitar las colisiones verticales .....	211
4.5 Espaciado horizontal .....	211
4.5.1 Panorámica del espaciado horizontal .....	211
4.5.2 Área de espaciado nueva .....	211
4.5.3 Cambiar el espaciado horizontal .....	211
4.5.4 Longitud de la línea .....	211
4.5.5 Notación proporcional .....	212
4.6 Encajar la música en menos páginas .....	212
4.6.1 Mostrar el espaciado .....	212
4.6.2 Cambiar el espaciado .....	212
<b>5 Cambiar los valores por omisión .....</b>	<b>213</b>
5.1 Contextos de interpretación .....	213
5.1.1 Explicación de los contextos .....	213
5.1.2 Crear contextos .....	214
5.1.3 Modificar los complementos (plug-ins) de contexto .....	216
5.1.4 Cambiar los valores por omisión de los contextos .....	217
5.1.5 Definir contextos nuevos .....	218
5.1.6 Alinear los contextos .....	220
5.2 Explicación del Manual de referencia de funcionamiento interno .....	220
5.2.1 Navegar por la referencia del programa .....	220
5.2.2 Interfaces de la presentación .....	221
5.2.3 Determinar la propiedad del grob .....	222
5.2.4 Convenciones de nombres .....	223
5.3 Modificar las propiedades .....	223
5.3.1 Panorámica de la modificación de las propiedades .....	223
5.3.2 La instrucción <code>\set</code> .....	225
5.3.3 La instrucción <code>\override</code> .....	226
5.3.4 <code>\set</code> frente a <code>\override</code> .....	227

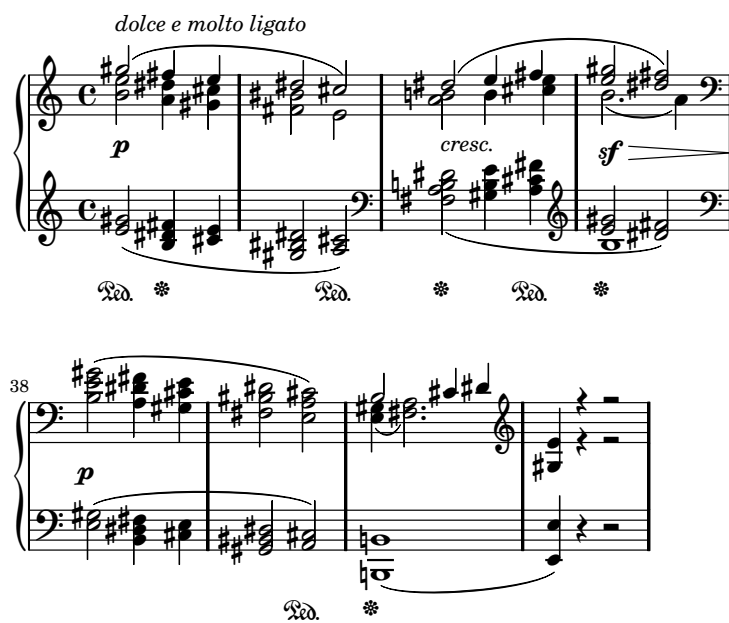
5.3.5	La instrucción <code>\tweak</code> .....	227
5.4	Conceptos y propiedades útiles .....	228
5.4.1	Modos de entrada .....	228
5.4.2	Dirección y posición .....	230
5.4.3	Distancias y medidas .....	230
5.4.4	Objetos de extensión .....	231
5.5	Propiedades más usuales .....	231
5.5.1	Control de la visibilidad de los objetos .....	231
5.5.2	Estilos de línea .....	231
5.5.3	Rotación de objetos .....	233
5.5.4	Alineación de objetos .....	233
5.6	Trucos avanzados .....	233
5.6.1	Agrupación vertical de objetos gráficos .....	233
5.6.2	Modificación de los extremos de los extensores .....	234
5.6.3	Modificación de los sellos .....	234
<b>6</b>	<b>Interfaces para programadores .....</b>	<b>235</b>
6.1	Funciones musicales .....	235
6.1.1	Panorámica de las funciones musicales .....	235
6.1.2	Funciones de sustitución sencillas .....	235
6.1.3	Funciones de sustitución en parejas .....	235
6.1.4	Matemáticas dentro de las funciones .....	235
6.1.5	Funciones vacías .....	235
6.1.6	Funciones sin argumentos .....	235
6.1.7	Panorámica de las funciones musicales disponibles .....	235
6.2	Interfaces para el programador .....	238
6.2.1	Variables de entrada y Scheme .....	238
6.2.2	Representación interna de la música .....	238
6.3	Construcción de funciones complejas .....	238
6.3.1	Presentación de expresiones musicales .....	238
6.3.2	Propiedades de la música .....	238
6.3.3	Doblar una nota con ligaduras (ejemplo) .....	238
6.3.4	Añadir articulación a las notas (ejemplo) .....	238
6.4	Interfaz de marcado para el programador .....	238
6.4.1	Construcción del marcado en Scheme .....	238
6.4.2	Cómo funciona internamente el marcado .....	238
6.4.3	Definición de una instrucción de marcado nueva .....	238
6.4.4	Definición de nuevas instrucciones de lista de marcado .....	238
6.5	Contextos para programadores .....	238
6.5.1	Evaluación de contextos .....	238
6.5.2	Ejecutar una función sobre todos los objetos de la presentación .....	238
6.6	Procedimientos de Scheme como propiedades .....	238
6.7	HACER trasladado al apartado sobre scheme .....	238
6.7.1	Usar código de Scheme en lugar de <code>\tweak</code> .....	238
6.7.2	Trucos difíciles .....	238
<b>Apéndice A</b>	<b>Lista bibliográfica .....</b>	<b>239</b>

<b>Apéndice B</b>	<b>Tablas del manual sobre notación .....</b>	<b>240</b>
B.1	Carta de nombres de acordes .....	240
B.2	Instrumentos MIDI .....	241
B.3	Lista de colores .....	242
B.4	La tipografía Feta .....	244
B.5	Estilos de cabezas de nota .....	257
B.6	Text markup commands .....	258
B.6.1	Font .....	258
B.6.2	Align .....	267
B.6.3	Graphic .....	280
B.6.4	Music .....	284
B.6.5	Instrument Specific Markup .....	288
B.6.6	Other .....	290
B.7	Text markup list commands .....	294
B.8	Lista de articulaciones .....	295
B.9	Todas las propiedades de contexto .....	296
B.10	Propiedades de disposición .....	306
B.11	Identificadores .....	319
B.12	Funciones de Scheme .....	322
<b>Apéndice C</b>	<b>Hoja de referencia rápida .....</b>	<b>340</b>
<b>Apéndice D</b>	<b>Licencia de documentación libre de GNU ....</b>	<b>344</b>
<b>Apéndice E</b>	<b>Índice de instrucciones de LilyPond .....</b>	<b>350</b>
<b>Apéndice F</b>	<b>Índice de LilyPond .....</b>	<b>354</b>

# 1 Notación musical

Este capítulo explica cómo crear notación musical.

## 1.1 Alturas



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
c d e f
g a b c
d e f g
```



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

```
\clef treble
c' c'' e' g
d'' d' d c
\clef bass
```

c, c,, e, g  
d,, d, d c



## Véase también

Glosario musical: [Sección “Nombres de las notas”](#) in *Glosario Musical*.

Fragmentos de código: [Sección “Pitches”](#) in *Fragmentos de código*.

## Escritura de octava relativa

Cuando las octavas se especifican en modo absoluto, es fácil poner por accidente una nota en la octava equivocada. El modo de octavas relativas reduce estos errores porque casi todo el tiempo es innecesario indicar octava alguna. Es más, en el modo absoluto puede ser difícil encontrar un error aislado, mientras que en el modo relativo un solo fallo hace que el resto de la pieza esté una octava más alta o más baja de lo que debería.

```
\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, y se recomienda que sea un Do (una octava de c).

Aquí podemos ver el modo relativo en acción:

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



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

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



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
  <c e g>
  % the C is now an octave higher
  <c' e g'>
  % the C returns to the original pitch
  <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'' {
  c2 fis
  c2 ges
  b2 eisis
  b2 fes
}
```





## Véase también

Glosario musical: [Sección “quinta” in \*Glosario Musical\*](#), [Sección “intervalo” in \*Glosario Musical\*](#), [Sección “Nombres de las notas” in \*Glosario Musical\*](#).

Referencia de la notación: [\[Comprobación de octava\]](#), página 7.

Fragmentos de código: [Sección “Pitches” in \*Fragmentos de código\*](#).

Referencia de funcionamiento interno: `RelativeOctaveMusic`.

## Advertencias y problemas conocidos

La conversión relativa no afecta a las secciones `\transpose`, `\chordmode` o `\relative` en su argumento. Para usar el modo relativo dentro de música transportada, se debe escribir un `\relative` adicional dentro de `\transpose`.

Si no se especifica ninguna *altura\_inicial* para `\relative`, entonces se supone que es `c'`. Sin embargo, ésta es una opción en desuso y podría desaparecer en versiones posteriores, por lo que su utilización está desaconsejada.

## 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 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 “Accidentals and key signatures” in \*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 *doble 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 6.

```
ais1 aes aisis aeses
```



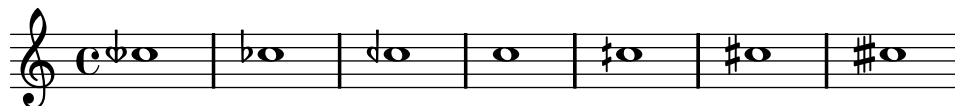
Un becuadro cancelará el efecto de una alteración accidental o de la armadura. Sin embargo, los becuadros no se codifican dentro de la sintaxis del nombre de la nota con un sufijo; una nota becuadro se muestra como un simple nombre de nota natural:

```
a4 aes a2
```



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

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. Estas alteraciones adicionales se pueden usar también para producir notas con becuadro.

cis cis cis! cis? c c? c! c



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

```
cis1 ~ cis ~
\break
cis
```



## Fragmentos de código seleccionados

Según las reglas estándar de composición tipográfica, se imprime un becuadro antes de un sostenido o un bemol cuando se tiene que cancelar una alteración anterior en la misma nota. Para modificar este comportamiento, 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 “sostenido” in *Glosario Musical*, Sección “bemol” in *Glosario Musical*, Sección “doble sostenido” in *Glosario Musical*, Sección “doble bemol” in *Glosario Musical*, Sección “Nombres de las notas” in *Glosario Musical*, Sección “semitono” in *Glosario Musical*.

Manual de aprendizaje: Sección “Accidentals and key signatures” in *Manual de Aprendizaje*.

Referencia de la notación: [Alteraciones accidentales automáticas], página 18, Sección 2.8.1.15 [Alteraciones de la Musica Ficta], página 197, [Nombres de las notas en otros idiomas], página 6.

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

Referencia de funcionamiento interno: `Accidental_engraver`, `Accidental`, `AccidentalCautionary`, `accidental-interface`.

## 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 para algunos idiomas aparte del inglés. Para usarlos, incluya el archivo de inicio específico del idioma. Por ejemplo, para usar los nombres de las notas en español, escriba `\include "espanol.ly"` al principio del archivo de entrada. Los archivos de idioma disponibles y los nombres de las notas que definen son los siguientes:

Language	Note names	sharp	flat	double sharp	double flat
<code>nederlands.ly</code>	c d e f g a bes b	-is	-es	-isis	-eses
<code>english.ly</code>	c d e f g a bf b	-s/-sharp	-f/-flat	-ss/-x/-sharpsharp	-ff/-flatflat
<code>deutsch.ly</code>	c d e f g a b h	-is	-es	-isis	-eses
<code>norsk.ly</code>	c d e f g a b h	-iss/-is	-ess/-es	-ississ/-isis	-essess/-eses
<code>svenska.ly</code>	c d e f g a b h	-iss	-ess	-ississ	-essess
<code>suomi.ly</code>	c d e f g a b h	-is	-es	-isis	-eses
<code>italiano.ly</code>	do re mi fa sol la sib si	-d	-b	-dd	-bb
<code>catalan.ly</code>	do re mi fa sol la sib si	-d/-s	-b	-dd/-ss	-bb
<code>espanol.ly</code>	do re mi fa sol la sib si	-s	-b	-ss	-bb
<code>portugues.ly</code>	do re mi fa sol la sib si	-s	-b	-ss	-bb
<code>vlaams.ly</code>	do re mi fa sol la sib si	-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.

`a2 as e es a ases e eses`



Algunas músicas utilizan microtonos cuyas alteraciones son fracciones de un sostenido o bemol ‘normales’. Los nombres de nota para cuartos de tono que se definen en los distintos archivos de idioma se encuentran listados en la tabla siguiente. Aquí los prefijos ‘semi-’ y ‘sesqui-’ significan ‘medio’ y ‘uno y medio’ respectivamente. Para el noruego, sueco, catalán y español no se ha definido aún ningún nombre especial.

Idioma	Nombre de la nota	semi-sharp	semi-flat	sesqui-sharp	sesqui-flat
nederlands.ly	c d e f g a bes b	-ih	-eh	-isih	-eseh
english.ly	c d e f g a bf b	-qs	-qf	-tqs	-tqf
deutsch.ly	c d e f g a b h	-ih	-eh	-isih	-eseh
italiano.ly	do re mi fa sol la sib si	-sd	-sb	-dsd	-bsb
portugues.ly	do re mi fa sol la sib si	-sqt	-bqt	-stqt	-btqt

## Véase también

Glosario musical: [Sección “Nombres de las notas” in \*Glosario Musical\*](#).

Fragmentos de código: [Sección “Pitches” in \*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'' {
  c2 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 falla se imprime un mensaje de advertencia, pero la nota previa no se modifica. Las notas posteriores están en relación a la `altura_de_control`.

```
\relative c'' {
  c2 d
  \octaveCheck c'
```

```
e2 f
}
```



Compare los dos compases siguientes. La primera y tercera comprobaciones de `\octaveCheck` fracasan, pero el segundo es correcto.

```
\relative c'' {
  c4 f g f

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



Véase también

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

Referencia de funcionamiento interno: RelativeOctaveCheck.

## 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.

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 c' {
    \key d \major
    d4 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 c' {
    \key c \major
    c4 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 transportarán un semitono hacia arriba. La primera versión imprimirá sostenidos y las notas no se moverán de su lugar en la escala, la segunda imprimirá bemoles de la nota siguiente.

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



`\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 17.

## Fragmentos de código seleccionados

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 (eq? n 6) (eq? n 2)))
       (set! a (- a 2))
       (set! n (+ n 1)))
      ((and (< a -1) (or (eq? n 0) (eq? 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 musica)
  (let* ((es (ly:music-property musica 'elements))
        (e (ly:music-property musica 'element))
        (p (ly:music-property musica 'pitch)))
    (if (pair? es)
        (ly:music-set-property!
         musica 'elements
         (map (lambda (x) (naturalize x)) es)))
    (if (ly:music? e)
        (ly:music-set-property!
         musica 'element
         (naturalize e)))
    (if (ly:pitch? p)
        (begin
          (set! p (naturalize-pitch p))
          (ly:music-set-property! musica 'pitch p)))
    musica))

naturalizeMusic =
#(define-music-function (parser location m)
  (ly:music?)
```

```

(naturalize m))

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

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

```



## Véase también

Referencia de la notación: [\[Transposición de los instrumentos\]](#), página 17.

Fragmentos de código: [Sección “Pitches”](#) in *Fragmentos de código*.

Referencia de funcionamiento interno: `TransposedMusic`.

## 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`.

### 1.1.3 Imprimir las alturas

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

#### Clave

La clave se establece con la instrucción `\clef nombre_de_clave`. En todos los ejemplos se muestra el Do central.

```

\clef treble
c2 c
\clef alto
c2 c
\clef tenor
c2 c
\clef bass
c2 c

```





Otras claves son las siguientes:

```
\clef french
c2 c
\clef soprano
c2 c
\clef mezzosoprano
c2 c
\clef baritone
c2 c
```

```
\break
```

```
\clef varbaritone
c2 c
\clef subbass
c2 c
\clef percussion
c2 c
\clef tab
c2 c
```



Están contempladas claves adicionales que se describen bajo [Sección 2.8.1.4 \[Claves antiguas\]](#), [página 182](#).

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. El argumento *nombre\_de\_clave* se debe encerrar entre comillas cuando contiene guiones bajos o dígitos.

```
\clef "treble_8"
c2 c
\clef "bass^15"
c2 c
```



## Fragmentos de código seleccionados

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

Observe que la modificación del glifo, la posición de la clave o su octavación, no cambian 'per se' la posición de las siguientes notas del pentagrama: para hacer esto también se debe especificar la posición del Do central. Los parámetros posicionales están en relación con la tercera línea del pentagrama, los números positivos desplazan hacia arriba, contando una unidad por cada línea y espacio. El valor de `clefOctavation` se establecería normalmente a 7, -7, 15 or -15, pero son válidos otros valores.

Cuando se produce un cambio de clave en el salto de línea se imprime la clave nueva tanto al final de la línea anterior como al principio de la nueva, de forma predeterminada. Si no se necesita la clave de advertencia al final de la línea anterior, se puede quitar estableciendo el valor de la propiedad `explicitClefVisibility` de `Staff`, a `end-of-line-invisible`. El comportamiento predeterminado se puede recuperar con `\unset Staff.explicitClefVisibility`.

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
  c'1
  % The standard bass clef
  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  \set Staff.middleCPosition = #6
  c'1
  % The baritone clef
  \set Staff.clefGlyph = #"clefs.C"
  \set Staff.clefPosition = #4
  \set Staff.middleCPosition = #4
  c'1
  % The standard choral tenor clef
  \set Staff.clefGlyph = #"clefs.G"
  \set Staff.clefPosition = #-2
  \set Staff.clefOctavation = #-7
  \set Staff.middleCPosition = #1
  c'1
  % A non-standard clef
  \set Staff.clefPosition = #0
  \set Staff.clefOctavation = #0
  \set Staff.middleCPosition = #-4
  c'1 \break

  % The following clef changes do not preserve
  % the normal relationship between notes 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.clefOctavation = #7
  c'1
```

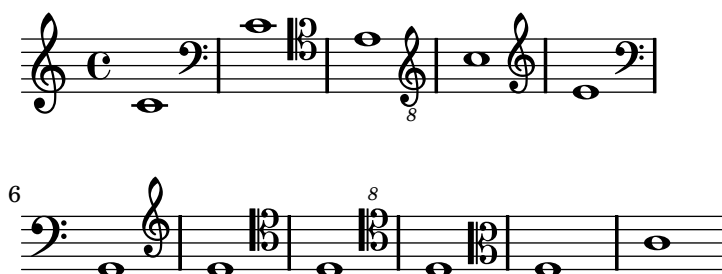
```

\set Staff.clefOctavation = #0
\set Staff.clefPosition = #0
c'1

% Here we go back to the normal clef:

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

```



## Véase también

Referencia de la notación: [Sección 2.8.1.4 \[Claves antiguas\]](#), página 182.

Fragmentos de código: [Sección “Pitches” in Fragmentos de código](#).

Referencia de funcionamiento interno: `Clef_engraver`, `Clef`, `OctavateEight`, `clef-interface`.

## Armadura de la tonalidad

Las alteraciones accidentales y las armaduras son una frecuente causa de confusión para los nuevos usuarios, porque las notas sin alteración pueden llevar signos de becuadro en función de la armadura. Para ver más información, consulte [\[Alteraciones accidentales\]](#), página 4 o [Sección “Accidentals and key signatures” in Manual de Aprendizaje](#).

**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 “Accidentals and key signatures” in 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 tipo
```

Aquí, *tipo* 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).

```
\key g \major
fis1
f
fis
```



## Fragmentos de código seleccionados

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
}
```



La muy utilizada instrucción `\key` establece la propiedad `keySignature` property, 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.keySignature = #`(((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. (observe la coma precedente.)

Alternativamente, para cada elemento de la lista el uso del formato más conciso (`paso . alteración`) especifica que la misma alteración debe estar en todas las octavas.

He aquí un ejemplo de una posible armadura para generar una escala exátona:

```
\relative c' {
  \set Staff.keySignature = #`(((0 . 3) . ,SHARP) ((0 . 5) . ,FLAT) ((0 . 6) . ,FLAT))
  c4 d e fis
  aes4 bes c2
}
```



## Véase también

Glosario musical: [Sección “modo eclesiástico” in \*Glosario Musical\*](#), [Sección “scordatura” in \*Glosario Musical\*](#).

Manual de aprendizaje: [Sección “Accidentals and key signatures” in \*Manual de Aprendizaje\*](#).

Fragmentos de código: [Sección “Pitches” in \*Fragmentos de código\*](#).

Referencia de funcionamiento interno: `KeyChangeEvent`, `Key_engraver`, `Key_performer`, `KeyCancellation`, `KeySignature`, `key-cancellation-interface`, `key-signature-interface`.

## Corchetes de octava

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

```
a'2 b
\ottava #1
a b
\ottava #0
a b
```



La función `ottava` también admite los valores -1 (para 8va bassa), 2 (para 15ma) y -2 (para 15ma bassa) como argumentos.

## Fragmentos de código seleccionados

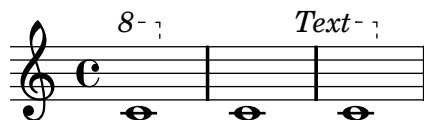
Internamente, la función `set-octavation` establece las propiedades `ottavation` (por ejemplo, a "8va" o a "8vb") y `middleCPosition`. Para sobrescribir el texto del corchete, ajuste `ottavation` despues de invocar a `set-octavation`.

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

```

\ottava #1
\set Staff.ottavation = #"Text"
c''1
}

```



## Véase también

Glosario musical: [Sección “octavación” in \*Glosario Musical\*](#).

Fragmentos de código: [Sección “Pitches” in \*Fragmentos de código\*](#).

Referencia de funcionamiento interno: `Ottava_spanner_engraver`, `OttavaBracket`, `ottava-bracket-interface`.

## 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 105.

```
\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 = "Vln" {
    \relative c'' {
      \set Staff.instrumentName = "Vln"
      \set Staff.midiInstrument = "violin"
      % strictly speaking not necessary, but a good reminder
      \transposition c'

      \key c \major
      g4( c8) r c r c4
    }
  }
  \new Staff = "clarinet" {
    \relative c'' {
      \set Staff.instrumentName = \markup { Cl (B\flat) }
      \set Staff.midiInstrument = "clarinet"

```

```

\transposition bes

% not concert pitch
\key d \major
a4( d8) r d r d4
}
}
>>

```



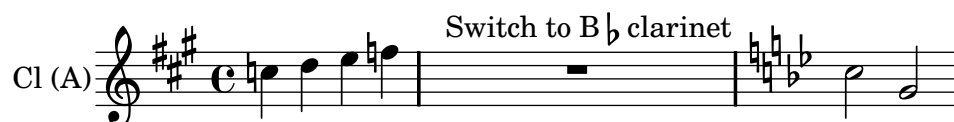
La `\transposition` se puede cambiar durante la pieza. Por ejemplo, un clarinetista puede cambiar del clarinete en La al clarinete en Si bemol.

```

\set Staff.instrumentName = "Cl (A)"
\key a \major
\transposition a
c d e f
\textLengthOn
s1*0^\markup { Switch to B\flat clarinet }
R1

\key bes \major
\transposition bes
c2 g

```



## Véase también

Glosario musical: Sección “altura de concierto” in *Glosario Musical*, Sección “instrumento transpositor” in *Glosario Musical*.

Referencia de la notación: [Citar otras voces], página 105, [Transposición], página 8.

Fragmentos de código: Sección “Pitches” in *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 <<
  #(set-accidental-style 'voice)
  { ... }

```

&gt;&gt;

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

```

#(set-accidental-style 'voice 'StaffGroup)

```

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

```

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

musicaB = {
  \clef bass
  \new Voice {
    \voiceTwo \relative c' {
      <fis, a cis>4
      \change Staff = up
      cis'
      \change Staff = down
      <fis, a>
      \change Staff = up
      dis' |
      \change Staff = down
      <fis, a cis>4 gis <f a d>2 |
    }
  }
}

\new PianoStaff {
  <<
    \context Staff = "up" {
      %% change the next line as desired:
      #(set-accidental-style 'default)
      \musicaA
    }
    \context Staff = "down" {
      %% change the next line as desired:
      #(set-accidental-style 'default)
      \musicaB
    }
  >>
}

```





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.

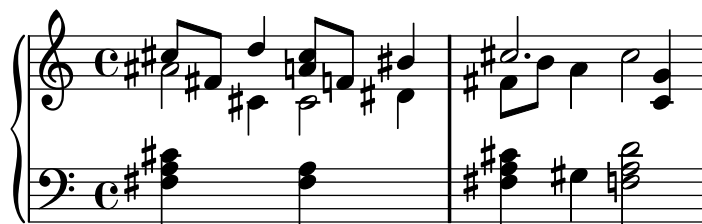
```

\new PianoStaff {
  <<
    \context Staff = "up" {
      %% cambie la línea siguiente como desee:
      #(set-accidental-style 'default 'Score)
      \musicA
    }
    \context 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. Imprime las mismas alteraciones que el estilo **default**, con dos excepciones 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’ (las que no se imprimen en el estilo **default**) se imprimen como alteraciones de precaución. Se imprimen de forma predeterminada con paréntesis, pero también se pueden imprimir en tamaño reducido definiendo la propiedad **cautionary-style** de **AccidentalSuggestion**.

**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:



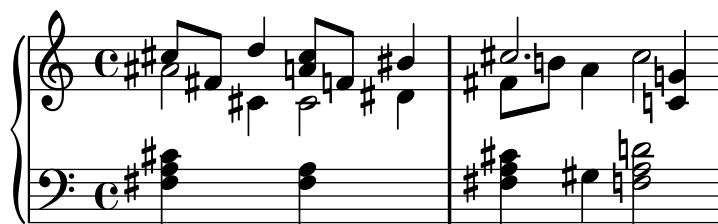
**modern-voice-cautionary** (moderno, voz, de precaución)

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.

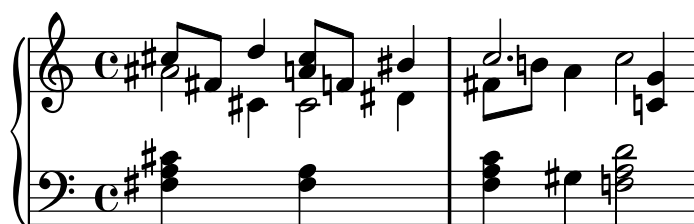
Este estilo de alteración se aplica de manera predeterminada al grupo **GrandStaff** o **PianoStaff** en curso.

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

Igual que `#(set-accidental-style 'piano)` pero con las alteraciones añadidas compuestas como de precaución.

**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:



## Fragmentos de código seleccionados

En las obras de principios del s.XX, empezando por Schoenberg, Berg y Webern (la "Segunda" escuela de Viena), cada nota de la escala de doce tonos se debe tratar con igualdad, sin ninguna jerarquía como los grados clásicos tonales. Por tanto, estos compositores imprimen una alteración accidental para cada nota, incluso en las notas naturales, para enfatizar su nuevo enfoque de la teoría y el lenguaje musicales.

Este fragmento de código muestra cómo conseguir dichas reglas de notación.

```
webernAccidentals = {
  % the 5s are just "a value different from any accidental"
  \set Staff.keySignature = #'((0 . 5) (1 . 5) (2 . 5) (3 . 5)
                                (4 . 5) (5 . 5) (6 . 5))

  \set Staff.extraNatural = ##f
  #(set-accidental-style 'forget)
}

\score {
  {
    \webernAccidentals
    c'4 dis' cis' cis'
    c'4 dis' cis' cis'
    c'4 c' dis' des'
  }
  \layout {
    \context {
      \Staff
      \remove "Key_engraver"
    }
  }
}
```



## Véase también

Fragmentos de código: [Sección “Pitches” in \*Fragmentos de código\*](#).

Referencia de funcionamiento interno: `Accidental`, `Accidental_engraver`, `GrandStaff` and `PianoStaff`, `Staff`, `AccidentalSuggestion`, `AccidentalPlacement`, `accidental-suggestion-interface`.

## Advertencias y problemas conocidos

Las notas simultáneas se consideran introducidas en modo secuencial. Esto significa que dentro de un acorde las alteraciones se tipografían como si las notas del acorde se sucedieran una a una en el tiempo, en el orden en que aparecen en el archivo de entrada. Esto plantea un problema cuando las alteraciones de un acorde dependen unas de otras, lo que no se da en el estilo de alteraciones predeterminado. El problema se puede resolver insertando manualmente `!` y `?` para las notas problemáticas.

## 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 c'' {
  aes c e2
  cis,1
}
```



## Fragmentos de código seleccionados

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
}
>>

```



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
}
>>

```



## Véase también

Glosario musical: [Sección “tesitura” in \*Glosario Musical\*](#).

Fragmentos de código: [Sección “Pitches” in \*Fragmentos de código\*](#).

Referencia de funcionamiento interno: `Ambitus_engraver`, `Voice`, `Staff`, `Ambitus`, `AmbitusAccidental`, `AmbitusLine`, `AmbitusNoteHead`, `ambitus-interface`.

## Advertencias y problemas conocidos

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

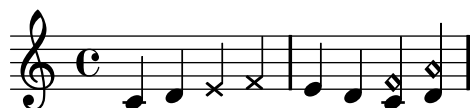
### 1.1.4 Cabeza de las notas

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

#### Cabezas de nota especiales

Varios instrumentos usan distintas formas para la cabeza de las notas con distintos significados: las aspas se usan para el ‘parlato’ de los vocalistas, las notas apagadas en la guitarra; los romps se usan para los armónicos de los instrumentos de arco, etc. Hay una abreviatura (`\harmonic`) para las formas en rombo; los otros estilos de cabeza se producen trucando la propiedad:

```
c4 d
\override NoteHead #'style = #'cross
e f
\revert NoteHead #'style
e d <c f\harmonic> <d a'\harmonic>
```



Para ver todos los estilos de cabeza de las notas, consulte [Sección B.5 \[Estilos de cabezas de nota\]](#), página 257.

## Véase también

Fragmentos de código: [Sección “Pitches” in Fragmentos de código](#).

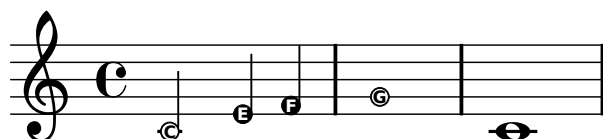
Referencia de la notación: [Sección B.5 \[Estilos de cabezas de nota\]](#), página 257.

Referencia de funcionamiento interno: `note-event`, `Note_heads_engraver`, `Ledger_line_engraver`, `NoteHead`, `LedgerLineSpanner`, `note-head-interface`, `ledger-line-spanner-interface`.

## Cabezas de notas de Notación Fácil

Las cabezas ‘easy play’ (fáciles de tocar) tienen el nombre de la nota (en inglés) dentro de la cabeza. Se usa en la música para principiantes.

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



La instrucción `\easyHeadsOn` sobrescribe los ajustes del objeto `NoteHead`. Estos ajustes se pueden deshacer mediante la instrucción `\easyHeadsOff`. Para que las letras sean legibles, se tienen que imprimir en un tamaño grande de la tipografía. Para imprimir con un tamaño de tipografía mayor, consulte [Sección 4.2.1 \[Establecer el tamaño del pentagrama\]](#), página 211.

## Instrucciones predefinidas

`\easyHeadsOn`, `\easyHeadsOff`

## Véase también

Referencia de la notación: [Sección 4.2.1 \[Establecer el tamaño del pentagrama\]](#), página 211.

Fragmentos de código: [Sección “Pitches” in \*Fragmentos de código\*](#).

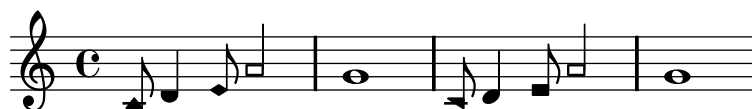
Referencia de funcionamiento interno: `note-event`, `Note_heads_engraver`, `NoteHead`, `note-head-interface`.

## 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 con formas se pueden producir estableciendo `\aikenHeads` o `\sacredHarpHeads`, según el estilo deseado.

```
\aikenHeads
c8 d4 e8 a2
g1
\sacredHarpHeads
c,8 d4 e8 a2
g1
```



Las formas se determinan en función del grado de la escala, donde la tónica está determinada por la instrucción `\key`.

## Fragmentos de código seleccionados

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`, espas, y `xcircle`, círculo con espas) 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
}

\score {
  \new Staff {
    \transpose c d
    \relative c' {
      \set shapeNoteStyles = #'#(do re mi fa #f la ti)
      \fragment
    }

    \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 B.5 \[Estilos de cabezas de nota\]](#), página 257.

## Véase también

Fragmentos de código: [Sección “Pitches” in Fragmentos de código](#).

Referencia de la notación: [Sección B.5 \[Estilos de cabezas de nota\]](#), página 257.

Referencia de funcionamiento interno: `note-event`, `Note_heads_engraver`, `NoteHead`, `note-head-interface`.

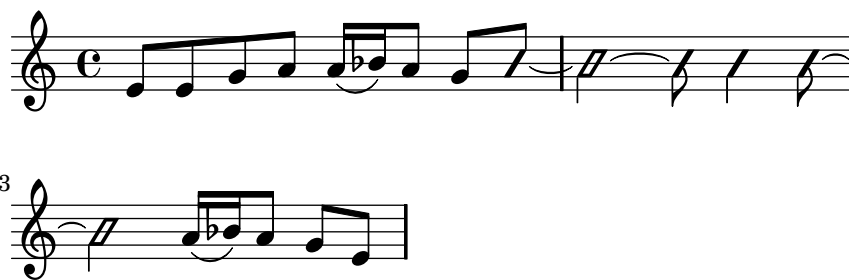
## Improvisación

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"
} \transpose c c' {
  e8 e g a a16( bes) a8 g
  \improvisationOn
  e8 ~
  e2 ~ e8 f4 f8 ~
  f2
  \improvisationOff
  a16( bes) a8 g e
}

```



## Instrucciones predefinidas

`\improvisationOn`, `\improvisationOff`

## Véase también

Fragmentos de código: [Sección “Pitches” in Fragmentos de código.](#)

Referencia de funcionamiento interno: `Pitch_squash_engraver`, `Voice`, `RhythmicStaff`.

## 1.2 Duraciones

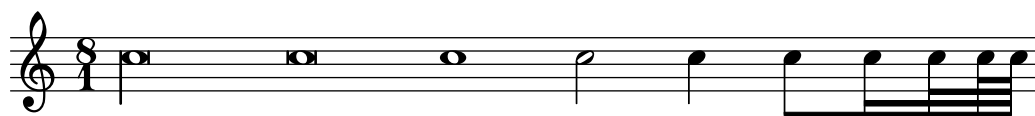
Esta sección trata de los ritmos, los silencios, las duraciones, las barras y los compases.

## 1.2.1 Escritura de las duraciones (valores rítmicos)

### 1.2.1.1 Duraciones

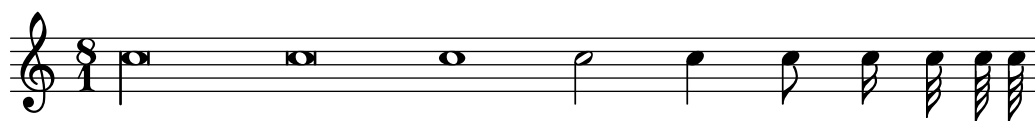
Las duraciones se designan mediante números y puntos. Las duraciones se introducen como sus valores recíprocos respecto de la redonda. Por ejemplo, una negra se escribe usando un 4 (puesto que es 1/4 de redonda), mientras que una blanca se escribe con un 2 (por ser 1/2 de redonda). Para notas mayores de la redonda se deben usar los comandos `\longa` (que es una breve doble) y `\breve`. Se pueden especificar duraciones tan cortas como la semifusa (con el número 64). Son posibles valores más cortos, pero sólo como notas unidas por una barra.

```
\time 8/1
c\longa c\breve c1 c2
c4 c8 c16 c32 c64 c64
```



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

```
\time 8/1
\autoBeamOff
c\longa c\breve c1 c2
c4 c8 c16 c32 c64 c64
```



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 2.8 \[Notación antigua\]](#), página 180.

Si la duración se omite, su valor será el de la duración de la nota anterior. Por omisión, el valor de la primera nota es el de una negra.

```
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.

```
a4 b c4. b8 a4. b4.. c8.
```



Ciertas duraciones no se pueden representar sólo con duraciones binarias y puntillos; la única forma de representarlas es ligando dos o más notas. Para ver más detalles, consulte [Sección 1.2.1.4 \[Ligaduras de unión\]](#), página 35.

Para ver de qué manera se pueden especificar las duraciones de las sílabas de la letra y cómo alinear la letra a las notas, consulte [Sección 2.1 \[Música vocal\]](#), página 130.

Opcionalmente se pueden espaciar las notas de forma estrictamente proporcional a sus duraciones. Para ver más detalles sobre este y otros ajustes que controlan la notación proporcional, consulte [Sección 4.5.5 \[Notación proporcional\]](#), página 212.

Normalmente los puntillos se desplazan hacia arriba para evitar las líneas del pentagrama, excepto en situaciones de polifonía. Para forzar una dirección concreta de forma manual, véase [Sección 5.4.2 \[Dirección y posición\]](#), página 230.

## Instrucciones predefinidas

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

## Véase también

Glosario musical: [Sección “breve” in \*Glosario Musical\*](#), [Sección “longa” in \*Glosario Musical\*](#), [Sección “figura” in \*Glosario Musical\*](#), [Sección “Nombre de las duraciones de notas y silencios” in \*Glosario Musical\*](#).

Referencia de la notación: [Sección 1.2.4.1 \[Barras automáticas\]](#), página 47, [Sección 1.2.1.4 \[Ligaduras de unión\]](#), página 35, [Sección 1.2.1 \[Escritura de las duraciones \(valores rítmicos\)\]](#), página 30, [Sección 1.2.2 \[Escritura de silencios\]](#), página 37, [Sección 2.1 \[Música vocal\]](#), página 130, [Sección 2.8 \[Notación antigua\]](#), página 180, [Sección 4.5.5 \[Notación proporcional\]](#), página 212.

Fragmentos de código: [Sección “Rhythms” in \*Fragmentos de código\*](#).

Referencia de funcionamiento interno: `Dots`, `DotColumn`.

## Advertencias y problemas conocidos

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 garrapear (128) hasta el de máxima (8 redondas).

### 1.2.1.2 Grupos especiales

Los grupos especiales se obtienen a partir de una expresión musical multiplicando todas las duraciones por una fracción:

```
\times fracción { música }
```

La duración de *música* se multiplicará por la fracción. El denominador de la fracción se imprimirá sobre las notas, opcionalmente con un corchete. El grupo especial más común es el tresillo, en el que 3 notas tienen el valor de 2, por tanto las notas duran 2/3 de su valor escrito

```
a2 \times 2/3 {b4 b b}
```

```
c4 c \times 2/3 {b4 a g}
```



La colocación automática del corchete de grupo especial por encima o por debajo de las notas se puede sobrescribir manualmente mediante instrucciones predefinidas, para ver más detalles consulte [Sección 5.4.2 \[Dirección y posición\]](#), página 230.

Los grupos pueden anidarse unos dentro de otros:

```
\autoBeamOff
c4 \times 4/5 { f8 e f \times 2/3 { e[ f g] } } f4 |
```



La modificación de los grupos especiales anidados que comienzan en el mismo instante musical se debe hacer con `\tweak`.

Para modificar la duración de las notas sin imprimir un corchete de grupo especial, véase [Sección 1.2.1.3 \[Escalar las duraciones\]](#), página 34.

## Instrucciones predefinidas

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

## Fragmentos de código seleccionados

*Escribir varios grupos especiales usando una sola instrucción `imes`*

La propiedad `tupletSpannerDuration` establece cuánto debe durar cada grupo de valoración especial contenido dentro del corchete que aparece después de `imes`. Así, se pueden escribir muchos tresillos seguidos dentro de una sola expresión `imes`, ahorrando trabajo de teclado.

En el ejemplo se muestran dos tresillos, aunque se ha escrito `imes` una sola vez.

Para ver más información sobre `make-moment`, véase "Administración del tiempo".

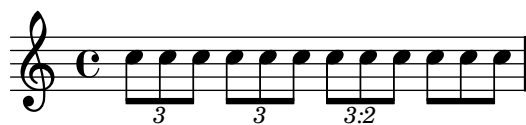
```
\relative c' {
  \time 2/4
  \set tupletSpannerDuration = #(ly:make-moment 1 4)
  \times 2/3 { c8 c c c c c }
}
```



*Cambiar el número del grupo especial*

De forma predeterminada sólo se imprime el numerador del grupo especial sobre el corchete de grupo, es decir, el denominador del argumento de la instrucción `imes`. De forma alternativa, se puede imprimir un quebrado en la forma numerador:denominador del número del grupo, o eliminar el número.

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



*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 8 { \times 2/3 { c[ b a] } }
  % Insert a manual line break within a triplet
  \times 2/3 { c[ b \bar "" \break a] }
  \repeat unfold 2 { \times 2/3 { c[ b a] } }
  c8
}
```



## Véase también

Glosario musical: Sección “tresillo” in *Glosario Musical*, Sección “grupo especial” in *Glosario Musical*, Sección “polimétrica” in *Glosario Musical*.

Manual de aprendizaje: Sección “Tweaking methods” in *Manual de Aprendizaje*.

Referencia de la notación: Sección 1.2.6.3 [Gestión del tiempo], página 63, Sección 1.2.1.3 [Escalar las duraciones], página 34, Sección 5.3.5 [La instrucción tweak], página 227, Sección 1.2.3.4 [Notación polimétrica], página 46.

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

Referencia de funcionamiento interno: `TupletBracket`, `TupletNumber`, `TimeScaledMusic`.

## Advertencias y problemas conocidos

Cuando la primera nota de un pentagrama es una nota de adorno (un mordente de una nota) seguida de un grupo de valoración especial, la nota de adorno se debe situar antes de la instrucción `\times` para evitar errores. En cualquier otro lugar, las notas de adorno sí se pueden colocar dentro de los corchetes de grupo especial.

### 1.2.1.3 Escalar las duraciones

Es posible alterar la duración de las figuras, silencios o acordes en un factor  $N/M$  añadiendo `*N/M` (o `*N` si  $M$  es 1) a la duración. Esto 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`.

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

```
\time 2/4
% Alterar la duración de los tresillos
a4*2/3 gis4*2/3 a4*2/3
% Duraciones normales
a4 a4
% Duplicar la duración del acorde
<a d>4*2
% Duration of quarter, appears like sixteenth
b16*4 c4
```



La duración de las notas de salto o espaciadoras 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. La sintaxis general de la instrucción que lo hace es:

```
\scaleDurations #'(numerador . denominador) { música }
```

Esto dejará intacta la apariencia de la *música*, pero la duración interna de las notas se multiplicará por la fracción  $numerador/denominador$ . Los espacios que rodean el punto son necesarios. He aquí un ejemplo que muestra cómo se puede comprimir y expandir la música:

```
\time 2/4
% Duraciones normales
<c a>4 c8 a
% Escalar la música en un factor 2/3
\scaleDurations #'(2 . 3) {
  <c a f>4. c8 a f
}
% Escalar la música en un factor de 2
\scaleDurations #'(2 . 1) {
  <c' a>4 c8 b
}
```



Una aplicación de esta instrucción se encuentra en la notación polimétrica, véase [Sección 1.2.3.4 \[Notación polimétrica\]](#), página 46.

## Véase también

Referencia de la notación: [Sección 1.2.1.2 \[Grupos especiales\]](#), página 31, [Sección 1.2.2.2 \[Silencios invisibles\]](#), página 39, [Sección 1.2.3.4 \[Notación polimétrica\]](#), página 46.

Fragmentos de código: [Sección “Rhythms” in Fragmentos de código](#).

### 1.2.1.4 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 usando el símbolo de la tilde curva ~

e' ~ e'



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:



Si necesita ligar muchas notas a través de las líneas divisorias, podría resultarle más fácil utilizar la división automática de las notas, véase [Sección 1.2.3.5 \[División automática de las notas\]](#), página 46. 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 la ligadura dentro del acorde.

<c e g> ~ <c e g>  
<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:

```
\repeat volta 2 { c g <c e>2 ~ }
\alternative {
  % Primera alternativa: la nota siguiente se liga normalmente
  { <c e>2. r4 }
  % Segunda alternativa, la nota siguiente lleva una ligadura repetida
  {<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>\laissezVibrer
```



Es posible controlar la colocación vertical de las ligaduras de unión; véase la sección de instrucciones predefinidas, o para ver más detalles, consulte [Sección 5.4.2 \[Dirección y posición\]](#), [página 230](#).

Se pueden especificar ligaduras continuas, de puntos o intermitentes, véase la sección de instrucciones predefinidas.

## Instrucciones predefinidas

```
\tieUp, \tieDown, \tieNeutral, \tieDotted, \tieDashed, \tieSolid, \repeatTie.
```

## 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 "verdadero". 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, como se muestra en este ejemplo.

```
\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 g~ 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).

```
\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> ~ <c e g>
}
```



## Véase también

Glosario musical: [Sección “ligadura de unión” in \*Glosario Musical\*](#), [Sección “laissez vibrer” in \*Glosario Musical\*](#).

Referencia de la notación: [Sección 1.2.3.5 \[División automática de las notas\]](#), página 46.

Fragmentos de código: [Sección “Rhythms” in \*Fragmentos de código\*](#).

Referencia de funcionamiento interno: `LaissezVibrerTie`, `LaissezVibrerTieColumn`, `TieColumn`, `Tie`.

## Advertencias y problemas conocidos

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

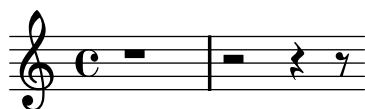
Los cambios de clave o de octava durante una ligadura de unión no están bien definidos realmente. En estos casos puede ser preferible una ligadura de expresión.

### 1.2.2 Escritura de silencios

#### 1.2.2.1 Silencios

Los silencios se introducen como notas con el nombre `r`

`r1 r2 r4 r8`



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 [Sección 1.2.2.3 \[Silencios de compás completo\]](#), página 40.

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:

```
a'4\rest d'4\rest
```



Esto hace mucho más sencillo el formateo de la música polifónica, porque el formateador automático de colisiones de silencios se olvidará de ellos.

## Fragmentos de código seleccionados

### *Estilos de silencios*

Los silencios se pueden imprimir en distintos estilos.

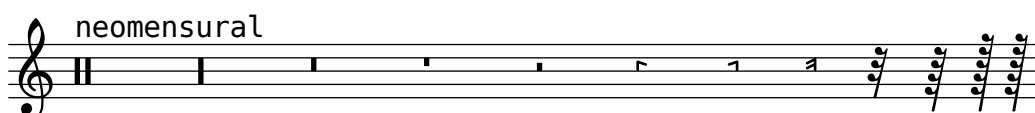
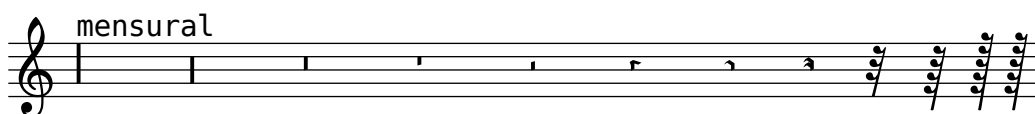
```
\layout {
  indent = 0.0
  \context {
    \Staff
    \remove "Time_signature_engraver"
  }
}

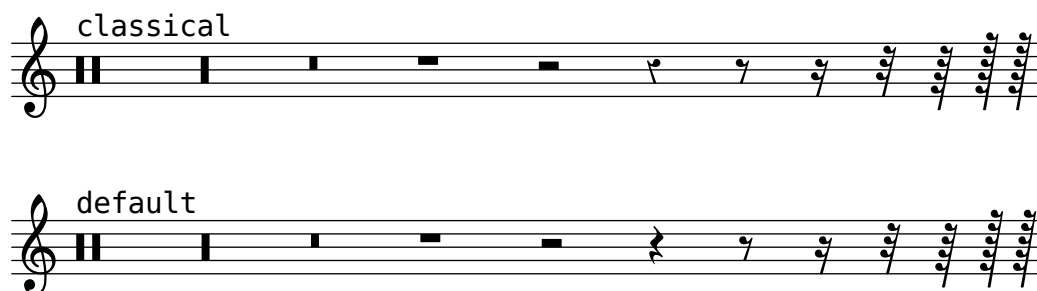
\relative c {
  \set Score.timing = ##f
  \override Staff.Rest #'style = #'mensural
  r\maxima^\markup \typewriter { mensural }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 r128
  \bar ""

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

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

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





## Véase también

Referencia de la notación: [Sección 1.2.2.3 \[Silencios de compás completo\]](#), página 40.

Fragmentos de código: [Sección “Rhythms” in Fragmentos de código](#).

Referencia de funcionamiento interno: `Rest`.

## 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 la garrapatea (128) hasta la máxima (8 redondas).

### 1.2.2.2 Silencios invisibles

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

```
a4 a4 s4 a4 \skip 1 a4
```



La sintaxis de `s` sólo está disponible en el modo de notas y en el modo de acordes. En otras situaciones, por ejemplo, cuando se introduce la letra, debe usar el comando `\skip`

```
<<
  \relative { a'2 a2 }
  \new Lyrics \lyricmode { \skip 2 bla2 }
>>
```



bla

El comando `skip` es un mero contenedor musical vacío. No produce ninguna salida, ni siquiera una salida transparente.

El comando de desplazamiento `s` crea una `Staff` y una `Voice` cuando es necesario, de forma similar a los comandos de nota y de silencio. Por ejemplo, el fragmento siguiente produce un pentagrama vacío.

```
{ s4 }
```



El fragmento `{ \skip 4 }` produciría una página vacía.

## Véase también

Referencia de funcionamiento interno: `SkipMusic`.

### 1.2.2.3 Silencios de compás completo

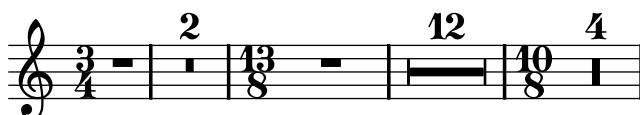
Los silencios de un compás completo (o muchos compases) se escriben como `R`. Esto está indicado específicamente para silencios de compás completo y para escribir particellas: el silencio se puede expandir de forma que rellene una partitura con silencios, o se puede imprimir como un solo silencio multicompás. Esta expansión se controla por medio de la propiedad `Score.skipBars`. Si está establecido al valor `true`, los compases vacíos no se expandirán, y se añadirá automáticamente el número correspondiente

```
\time 4/4 r1 | R1 | R1*2 \time 3/4 R2. \time 2/4 R2 \time 4/4
\set Score.skipBars = ##t R1*17 R1*4
```



El número 1 de `R1` es similar a la notación de la duración que se usa para las notas. De aquí que para indicaciones de compás distintas al 4/4, tengamos que escribir otras duraciones. Esto se puede hacer con puntillos de prolongación o con fracciones

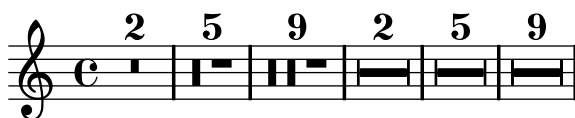
```
\set Score.skipBars = ##t
\time 3/4
R2. | R2.*2
\time 13/8
R1*13/8
R1*13/8*12 |
\time 10/8 R4*5*4 |
```



Una `R` que dura un solo compás se imprime como un silencio de redonda o breve, centrado en el compás independientemente de la indicación de compás.

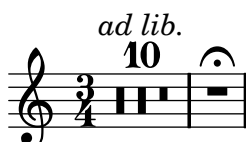
Si hay sólo algunos compases de silencio, LilyPond imprime ‘silencios eclesiásticos’ (un conjunto de pequeños rectángulos) en el pentagrama. Para sustituirlos con silencios ordinarios, utilice `MultiMeasureRest.expand-limit`.

```
\set Score.skipBars = ##t
R1*2 | R1*5 | R1*9
\override MultiMeasureRest #'expand-limit = 1
R1*2 | R1*5 | R1*9
```



Se pueden añadir textos a los silencios multicompas utilizando la sintaxis *nota-marca* de [Sección 1.8.2 \[Formatear el texto\]](#), página 125. Hay una variable (`\fermataMarkup`) para añadir calderones

```
\set Score.skipBars = ##t
\time 3/4
R2.*10^\markup { \italic "ad lib." }
R2.^ \fermataMarkup
```



¡Advertencia! Este texto se crea por parte de `MultiMeasureRestText`, no de `TextScript`.

```
\override TextScript #'padding = #5
R1^"low"
\override MultiMeasureRestText #'padding = #5
R1^"high"
```

high



Si queremos poner texto a la izquierda de un silencio multicompas, adjuntamos el texto a una nota de salto de longitud cero, esto es

```
s1*0^"Allegro"
R1*4
```

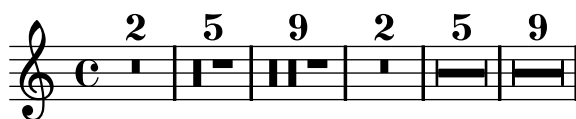
## Fragmentos de código seleccionados

*Cambiar la forma de los silencios multicompas*

Si hay diez compases de silencio o menos, se imprime en el pentagrama una serie de silencios de breve y larga (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'' {
  \compressFullBarRests
  R1*2 | R1*5 | R1*9
  \override MultiMeasureRest #'expand-limit = 3
```

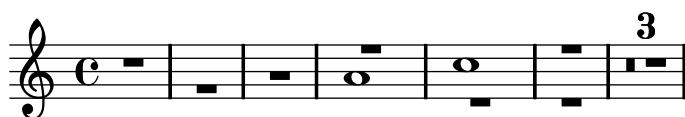
```
R1*2 | R1*5 | R1*9
}
```



### *Posicionar los silencios multicomps*

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

```
\relative c' {
  % Multi-measure rests by default are set under the second line
  R1
  % They can be moved with an override
  \override MultiMeasureRest #'staff-position = #-2
  R1
  % A value of 0 is the default position;
  % the following trick moves the rest to the center line
  \override MultiMeasureRest #'staff-position = #-0.01
  R1
  % Multi-measure rests in odd-numbered voices are under the top line
  << { R1 } \ { a1 } >>
  % Multi-measure rests in even-numbered voices are under the bottom line
  << { c1 } \ { R1 } >>
  % They remain separated even in empty measures
  << { R1 } \ { R1 } >>
  % This brings them together even though there are two voices
  \compressFullBarRests
  <<
    \revert MultiMeasureRest #'staff-position
    { R1*3 }
    \
    \revert MultiMeasureRest #'staff-position
    { R1*3 }
  >>
}
```



### Véase también

Referencia de funcionamiento interno: `MultiMeasureRest`.

El objeto de presentación `MultiMeasureRestNumber` es para el número por defecto, y `MultiMeasureRestText` para los textos definidos por el usuario.

## Advertencias y problemas conocidos

No se pueden usar digitaciones (p.ej. R1-4) para poner números encima de los silencios multicompás. Y la altura de los silencios multicompás (o silencios centrados en el pentagrama) no se puede modificar.

No hay ninguna forma de condensar automáticamente muchos silencios en un solo silencio multicompás. Los silencios multicompás no toman parte en las comprobaciones de colisión de silencios.

Tenga cuidado al escribir silencios multicompás seguidos por redondas. El código siguiente introduce dos notas que tienen una duración de cuatro compases cada una

```
R1*4 cis cis
```

Cuando `skipBars` esté establecido, el aspecto del resultado será correcto, pero la numeración de los compases estará desactivada.

### 1.2.3 Impresión de las duraciones

#### 1.2.3.1 Indicación de compás

La indicación de compás indica el metro de la pieza: su patrón regular de partes fuertes y débiles. Se denota por una fracción al comienzo del pentagrama.

La indicación de compás se establece mediante el comando `\time`

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



## Propiedades trucadas frecuentemente

Se puede personalizar el símbolo que se imprime mediante la propiedad `style`. Un valor de `#'()` produce un estilo de fracción para los compases de 4/4 y de 2/2,

```
\time 4/4 c'1
\time 2/2 c'1
\override Staff.TimeSignature #'style = #'()
\time 4/4 c'1
\time 2/2 c'1
```



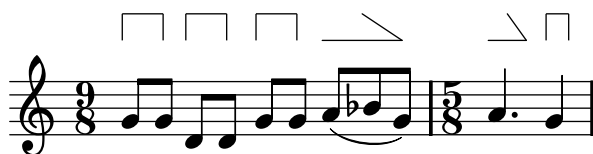
Existen otras muchas opciones para su presentación. Para ver más ejemplos, consulte [Sección 2.8.1.6 \[Indicaciones antiguas de compás\], página 185](#).

`\time` establece las propiedades `timeSignatureFraction`, `beatLength` y `measureLength` en el contexto `Timing`, cuyo nombre alternativo normalmente es `Score`. La propiedad `measureLength` determina el lugar en que se deben insertar las barras de compás y cómo se generan las barras de corchea automáticas. La modificación del valor de `timeSignatureFraction` también provoca que se imprima el símbolo.



Están disponibles más opciones a través de la función de Scheme `set-time-signature`. En combinación con el grabador `Measure_grouping_engraver` (agrupación de compases), creará signos de `MeasureGrouping`. Estos signos facilitan la lectura de música moderna rítmicamente compleja. En el siguiente ejemplo el compás de 9/8 se subdivide en 2, 2, 2 y 3. Esto se pasa a `set-time-signature` como un tercer argumento con el valor (2 2 2 3)

```
\score {
  \relative c' {
    #(set-time-signature 9 8 '(2 2 2 3))
    g8[ g] d[ d] g[ g] a8[( bes g)] |
    #(set-time-signature 5 8 '(3 2))
    a4. g4
  }
  \layout {
    \context {
      \Staff
      \consists "Measure_grouping_engraver"
    }
  }
}
```



## Fragmentos de código seleccionados

### *Indicaciones de compases compuestos*

Las indicaciones de compás poco frecuentes como "5/8" se pueden ejecutar como compases compuestos (p.ej. "3/8 + 2/8"), que combinan dos o más metros diferentes. LilyPond puede hacer la música de este tipo fácil de leer e interpretar, imprimiendo explícitamente las indicaciones de compás compuesto y adaptando el comportamiento automático de las barras (también se pueden añadir indicaciones gráficas de la agrupación de compases; véase el fragmento de código apropiado en la base de datos).

```

#(define (compound-time one two num)
  (markup #:override '(baseline-skip . 0) #:number
    (:#:line ((#:column (one num)) #:vcenter "+") (:#:column (two num))))
  ))

\relative {
  \override Staff.TimeSignature #'stencil = #ly:text-interface::print
  \override Staff.TimeSignature #'text = #(compound-time "2" "3" "8")
  \time 5/8
  #(override-auto-beam-setting '(end 1 8 5 8) 1 4)
  c8 d e fis gis
  c8 fis, gis e d
  c8 d e4 gis8
}
```



## Véase también

Referencia de funcionamiento interno: `TimeSignature`, `Timing_translator`.

Ejemplos: Sección “Rhythms” in *Fragmentos de código*.

## Advertencias y problemas conocidos

El barrado automático no utiliza la agrupación de compases especificada con `set-time-signature`.

### 1.2.3.2 Anacrusas

Los compases parciales como las anacrusas o partes al alzar se escriben usando la construcción `\partial 16*5 c16 cis d dis e | a2. c,4 | b2`



La sintaxis de este comando es la siguiente:

`\partial duración`

donde *duración* es el valor rítmico que se se debe añadir antes del siguiente compás.

Esto se traduce internamente a

`\set Timing.measurePosition = -longitud de la duración`

La propiedad `measurePosition` contiene un número racional que indica qué parte del compás ha transcurrido hasta el momento. Observe que es un número negativo; `\partial 4` se traduce internamente con el significado de “queda un valor de negra en este compás”.

## Advertencias y problemas conocidos

Este comando no tiene en cuenta las notas de adorno al comienzo de la música. Cuando una pieza comienza con notas de adorno en la anacrusa, el `\partial` debe escribirse después de las notas de adorno

`\grace f16`

`\partial 4`

`g4`

`a2 g2`



El uso de `\partial` está previsto solamente para el comienzo de una pieza. Si se emplea después del comienzo se obtendrán varias advertencias desagradables.

### 1.2.3.3 Música sin compasear

Las líneas divisorias y los números de compás se calculan automáticamente. Para música sin compasear (por ejemplo cadencias), esto no es deseable. Para desactivar las barras y los números de compás automáticos, utilice los comandos `\cadenzaOn` y `\cadenzaOff`.

```
c4 d e d
\cadenzaOn
c4 c d8 d d f4 g4.
\cadenzaOff
\bar "|"
d4 e d c
```



### Advertencias y problemas conocidos

LilyPond insertará saltos de línea y de página solamente en las barras de compás. A menos que la música sin compasear acabe antes del final de la línea del pentagrama, tendrá que insertar barras de compás invisibles

```
\bar ""
```

para indicar dónde se pueden producir los saltos.

### 1.2.3.4 Notación polimétrica

### 1.2.3.5 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`. En los siguientes ejemplos, las notas que atraviesan la barra de compás se dividen y se unen mediante una ligadura.

```
\new Voice \with {
  \remove "Note_heads_engraver"
  \consists "Completion_heads_engraver"
} {
  c2. c8 d4 e f g a b c8 c2 b4 a g16 f4 e d c8. c2
}
```



Este grabador divide todas las notas largas en la barra de compás e inserta ligaduras. Uno de sus usos es depurar partituras complejas: si los compases no están completos, las ligaduras mostrarán exactamente cuánto le falta a cada compás.

Si quiere permitir el salto de línea en las barras de compás en las que `Completion_heads_engraver` divide las notas, debe eliminar también `Forbid_line_break_engraver`.

## Advertencias y problemas conocidos

No todas las duraciones (sobre todo las que contienen grupos especiales) se pueden representar exactamente con notas normales y puntillos, pero el grabador nunca insertará grupos especiales.

`Completion_heads_engraver` afecta sólo a las notas; no divide silencios.

## Véase también

Referencia de funcionamiento interno: `Completion_heads_engraver`.

### 1.2.3.6 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 {
  \time 4/4
  c4 e8 f g2 | r4 g r2 | g1 | r1 |
}
```



### 1.2.4 Barras

#### 1.2.4.1 Barras automáticas

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

```
\time 2/4 c8 c c c
\time 6/8 c c c c8. c16 c8
```



Cuando estas decisiones automáticas no son lo bastante buenas, se pueden escribir los barrados de forma explícita; véase [Sección 1.2.4.3 \[Barras manuales\]](#), página 52. También es posible definir patrones de barrado que difieran de los ajustes por defecto; véase [Sección 1.2.4.2 \[Establecer el comportamiento de las barras automáticas\]](#), página 49. Las reglas de barrado predeterminadas están definidas en el archivo ‘`scm/auto-beam.scm`’.

El barrado automático se puede habilitar o desactivar con las instrucciones `\autoBeamOff` y `\autoBeamOn`:

```
c4 c8 c8. c16 c8. c16 c8
\autoBeamOff
c4 c8 c8. c16 c8.
\autoBeamOn
c16 c8
```



## Fragmentos de código seleccionados

Los patrones de barrado se pueden alterar con la propiedad `beatGrouping`:

```
\time 5/16
\set beatGrouping = #'(2 3)
c8[^(2+3)" c16 c8]
\set beatGrouping = #'(3 2)
c8[^(3+2)" c16 c8]
```



De forma predeterminada, las barras de semicorchea o notas más cortas no se subdividen. Esto es, el grupo de tres o más barras se amplía sin romperse sobre grupos enteros de notas. Este comportamiento se puede modificar para que se subdividan las barras en subgrupos ajustando la propiedad `subdivideBeams`. Cuando está establecida, varias barras se verán subdivididas a intervalos definidos por el valor actual `beatLength` reduciendo las barras múltiples a una sola entre los subgrupos. Observe que `beatLength` reside dentro del contexto de `Score` y tiene un valor predeterminado de negra. Se debe ajustar para que sea una fracción que dé como resultado la duración del subgrupo de barras usando la función `make-moment`, como se muestra aquí:

```
c32[ c c c c c c c]
\set subdivideBeams = ##t
c32[ c c c c c c c]
% Establecer longitud de barra del subgrupo a corchea
\set Score.beatLength = #(ly:make-moment 1 8)
c32[ c c c c c c c]
% Establecer longitud de barra del subgrupo a semicorchea
\set Score.beatLength = #(ly:make-moment 1 16)
c32[ c c c c c c c]
```



Para más información sobre `make-moment`, véase [Sección 1.2.6.3 \[Gestión del tiempo\]](#), página 63.

Normalmente están prohibidos los saltos de línea cuando las barras atraviesan a la línea divisoria. Este comportamiento se puede modificar ajustando la propiedad `breakable`: `\override Beam #'breakable = ##t`.

```
\override Beam #'breakable = ##t
c8 \repeat unfold 15 { c[ c] } c
```



Cuando se detecta un salto muy grande entre las cabezas de las notas, se inserta automáticamente una barra doblada o en ángulo. Este comportamiento se puede afinar a través de la propiedad `auto-knee-gap`. Se dibuja una barra doblada si el salto es mayor que el valor de `auto-knee-gap` más el ancho del propio objeto barra (que depende de la duración de las notas y la inclinación de la barra). De forma predeterminada `auto-knee-gap` está establecido al valor de 5.5 espacios de pentagrama.

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



## Véase también

Referencia de la notación: [Sección 1.2.4.3 \[Barras manuales\]](#), página 52, [Sección 1.2.4.2 \[Establecer el comportamiento de las barras automáticas\]](#), página 49.

Fragmentos de código: [Sección “Rhythms” in \*Fragmentos de código\*](#).

Referencia de funcionamiento interno: `Beam`.

## Advertencias y problemas conocidos

No se pueden usar conjuntamente barras dobladas automáticamente que cruzan pentagramas y pentagramas ocultos. Véase [Sección 1.6.1.3 \[Ocultar pentagramas\]](#), página 100.

Las barras pueden colisionar con las cabezas de las notas y las alteraciones accidentales de las otras voces.

### 1.2.4.2 Establecer el comportamiento de las barras automáticas

En compases de tipo normal, las barras automáticas pueden comenzar en cualquier nota pero solamente pueden terminar en algunas posiciones dentro del compás: las barras pueden terminar sobre una parte, o sobre duraciones que estén especificadas por las propiedades de `autoBeamSettings`. Las propiedades de `autoBeamSettings` consisten en una lista de reglas que especifican dónde pueden comenzar y terminar las barras. Las reglas predeterminadas de `autoBeamSettings` están definidas en `'scm/auto-beam.scm'`.

Para poder añadir una regla a la lista, utilice

```
#(override-auto-beam-setting '(be p q n m) a b [context])
```

- `be` es `begin` o bien `end`.
- `p/q` es la duración de la nota para la que quiere añadir una regla. Se considera que una barra tiene la duración de su nota más breve. Ajuste `p` y `q` a `'*` para que esto se aplique a cualquier barra.
- `n/m` es la indicación de compás al que se aplica esta regla. Ajuste `n` y `m` a `'*` para hacer que se aplique a cualquier tipo de compás.
- `a/b` es la posición dentro del compás en que la barra debe comenzar o acabar.

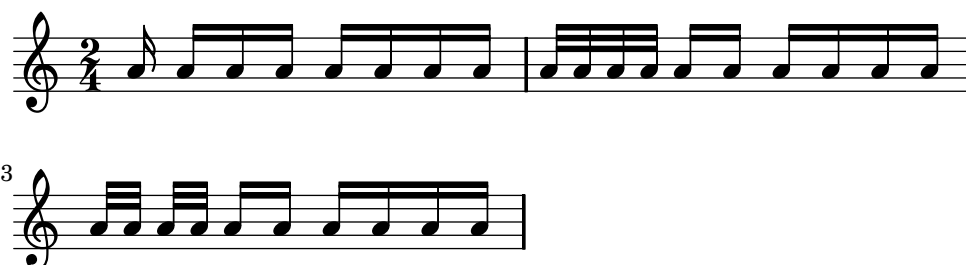
- `contexto` es opcional, y especifica el contexto en que se debe hacer el cambio. El valor predeterminado es `'Voice`.

`#(score-override-auto-beam-setting '(A B C D) E F)` es equivalente a `#(override-auto-beam-setting '(A B C D) E F 'Score)`.

Por ejemplo, si las barras automáticas siempre debe terminar sobre la primera negra, use `#(override-auto-beam-setting '(end * * * *) 1 4)`

Podemos forzar los ajustes de las barras para que sólo tengan efecto sobre barras cuya nota más breve es una cierta duración:

```
\time 2/4
#(override-auto-beam-setting '(end 1 16 * *) 1 16)
a16 a a a a a a a |
a32 a a a a16 a a a a a |
#(override-auto-beam-setting '(end 1 32 * *) 1 16)
a32 a a a a16 a a a a a |
```



Podemos forzar los ajustes de las barras para que sólo tengan efecto en ciertos tipos de compás:

```
\time 5/8
#(override-auto-beam-setting '(end * * 5 8) 2 8)
c8 c d d d
\time 4/4
e8 e f f e e d d
\time 5/8
c8 c d d d
```



También podemos eliminar una regla de fin de barrado previamente establecida usando `#(revert-auto-beam-setting '(be p q n m) a b [contexto])`

`be`, `p`, `q`, `n`, `m`, `a`, `b` y `context` son los mismos que antes. Observe que las reglas predeterminadas se especifican en `'scm/auto-beam.scm`, así que podemos revertir las reglas que no hayamos creado explícitamente.

```
\time 4/4
a16 a a a a a a a a a a a a a a
#(revert-auto-beam-setting '(end 1 16 4 4) 1 4)
a16 a a a a a a a a a a a a a a
```





La regla de un enunciado `revert-auto-beam-setting` debe corresponderse exactamente con la regla original. Esto es, no se tiene en cuenta ninguna expansión de comodines.

```
\time 1/4
#(override-auto-beam-setting '(end 1 16 1 4) 1 8)
a16 a a a
#(revert-auto-beam-setting '(end 1 16 * *) 1 8) % ¡esto no lo devuelve a su valor anterior!
a a a a
#(revert-auto-beam-setting '(end 1 16 1 4) 1 8) % esto sí
a a a a
```



Si las barras automáticas han de terminar en cada negra de un compás de 5/4 time, especifique todas las terminaciones

```
#(override-auto-beam-setting '(end * * * *) 1 4 'Staff)
#(override-auto-beam-setting '(end * * * *) 1 2 'Staff)
#(override-auto-beam-setting '(end * * * *) 3 4 'Staff)
#(override-auto-beam-setting '(end * * * *) 5 4 'Staff)
...
```

Se puede usar la misma sintaxis para especificar los puntos de inicio de las barras. En este ejemplo, las barras automáticas sólo pueden terminar sobre una negra con puntillo:

```
#(override-auto-beam-setting '(end * * * *) 3 8)
#(override-auto-beam-setting '(end * * * *) 1 2)
#(override-auto-beam-setting '(end * * * *) 7 8)
```

En el compás de 4/4, esto significa que las barras automáticas sólo pueden terminar sobre 3/8 y sobre el cuarto pulso del compás (después de que 3/4, esto es, 2 veces 3/8, han transcurrido del compás).

Si se produce cualquier comportamiento inesperado de las barras, compruebe los ajustes de barrado automático predeterminados que están en `'scm/auto-beam.scm'` en busca de una posible interferencia, porque los finales de barra que están allí definidos aún se aplicarán encima de sus propias sobreescrituras. Cualquier final no deseado en los valores predeterminados se debe revertir para nuestro compás concreto.

Por ejemplo, para tipografiar finales de barra de (3 4 3 2) en un 12/8, comience por

```
%% revertir los valores predeterminados de scm/auto-beam.scm respecto al compás de 12/8
#(revert-auto-beam-setting '(end * * 12 8) 3 8)
#(revert-auto-beam-setting '(end * * 12 8) 3 4)
#(revert-auto-beam-setting '(end * * 12 8) 9 8)

%% sus nuevos valores
#(override-auto-beam-setting '(end 1 8 12 8) 3 8)
#(override-auto-beam-setting '(end 1 8 12 8) 7 8)
#(override-auto-beam-setting '(end 1 8 12 8) 10 8)
```

Si se usan barras para indicar melismas en canciones, entonces el barrado automático se debe inhabilitar con `\autoBeamOff`.



## Instrucciones predefinidas

`\autoBeamOff, \autoBeamOn.`

## Advertencias y problemas conocidos

Si una partitura termina mientras una barra automática no ha terminado aún todavía está aceptando notas, esta última barra no se imprimirá en absoluto. Lo mismo sirve para las voces polifónicas, introducidas con `<< ... \ \ ... >>`. Si una voz polifónica termina mientras una barra automática está aún aceptando notas, no se imprime.

## Véase también

Fragmentos de código: [Sección “Rhythms” in \*Fragmentos de código\*](#).

### 1.2.4.3 Barras manuales

En ciertos casos puede ser preciso sobrecribir 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 ]`

```
{
  r4 r8[ g' a r8] r8 g[ | a] r8
}
```



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

```
\time 2/4 c8 c\noBeam c c
```



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.

```
{
  f8[ r16 f g a]
  f8[ r16
  \set stemLeftBeamCount = #1
  f g a]
}
```



## Fragmentos de código seleccionados

### 1.2.4.4 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 acelerando 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.

```
\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] }
% volver a usar barras continuas
\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.

## Advertencias y problemas conocidos

La instrucción `\featherDurations` sólo funciona con fragmentos musicales muy breves, y cuando los números de la fracción son pequeños.

## Véase también

Fragmentos de código: [Sección “Rhythms” in \*Fragmentos de código\*](#).

### 1.2.5 Barras de compás

### 1.2.5.1 Barras de compás

Las líneas divisorias delimitan a los compases pero también se pueden usar para indicar las repeticiones. Normalmente se insertan de manera automática. Los saltos de línea solamente pueden producirse sobre las barras de compás.

Se pueden forzar tipos especiales de línea divisoria con el comando `\bar`

```
c4 \bar " | : " c4
```



Se encuentran disponibles los siguientes tipos de barra:

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.

Para permitir un salto de línea donde no hay ninguna línea divisoria visible, utilice

```
\bar ""
```

De esta manera se insertará una barra invisible de compás y se hará posible el salto de línea en este punto (sin incrementar el contador de los números de compás).

En las partituras con muchos pentagramas, un comando `\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 {
      e'4 d'
      \bar " || "
      f' e'
    }
    \new Staff { \clef bass c4 g e g }
  >>
  \new Staff { \clef bass c2 c2 }
>>
```



## Propiedades trucadas frecuentemente

El comando `\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.

Una línea divisoria se crea cuando se establece la propiedad `whichBar`. Al comienzo del compás su valor se toma de `Timing.defaultBarType`. Los contenidos de los `repeatCommands` se utilizan para sobrescribir a las barras de compás por omisión.

Le recomendamos que utilice siempre `\repeat` para hacer repeticiones. Consulte [Sección 1.4 \[Repeticiones\]](#), página 78.

## Véase también

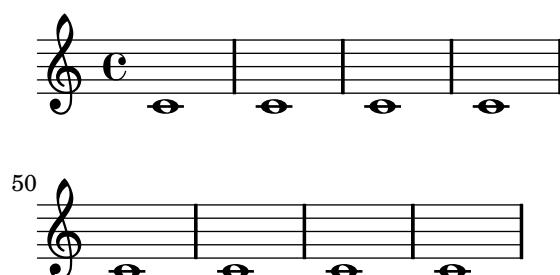
Referencia de la notación: [Sección 1.4 \[Repeticiones\]](#), página 78, [Sección 1.6.1.1 \[Delimitadores del comienzo de un sistema\]](#), página 98.

Referencia de funcionamiento interno: `BarLine` (creada al nivel de `Staff` (pentagrama)), `SpanBar` (a través de los pentagramas).

### 1.2.5.2 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:

```
c1 c c c
\break
\set Score.currentBarNumber = #50
c1 c c c
```



## Fragmentos de código seleccionados

Los números de compás sólo se pueden imprimir en las líneas divisorias; para imprimir un número de compás al principio de la pieza, se debe añadir una línea divisoria vacía

```
\set Score.currentBarNumber = #50
\bar ""
\repeat unfold 4 {c4 c c c} \break
\repeat unfold 4 {c4 c c c}
```





Los números de compás se pueden tipografiar a intervalos regulares en vez de al principio de cada línea.

Los números de compás se pueden suprimir completamente quitando el grabador de números de compás de la partitura.

```
\layout {
  \context {
    \Score
    \remove "Bar_number_engraver"
  }
}
\relative c''{
c4 c c c \break
c4 c c c
}
```



## Véase también

Referencia de funcionamiento interno: `BarNumber`.

Fragmentos de código: [Sección “Rhythms”](#) in *Fragmentos de código*.

## 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.

### 1.2.5.3 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 |
```

Las comprobaciones de compás también se pueden usar en la letra de las canciones, por ejemplo

```
\lyricmode {
  \time 2/4
  Bri -- lla | Bri -- lla |
}
```

Las duraciones incorrectas a menudo desbaratan la partitura completamente, sobre todo si la música es polifónica, por ello el mejor sitio para empezar a corregir la entrada es hacer un barrido en busca de comprobaciones de compás fallidas y duraciones incorrectas.

También es posible redefinir la acción que se lleva a cabo cuando se encuentra un símbolo de comprobación de compás o «símbolo de tubería», |, en el código de entrada, de forma que haga algo distinto a una comprobación de compás. Se hace asignando una expresión musical a `pipeSymbol` («símbolo de tubería»). 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.

```
simboloTubo = \bar "||"
{
  c'2 c'2 |
  c'2 c'2
  c'2 | c'2
  c'2 c'2
}
```



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:

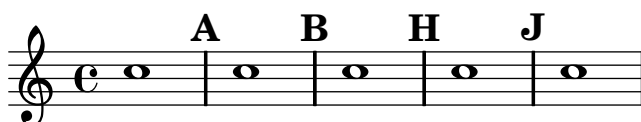
```
\barNumberCheck #123
```

imprimirá una advertencia si el `currentBarNumber` (número del compás actual) no es 123 cuando se procesa.

#### 1.2.5.4 Llamadas de ensayo

Para imprimir una letra de ensayo, utilice la orden `\mark`

```
c1 \mark \default
c1 \mark \default
c1 \mark #8
c1 \mark \default
c1 \mark \default
```



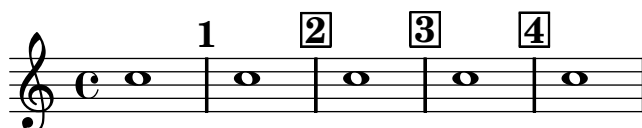
La letra 'I' se salta de acuerdo con las tradiciones de grabado. Si quiere incluir la letra 'I', use

```
\set Score.markFormatter = #format-mark-alphabet
```

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`.

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.

```
\set Score.markFormatter = #format-mark-numbers
c1 \mark \default
c1 \mark \default
\set Score.markFormatter = #format-mark-box-numbers
c1 \mark \default
c1 \mark \default
c1
```



El archivo `'scm/translation-functions.scm'` contiene las definiciones de `format-mark-numbers` (el formato por omisión), `format-mark-box-numbers`, `format-mark-letters` y `format-mark-box-letters`. 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.

Otros estilos de marca de ensayo se pueden especificar de forma manual:

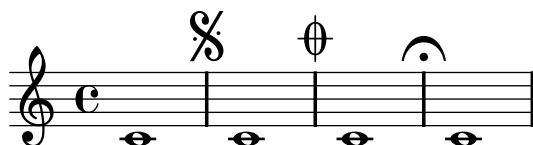
```
\mark "A1"
```

`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`

```
c1 \mark \markup { \musicglyph #"scripts.segno" }
c1 \mark \markup { \musicglyph #"scripts.coda" }
c1 \mark \markup { \musicglyph #"scripts.ufermata" }
c1
```



Consulte [Sección B.4 \[La tipografía Feta\]](#), página 244 para ver una lista de los símbolos que se pueden imprimir con `\musicglyph`.

Para ver formas comunes de trucar la colocación de las marcas de ensayo, consulte [Sección 1.8.1.3 \[Indicaciones de texto\]](#), página 120.

## Véase también

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

En este manual: Sección B.4 [La tipografía Feta], página 244, Sección 1.8.2 [Formatear el texto], página 125.

Referencia de funcionamiento interno: RehearsalMark.

Archivos de inicio: ‘scm/translation-functions.scm’ contiene la definición de `format-mark-numbers` y de `format-mark-letters`. Se pueden usar como inspiración para escribir otras funciones de formateo.

## 1.2.6 Asuntos rítmicos especiales

### 1.2.6.1 Notas de adorno

Los mordentes y notas de adorno son ornamentos escritos. El más común es la acciaccatura o mordente de una nota, que se debe tocar muy corta. Se denota mediante una corchea pequeña, ligada, con un corchete tachado o atravesado por una barrita. La appoggiatura es una nota de adorno que toma una fracción fija de la nota principal, y se denota como una nota pequeña, ligada, sin tachar. Ambas se introducen con los comandos `\acciaccatura` y `\appoggiatura`, como se muestra en el ejemplo siguiente

```
b4 \acciaccatura d8 c4 \appoggiatura e8 d4
\acciaccatura { g16[ f] } e4
```



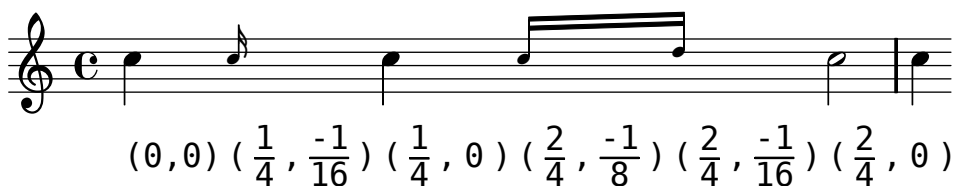
Ambas son formas especiales del comando `\grace`. Al anteponer esta palabra a una expresión musical se crea una nueva expresión, que se imprime en un tipo más pequeño y no ocupa ningún tiempo lógico en el compás.

```
c4 \grace c16 c4
\grace { c16[ d16] } c2 c4
```



Al contrario que `\acciaccatura` y `\appoggiatura`, el comando `\grace` no inicia una ligadura.

Internamente, la cuenta del tiempo para las notas de adorno se hace utilizando un segundo contador, el contador ‘de adorno’. Cada instante en el tiempo comprende dos números racionales: uno denota el tiempo lógico, y el otro denota el tiempo de adorno. El ejemplo anterior se muestra a continuación con vectores de tiempo



La colocación de las 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 { e4 \grace { c16[ d e f] } e4 }
    \new Staff { c4 \grace { g8[ b] } c4 } >>
```



Si quiere acabar una nota con un adorno, use el comando `\afterGrace`. Toma dos argumentos: la nota principal y las notas de adorno que siguen a la nota principal.

```
c1 \afterGrace d1 { c16[ d] } c4
```



Esto pondrá las notas de adorno después de un ‘espacio’ que dura  $3/4$  de la longitud de la nota principal. La fracción  $3/4$  se puede cambiar estableciendo `afterGraceFraction`, por ejemplo

```
#(define afterGraceFraction (cons 7 8))
```

pondrá la nota de adorno a  $7/8$  de la nota principal.

Se puede conseguir el mismo efecto de forma manual haciendo

```
\new Voice {
  << { d1^\trill_( }
    { s2 \grace { c16[ d] } } >>
  c4)
}
```



Ajustando la duración de la nota de desplazamiento (aquí es una blanca), se ajusta el espacio entre la nota principal y la de adorno.

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, por ejemplo:

```
\new Voice {
  \acciaccatura {
    \stemDown
    f16->
    \stemNeutral
  }
  g4
}
```



Las sobreescrituras (overrides) también se deben deshacer dentro de la expresión de adorno.

La disposición de las expresiones de adorno se pueden cambiar a lo largo de la música usando la función `add-grace-property`. El siguiente ejemplo suprime la definición de la dirección de la `Stem` (plica) para este adorno, y así las plicas no siempre apuntan hacia arriba.

```
\new Staff {
  #(add-grace-property 'Voice 'Stem 'direction '())
  ...
}
```

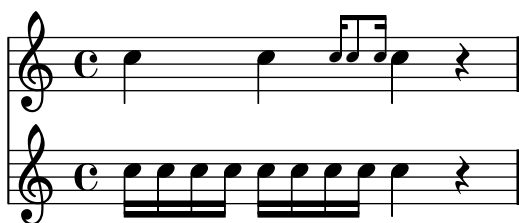
Otra opción es modificar las variables `startGraceMusic`, `stopGraceMusic`, `startAcciaccaturaMusic`, `stopAcciaccaturaMusic`, `startAppoggiaturaMusic` y `stopAppoggiaturaMusic`. Hay más información en el archivo `ly/grace-init.ly`.

La barrita que cruza el corchete de las acciaccaturas se puede obtener en otras situaciones mediante `\override Stem #'stroke-style = #"grace"`.

## Propiedades trucadas frecuentemente

Se puede forzar la alineación de las notas de adorno con las notas normales de los otros pentagramas `Grace notes may be forced to use align with regular notes` estableciendo `strict-grace-spacing` al valor `##t`:

```
<<
\override Score.SpacingSpanner #'strict-grace-spacing = ##t
\new Staff {
  c4
  \afterGrace c4 { c16[ c8 c16] }
  c4 r
}
\new Staff {
  c16 c c c c c c c c4 r
}
>>
```



## Véase también

Referencia de funcionamiento interno: `GraceMusic`.

## Advertencias y problemas conocidos

Una partitura que comienza con una expresión `\grace` requiere una declaración explícita `\new Voice` pues en caso contrario la nota principal y la de adorno acabarán en pentagramas distintos.

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 { e4 \bar "|" \grace c16 d4 }
    \new Staff { c4 \bar "|" d4 } >>
```



Esto se puede remediar insertando desplazamientos de adorno de las duraciones correspondientes en los otros pentagramas. Para el ejemplo anterior

```
<< \new Staff { e4 \bar "|" \grace c16 d4 }
    \new Staff { c4 \bar "|" \grace s16 d4 } >>
```



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.

### 1.2.6.2 Alinear con una cadenza

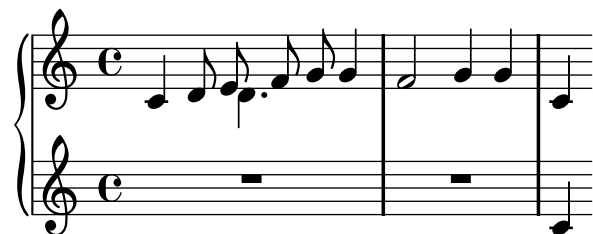
En un contexto orquestal, las cadenzas presentan un problema especial: al construir una partitura que tiene una cadenza, todos los demás instrumentos deben saltar tantas notas como la longitud de la cadenza, pues en caso contrario empezarán demasiado pronto o tarde.

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 multicomás, de la longitud exacta del fragmento. El uso de `mmrest-of-length` se muestra en el ejemplo siguiente.

```
cadenza = \relative c' {
  c4 d8 << { e f g } \ { d4. } >>
  g4 f2 g4 g
}

\new GrandStaff <<
  \new Staff { \cadenza c'4 }
  \new Staff {
    #(ly:export (mmrest-of-length cadenza))
    c'4
  }
>>
```

}  
>>



### 1.2.6.3 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`.

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 [Sección 1.2.5.2 \[Numeración de compases\]](#), página 55.

#### `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 determinad 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.

```
\set Score.measureLength = #(ly:make-moment 5 4)
c1 c4
c1 c4
c4 c4
\set Score.measurePosition = #(ly:make-moment 5 8)
b4 b4 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

En este manual: [Sección 1.2.5.2 \[Numeración de compases\]](#), página 55, [Sección 1.2.3.3 \[Música sin compasear\]](#), página 46

Fragmentos de código: [Sección “Rhythms” in \*Fragmentos de código\*](#).

Referencia de funcionamiento interno: `Timing_translator`, `Score`.

## 1.3 Expresiones



### 1.3.1 Adosado a las notas

#### 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 B.8 \[Lista de articulaciones\]](#), página 295. Por ejemplo:

`c4\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:

```
c4-^ c-+ c-- c-|
c-> c-. c-_
```



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 230.

## 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`, `dashBar`, `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 semitritino 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 semitritino (el `Script`) 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 }
}
```



## Véase también

Gloario musical: Sección “tenuto” in *Glosario Musical*, Sección “acento” in *Glosario Musical*, Sección “staccato” in *Glosario Musical*, Sección “portato” in *Glosario Musical*.

Referencia de la notación: Sección 5.4.2 [Dirección y posición], página 230.

Archivos instalados: ‘scm/script.scm’.

Fragmentos de código: Sección “Expressive marks” in *Fragmentos de código*.

Referencia de funcionamiento interno: Script, TextScript.

## Advertencias y problemas conocidos

Estos signos aparecen en la salida impresa pero no tienen ningún efecto sobre la representación de MIDI de la música.

## Matices dinámicos

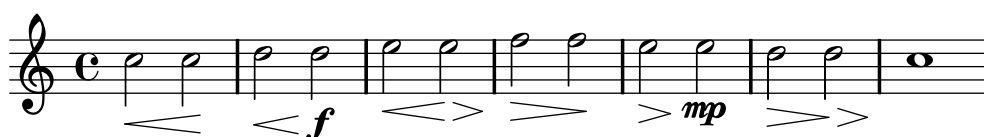
Las marcas dinámicas de matiz absoluto se especifican usando un comando después de una nota: `c4\ff`. Las marcas dinámicas disponibles son `\ppppp`, `\pppp`, `\ppp`, `\pp`, `\p`, `\mp`, `\mf`, `\f`, `\ff`, `\fff`, `\ffff`, `\fp`, `\sf`, `\sff`, `\sp`, `\spp`, `\sfz` y `\rfz`. Las indicaciones dinámicas se pueden colocar manualmente por encima o por debajo del pentagrama; para ver más detalles, consulte Sección 5.4.2 [Dirección y posición], página 230.

```
c2\ppp c\mp
c\rfz c^\mf
c_\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 `\>`. De forma predeterminada, se tipografían reguladores en ángulo cuando se utiliza esta notación.

```
c2\< c\!
d2\< d\f
e2\< e\>
f2\> f\!
e2\> e\mp
d2\> d\>
c1\!
```

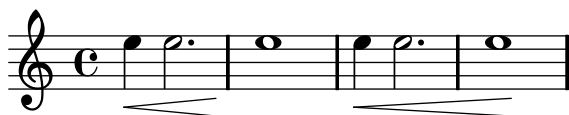


Se requieren notas espaciadoras para tipografiar varias indicaciones dinámicas sobre una sola nota.





```
\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\!
  \override Hairpin #'minimum-length = #5
  << f1 { s4 s\< s\> s\! } >>
}
```



*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`.

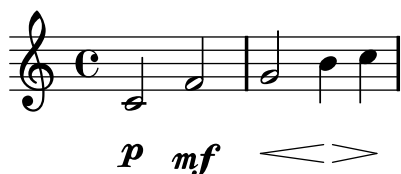
```
\relative c'' {
  \override Hairpin #'circled-tip = ##t
  c2\< c\!
  c4\> c\< c2\!
}
```



*Alinear verticalmente expresiones dinámicas que abarcan varias notas*

Las expresiones dinámicas que se comienzan, terminan o se producen en la misma nota se alinean verticalmente. Para asegurar que las expresiones dinámicas se alinean cuando no se producen sobre la misma nota, incrementa la propiedad `staff-padding` del objeto `DynamicLineSpanner`.

```
\relative c' {
  \override DynamicLineSpanner #'staff-padding = #4
  c2\p f\mf
  g2\< b4\> c\!
}
```



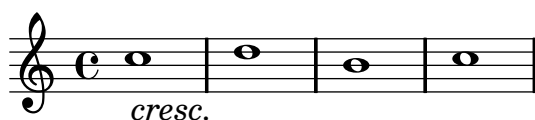
Los *Crescendi* y *decrescendi* que atraviesan un salto de línea continúan en la línea siguiente. Si terminan en la primera nota de la línea nueva, no se imprime nada en esa línea. Para cambiar este comportamiento, utilice

```
\override Score.Hairpin #'after-line-breaking = ##t
```

*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 #'dash-period = #-1.0
  \crescTextCresc
  c1\< | d | b | c\!
}
```



El texto que se emplea para los *crescendi* y *decrescendi* se puede modificar cambiando las propiedades de contexto `crescendoText` y `decrecendoText`. El estilo de la línea extensora se puede cambiar modificando la propiedad `'style` de `DynamicTextSpanner`. El valor predeterminado es `'hairpin` (regulador), y entre otros valores posibles se encuentran `'line`, `'dashed-line` y `'dotted-line`:

```
\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 “crescendo” in \*Glosario Musical\*](#), [Sección “decrecendo” in \*Glosario Musical\*](#).

Manual de aprendizaje: [Sección “Articulation and dynamics” in \*Manual de Aprendizaje\*](#).

Referencia de la notación: [Sección 5.4.2 \[Dirección y posición\]](#), página 230.

Fragmentos de código: [Sección “Expressive marks” in \*Fragmentos de código\*](#).

Referencia de funcionamiento interno: `DynamicText`, `Hairpin`, `DynamicLineSpanner`.

## Indicaciones dinámicas contemporáneas

Es posible imprimir indicaciones dinámicas de nueva creación o texto que deba ir alineado con los matices. Utilice `make-dynamic-script` para crear estas indicaciones. Observe que la fuente tipográfica para los matices solamente contiene los caracteres `f`, `m`, `p`, `r`, `s` y `z`.

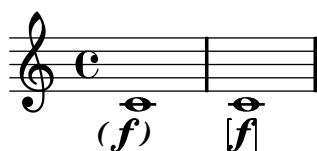
Algunas situaciones (como las indicaciones dinámicas) tienen un valor prefijado para las propiedades relacionadas con la fuente tipográfica. Si está creando texto en tales situaciones, es recomendable que cancele dichas propiedades con `normal-text`. Consulte [Sección B.6 \[Text markup commands\]](#), página 258 para ver más detalles.

```
sfzp = #(make-dynamic-script "sfzp")
\relative c' {
  c4 c c\sfpz c
}
```



También se pueden imprimir matices entre paréntesis o entre corchetes rectos. Se utilizan con frecuencia para añadir indicaciones dinámicas editoriales.

```
rndf = \markup { \center-align { \line { \bold { \italic ( }
  \dynamic f \bold { \italic ) } } } } }
boxf = \markup { \bracket { \dynamic f } }
\relative c' {
  c1\_rndf
  c1\_boxf
}
```



## Véase también

Fragmentos de código: [Sección “Expressive marks”](#) in *Fragmentos de código*.

### 1.3.2 Curvas

#### Ligaduras de expresión

Una *ligadura de expresión* indica que las notas se deben tocar unidas o *legato*. Se escriben utilizando paréntesis:

```
f( g a) a8 b( a4 g2 f4)
<c e>2( <b d>2)
```



La dirección de una ligadura de expresión se puede especificar con `\slurDIRECCIÓN`, donde *DIRECCIÓN* es o bien *Up* (arriba), *Down* (abajo), o *Neutral* (seleccionada automáticamente). *Neutral* es la predeterminada. Las ligaduras de expresión se pueden situar manualmente por encima o por debajo del pentagrama, véase [Sección 5.4.2 \[Dirección y posición\]](#), página 230.

```
c4_( c) c^( c)
```



Usando este método solamente se puede imprimir una ligadura de expresión cada vez. Para imprimir una ligadura larga por encima de varias ligaduras cortas, consulte [\[Ligaduras de fraseo\]](#), página 71.

## Instrucciones predefinidas

```
\slurUp, \slurDown, \slurNeutral, \slurDashed, \slurDotted, \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>)
}
```



## Véase también

Glosario musical: [Sección “ligadura de expresión”](#) in *Glosario Musical*.

Referencia de la notación: [Sección 5.4.2 \[Dirección y posición\]](#), página 230.

Fragmentos de código: [Sección “Expressive marks”](#) in *Fragmentos de código*.

Referencia de funcionamiento interno: `Slur`.

## Ligaduras de fraseo

Una ligadura de fraseo (o marca de fraseo) conecta las notas y se utiliza para indicar una frase musical. Se escribe usando `\(` y `\)` respectivamente

```
\time 6/4 c\( d( e) f( e) d\)
```



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; utilice en su lugar `\phrasingSlurUp`, `\phrasingSlurDown`, y `\phrasingSlurNeutral`. Las ligaduras de fraseo se pueden situar manualmente por encima o por debajo del pentagrama, véase [Sección 5.4.2 \[Dirección y posición\]](#), [página 230](#).

No se pueden tener varias ligaduras de fraseo simultáneas.

## Instrucciones predefinidas

`\phrasingSlurUp`, `\phrasingSlurDown`, `\phrasingSlurNeutral`.

## Véase también

Referencia de la notación: [Sección 5.4.2 \[Dirección y posición\]](#), [página 230](#).

Fragmentos de código: [Sección “Expressive marks” in Fragmentos de código](#).

Referencia de funcionamiento interno: `PhrasingSlur`.

## Marcas de respiración

Las respiraciones se introducen utilizando `\breathe`:

```
c4 \breathe d4
```



## 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
}
```



## Véase también

Fragmentos de código: [Sección “Winds” in Fragmentos de código](#).

Referencia de funcionamiento interno: `BreathingSign`.

## Caídas y elevaciones

Se pueden expresar *caídas* y *subidas* de tono (falls y doits) añadidas a las notas mediante la instrucción `\bendAfter`:

```
c2-\bendAfter #+4
c-\bendAfter #-4
c-\bendAfter #+8
c-\bendAfter #-8
```



El guión ( - ) que sigue al nombre de la nota es *necesario* al escribir *caídas* y *subidas*.

## 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 #-3
  c2-\bendAfter #+8
  c2-\bendAfter #-6
}
```



## Véase también

Fragmentos de código: [Sección “Expressive marks” in Fragmentos de código.](#)

### 1.3.3 Líneas

#### Glissando

Un *glissando* es un cambio gradual en la altura. Se denota por medio de una línea o una línea ondulada entre dos notas. El *glissando* se crea adjuntando `\glissando` a la nota:

```
c2\glissando c'
\override Glissando #'style = #'zigzag
c2\glissando c,
```



## Fragmentos de código seleccionados

```
I = \once \override NoteColumn #'ignore-collision = ##t

\relative c' <<
  { \oneVoice \stemDown f2 \glissando \stemNeutral a } \\  

  { \oneVoice \I c2 \glissando \I d, }
>>
```



## Véase también

Glosario musical: [Sección “glissando” in \*Glosario Musical\*.](#)

Fragmentos de código: [Sección “Expressive marks” in \*Fragmentos de código\*.](#)

Referencia de funcionamiento interno: [Glissando](#).

## Advertencias y problemas conocidos

La impresión de texto sobre la línea (como *gliss.*) no está contemplada.

## Arpeggio

Se puede especificar un signo de *arpeggio* (conocido también como *acorde quebrado*) sobre un acorde adjuntando un `\arpeggio` al acorde:

```
<c e g c>\arpeggio
```



Un corchete recto a la izquierda, indicado mediante `\arpeggioBracket`, denota que el intérprete *no* tiene que arpeggiar el acorde:

```
\arpeggioBracket
<c e g c>\arpeggio
```



La dirección del arpeggio se denota a veces mediante la adición de una punta de flecha a la línea ondulada. Esto se hace con las instrucciones `arpeggioArrowUp` y `arpeggioArrowDown`. `arpeggioNormal` nos devuelve a la versión sin flecha:

```
\new Voice {
  \arpeggioArrowUp
  <c e g c>\arpeggio
  \arpeggioArrowDown
}
```

```

<c e g c>\arpeggio
\arpeggioNormal
<c e g c>\arpeggio
}

```



Los Arpeggios se pueden desarrollar explícitamente utilizando *ligaduras de unión*. Para ver más información, consulte [Sección 1.2.1.4 \[Ligaduras de unión\]](#), página 35.

## Instrucciones predefinidas

`\arpeggio`, `\arpeggioArrowUp`, `\arpeggioArrowDown`, `\arpeggioNormal`, `\arpeggioBracket`.

## Fragmentos de código seleccionados

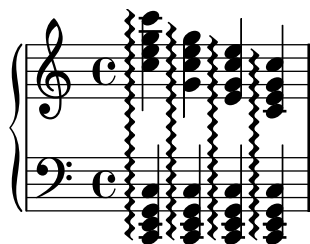
*Crear arpeggios que se cruzan entre pentagramas dentro de un sistema de piano*

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 `PianoStaff` si se incluye el grabador `Span_arpeggio_engraver` en el contexto de `Score`.

```

\score {
  \new StaffGroup {

```



```

\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` context:

```

\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” in \*Glosario Musical\*](#).

Referencia de la notación: [Sección 1.2.1.4 \[Ligaduras de unión\]](#), página 35.

Fragmentos de código: [Sección “Expressive marks” in \*Fragmentos de código\*](#).

Referencia de funcionamiento interno: [Arpeggio](#), [PianoStaff](#).

## Advertencias y problemas conocidos

No es posible mezclar arpeggios conectados y no conectados en un `PianoStaff` en el mismo instante de tiempo.

## Trinos

Los *trinos* cortos se imprimen con `\trill` como una articulación normal; véase [\[Articulaciones y ornamentos\]](#), página 64.

Los *trinos* largos mantenidos se hacen con `\startTrillSpan` y `\stopTrillSpan`. En el ejemplo siguiente, se muestra en la usual combinación con notas de adorno. Si se necesita un control más preciso sobre la colocación de las notas de adorno, véase [Sección 1.2.6.1 \[Notas de adorno\]](#), página 59.

```
c1 \afterGrace
d1\startTrillSpan { c16[\stopTrillSpan d] }
c4
```



Los *trinos* que se tienen que ejecutar sobre notas especificadas explícitamente se pueden tipografiar con la instrucción `pitchedTrill` utilizando la sintaxis siguiente:

```
\pitchedTrill nota_principal \startTrillSpan
nota_de_trino nota_final \stopTrillSpan
\pitchedTrill e2 \startTrillSpan fis
d\stopTrillSpan
```



El primer argumento es la nota principal. La altura de la segunda se imprime como una cabeza de nota, sin plica y entre paréntesis.

## Instrucciones predefinidas

`\startTrillSpan`, `\stopTrillSpan`.

## Véase también

Glosario musical: [Sección “trino” in \*Glosario Musical\*](#).

Fragmentos de código: [Sección “Expressive marks” in \*Fragmentos de código\*](#).

Referencia de funcionamiento interno: `TrillSpanner`.

## 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 alternativas.

### 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 Escribir repeticiones largas

### 1.4.1.1 Repeticiones normales

La sintaxis de una repetición simple es

```
\repeat variante número_de_repeticiones cuerpo_de_la_repetición
```

donde *cuerpo\_de\_la\_repetición* es una expresión musical.

Repeticiones normales sin finales alternativos:

```
\repeat volta 2 { c4 d e f }
c2 d
\repeat volta 2 { d4 e f g }
```



Repeticiones normales con primera y segunda vez:

```
\repeat volta 2 { g f e d }
\alternative {
  { cis2 g' }
  { cis,2 b }
}
c1
```



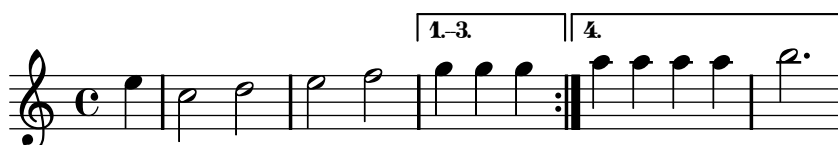
Repeticiones con anacrusa:

```
\new Staff {
  \partial 4
  e |
  \repeat volta 4 { c2 d2 | e2 f2 | }
  \alternative { { g4 g g e } { a a a a | b2. } }
}
```



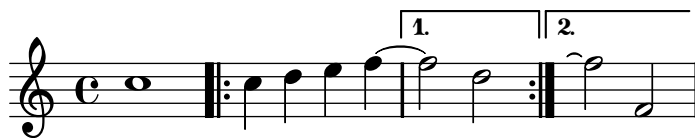
o bien

```
\new Staff {
  \partial 4
  \repeat volta 4 { e | c2 d2 | e2 f2 | }
  \alternative { { \partial 4*3 g4 g g } { a a a a | b2. } }
}
```



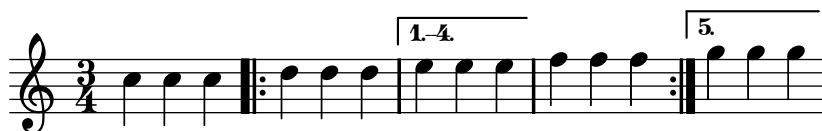
Se pueden añadir ligaduras de unión al segundo final:

```
c1
\repeat volta 2 {c4 d e f ~ }
\alternative { {f2 d} {f\repeatTie f,} }
```



De manera predeterminada, los corchetes de primera y segunda vez se dibujan por encima de todo el fragmento de finales alternativos, pero es posible acortar los corchetes de primera y segunda vez estableciendo `voltaSpannerDuration`. En el ejemplo siguiente, el corchete sólo dura un compás, lo que supone una duración de 3/4.

```
\time 3/4
c c c
\set Score.voltaSpannerDuration = #(ly:make-moment 3 4)
\repeat volta 5 { d d d }
\alternative { { e e e f f f }
{ g g g } }
```



El grabador `Volta_engraver` reside de manera predeterminada dentro del contexto `Score`, y así los corchetes de las repeticiones se imprimen solamente por encima del pentagrama superior. Se puede ajustar esto añadiendo el grabador `Volta_engraver` al contexto `Staff` en que queremos que aparezcan los corchetes; véase [Sección 5.1.3 \[Modificar los complementos \(plug-ins\) de contexto\]](#), página 216 y [Sección “Repeats” in Fragmentos de código](#).

```
\score { <<
  \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' } }
  >> }
```



Si quiere comenzar una repetición al principio de una línea y tiene una doble barra al final de la línea anterior, utilice

```
c4 c c c
\bar "||:" \break
\repeat volta 2 { c4 d e f }
```



consulte [Sección 1.2.5.1 \[Barras de compás\]](#), [página 54](#) para ver más información.

## Véase también

Glosario musical: [Sección “repetición” in \*Glosario Musical\*](#).

Referencia de la notación: [Sección 1.2.5.1 \[Barras de compás\]](#), [página 54](#), [Sección 5.1.3 \[Modificar los complementos \(plug-ins\) de contexto\]](#), [página 216](#).

Fragmentos de código: [Sección “Repeats” in \*Fragmentos de código\*](#).

Referencia de funcionamiento interno: [VoltaBracket](#), [RepeatedMusic](#), [VoltaRepeatedMusic](#), [UnfoldedRepeatedMusic](#).

## Advertencias y problemas conocidos

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.

La información del contador de tiempo no se recuerda al comienzo de una alternativa, por ello después de una repetición la información de la cuenta de tiempo se debe reajustar manualmente; por ejemplo, estableciendo el valor de `Score.measurePosition` o introduciendo `\partial`. De forma similar, las ligaduras de expresión o de unión tampoco se repiten.

### 1.4.1.2 Marcas de repetición manual

Se puede usar la propiedad `repeatCommands` para controlar la disposición de las repeticiones. Su valor es una lista de Scheme de comandos de repetición.

`start-repeat`

Imprimir una barra de compás | :

`end-repeat`

Imprimir una barra de compás : |

(volta *texto*)

Imprimir un corchete de volta que diga *texto*. El texto se puede especificar como una cadena de texto o como un texto de marcado, consulte [Sección 1.8.2 \[Formatear el texto\]](#), página 125. No olvide cambiar la fuente tipográfica, porque el tipo por defecto para los números no contiene caracteres alfabéticos;

(volta #f)

Detener un corchete de volta mantenido.

```
c4
\set Score.repeatCommands = #'((volta "93") end-repeat)
c4 c4
\set Score.repeatCommands = #'((volta #f))
c4 c4
```



## Véase también

Referencia de la notación: [Sección 1.2.5.1 \[Barras de compás\]](#), página 54.

Fragmentos de código: [Sección “Repeats” in Fragmentos de código](#).

Referencia del programa: `VoltaBracket`, `RepeatedMusic`, `VoltaRepeatedMusic` y `UnfoldedRepeatedMusic`.

### 1.4.1.3 Repeticiones explícitas

`unfold`

### 1.4.2 Repeticiones cortas

#### Repeticiones de compás o parte de ellos

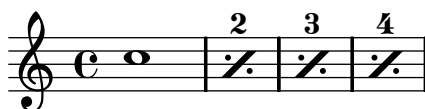
En el estilo `percent` (porcentaje), se puede hacer que se repita un patrón de notas. Se imprime una sola vez, y luego el patrón se sustituye con un símbolo especial. Los patrones de uno y dos compases se sustituyen por símbolos parecidos a un porcentaje, y los patrones que dividen a la duración de un compás se sustituyen por barras inclinadas. Las repeticiones de porcentaje se deben declarar dentro de un contexto `Voice`.

```
\new Voice \relative c' {
  \repeat percent 4 { c4 }
  \repeat percent 2 { c2 es2 f4 fis4 g4 c4 }
}
```



Las repeticiones de compás de más de 2 compases llevan un contador, si se activa la propiedad `countPercentRepeats`,

```
\new Voice {
\set countPercentRepeats = ##t
  \repeat percent 4 { c1 }
}
```



También se pueden imprimir porcentajes aislados. Esto se hace mediante la colocación de un silencio multicompa con una función de impresión distinta,

```
\override MultiMeasureRest #'stencil
= #ly:multi-measure-rest::percent
R1
```



## Véase también

Fragmentos de código: [Sección “Repeats” in \*Fragmentos de código\*](#).

Referencia del programa: `RepeatSlash`, `PercentRepeat`, `DoublePercentRepeat`, `DoublePercentRepeatCounter`, `PercentRepeatCounter`, `PercentRepeatedMusic`.

## Repeticiones de trémolo

Para colocar marcas de trémolo entre las notas, use `\repeat` con el estilo `tremolo` (trémolo):

```
\new Voice \relative c' {
  \repeat tremolo 8 { c16 d16 }
  \repeat tremolo 4 { c16 d16 }
  \repeat tremolo 2 { c16 d16 }
}
```



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 ‘:[*número*]’ después de la nota. El número indica la duración de la subdivisión, y debe ser 8 como mínimo. Un valor de 8 para *número* produce una línea que atraviesa la plica de la nota. Si se omite la longitud, se utiliza el último valor (almacenado en `tremoloFlags`)

```
c'2:8 c':32 | c': c': |
```



## Advertencias y problemas conocidos

Los trémolos que se escriben con ‘:[*número*]’ no se trasladan hacia la salida MIDI.

## Véase también

Referencia de la notación: [\[Repeticiones de trémolo\]](#), página 83.

Referencia de funcionamiento interno: `Beam`, `StemTremolo`.

Fragmentos de código: [Sección “Repeats” in Fragmentos de código](#).

En otros lugares: `StemTremolo`.

## 1.5 Notas simultáneas

La 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 la 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 o un conjunto de articulaciones, como si fueran simples notas.

<c e g>2 <c f a>4-> <e g c>-.



Para ver más información sobre los acordes, consulte [Sección 2.7 \[Notación de acordes\]](#), [página 163](#).

#### Véase también

Glosario musical: [Sección “acorde”](#) in *Glosario Musical*.

Manual de aprendizaje: [Sección “Combining notes into chords”](#) in *Manual de Aprendizaje*.

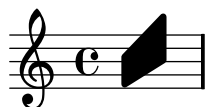
Referencia de la notación: [Sección 2.7 \[Notación de acordes\]](#), [página 163](#).

Fragmentos de código: [Sección “Simultaneous notes”](#) in *Fragmentos de código*.

#### 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.:

```
\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 “racimo \(cluster\)”](#) in *Glosario Musical*.

Fragmentos de código: [Sección “Simultaneous notes”](#) in *Fragmentos de código*.

Referencia de funcionamiento interno: `ClusterSpanner`, `ClusterSpannerBeacon`, `ClusterSpanner_engraver`.

## 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. Los racimos separados necesitarían silencios de separación entre ellos.

### 1.5.2 Varias voces

Esta sección trata las notas simultáneas en varias voces o varios pentagramas.

#### Polifonía en un solo pentagrama

Para 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, incorpórelas dentro de una construcción de música simultánea:

```
\new Voice <<
  { e4 f8 d e16 f g8 d4 }
  { c4 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. Para componer tipográficamente varias voces verdaderamente independientes dentro de un único pentagrama se utiliza la construcción <<{...} \\ {...}>>, en que las dos (o más) voces se encuentran separadas por dobles barras invertidas.

```
<<
  { r8 r16 g e8. f16 g8[ c,] f e16 d }
  \\
  { d16 c d8~ d16 b c8~ c16 b c8~ c16 b8. }
>>
```



Las plicas de las voces primera y tercera se dirigen hacia arriba, y las de las voces segunda y cuarta se dirigen hacia abajo; las cabezas de las notas en las voces tercera y cuarta se desplazan horizontalmente, y los silencios se mueven de manera que eviten las colisiones. En el ejemplo siguiente, la voz intermedia tiene las plicas hacia arriba, por lo que la introducimos en tercer lugar, de forma que se convierte en la voz número tres que tiene las plicas hacia arriba tal y como era nuestro propósito.

```
<<
  { r8 g g g g f16 es f8 d }
  \\
  { es,8 r es r d r d r }
  \\
  { d'8 s c s bes s a s }
>>
```



A menudo se utilizan silencios espaciadores para evitar la aparición de un número excesivo de silencios, como se ha podido ver en el ejemplo anterior.

El método que acabamos de presentar crea dos voces nuevas cuando se encuentra la construcción `<<{...} \ {...}>>` en el código de entrada; para añadir de forma temporal una sola voz adicional a una voz ya existente, es necesario instanciar dicha voz de forma explícita. Para este propósito, se pueden utilizar las instrucciones desde `\voiceOne` (voz uno) hasta `\voiceFour` (voz cuatro) y `\oneVoice` (una voz única):

```
<< { \voiceOne ... } \new Voice { \voiceTwo ... } >> \oneVoice
```

Esta construcción es más apropiada para mantener en todo momento el control de las voces, por ejemplo para facilitar una asignación coherente de la letra al contexto de voz adecuado.

## Instrucciones predefinidas

`\voiceOne`, `\voiceTwo`, `\voiceThree`, `\voiceFour`, `\oneVoice`.

## Véase también

Manual de aprendizaje: [Sección “Voices contain music” in \*Manual de Aprendizaje\*](#), [Sección “Explicitly instantiating voices” in \*Manual de Aprendizaje\*](#).

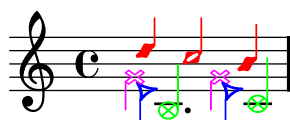
Referencia de la notación: [Sección 2.5.1.3 \[Pautas de percusión\]](#), página 158.

Fragmentos de código: [Sección “Simultaneous notes” in \*Fragmentos de código\*](#).

## Estilos de voz

Se pueden aplicar colores y formas distintos a las voces para permitir identificarlas fácilmente:

```
<<
  { \voiceOneStyle d4 c2 b4 }
  \
  { \voiceTwoStyle e,2 e }
  \
  { \voiceThreeStyle b2. c4 }
  \
  { \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 “I’m hearing Voices” in *Manual de Aprendizaje*, Sección “Other sources of information” in *Manual de Aprendizaje*.

Fragmentos de código: Sección “Simultaneous notes” in *Fragmentos de código*.

## Resolución de las colisiones

Las cabezas de notas con igual duración se combinan automáticamente, pero las que tienen duraciones distintas no se combinan. Los silencios que se encuentran en el lado opuesto de una plica se desplazan verticalmente.

```
<<
{
  c8 d e d c d c4
  g'2 fis
} \\ {
  c2 c8. b16 c4
  e,2 r
} \\ {
  \oneVoice
  s1
  e8 a b c d2
}
>>
```



Las cabezas de notas diferentes se pueden combinar, con la excepción de blancas con negras:

```
<<
{
  \mergeDifferentlyHeadedOn
  c8 d e d c d c4
  g'2 fis
} \\ {
  c2 c8. b16 c4
  e,2 r
} \\ {
  \oneVoice
  s1
  e8 a b c d2
}
>>
```



Se pueden combinar cabezas con puntillos diferentes:

```
<<
{
  \mergeDifferentlyHeadedOn
  \mergeDifferentlyDottedOn
  c8 d e d c d c4
  g'2 fis
} \\\ {
  c2 c8. b16 c4
  e,2 r
} \\\ {
  \oneVoice
  s1
  e8 a b c d2
}
>>
```



La colisión del segundo compás se produce porque `merge-differently-headed` no es capaz de completar satisfactoriamente la combinación cuando se encuentran tres o más notas alineadas en la misma columna (de hecho, se obtiene una advertencia por este motivo). Para conseguir que la combinación funcione adecuadamente, aplique un `\shift` 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 `merge-differently-headed` hace su trabajo correctamente.

```
<<
{
  \mergeDifferentlyHeadedOn
  \mergeDifferentlyDottedOn
  c8 d e d c d c4
  \shiftOn
  g'2 fis
} \\\ {
  c2 c8. b16 c4
  e,2 r
} \\\ {
  \oneVoice
  s1
  e8 a b c d2
}
>>
```



Las instrucciones `\shiftOn`, `\shiftOnn` y `\shiftOnnn` especifican el grado en que se deben desplazar los acordes de la voz en curso. Las voces exteriores (normalmente: la voz uno y la voz dos) tienen `\shiftOff` (desplazamiento desactivado), mientras que las voces interiores (tres y

cuatro) tienen `\shiftOn` (desplazamiento activado). `\shiftOnn` y `\shiftOnnn` definen niveles de desplazamiento más grandes.

Sólo se combinan las notas si tienen la plica en direcciones opuestas (o sea en las *Voices* 1 y 2).

## 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. 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
  <<
    { \voiceOne
      a4. a8
      e'4 e4. e8
      f4 d4. c8
    } \ {
      \voiceThree
      f,2
      bes4 a2
      a4 s2
    } \ {
      \voiceFive
      s2
      g4 g2
      f4 f2
    } \ {
      \voiceTwo
      d2
      d4 cis2
      d4 bes2
    }
  >>
}
```



*Forzar el desplazamiento horizontal de las notas*

Cuando el motor de tipografiado no es capaz de todo, se puede usar la propiedad `force-hshift` del objeto `NoteColumn` para sobrescribir decisiones de tipografiado. 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 “polifonía” in \*Glosario Musical\*](#).

Manual de aprendizaje: [Sección “Multiple notes at once” in \*Manual de Aprendizaje\*](#), [Sección “Voices contain music” in \*Manual de Aprendizaje\*](#), [Sección “Collisions of objects” in \*Manual de Aprendizaje\*](#).

Fragmentos de código: [Sección “Simultaneous notes” in \*Fragmentos de código\*](#).

Referencia de funcionamiento interno: `NoteColumn`, `NoteCollision`, `RestCollision`.

## Advertencias y problemas conocidos

Cuando se emplea `merge-differently-headed` con una corchea (o una nota más corta) con la plica hacia arriba, y una blanca con la plica hacia abajo, la corchea recibe un desplazamiento ligeramente incorrecto a causa de la anchura diferente del símbolo de la cabeza de la blanca.

No está claro en qué circunstancias se pueden combinar cabezas de nota distintas que tienen también distinto puntillo.

No están contemplados los acordes en que la misma nota se presenta con diferentes alteraciones accidentales dentro del mismo. En este caso se recomienda usar la transcripción enarmónica, o usar la notación especial de racimos (véase [\[Racimos \(clusters\)\]](#), página 85).

## Combinación automática de las partes

La combinación automática de particellas se usa para mezclar dos partes musicales sobre un pentagrama. Tiene como objetivo el tipografiado de partituras orquestales. Cuando las dos partes son idénticas durante un período de tiempo, sólo se presenta una de ellas. En los lugares en que las dos partes son diferentes, se tipografían como voces separadas, y las direcciones de las plicas se establecen de forma automática. También las partes de solo y *a due* quedan identificadas y es posible marcarlas.

La sintaxis para la combinación de las partes es:



```
\partcombine expresión_musical_1 expresión_musical_2
```

El ejemplo siguiente ejemplifica la funcionalidad básica del combinador de partes: poner las partes en un solo pentagrama, y establecer las direcciones de las plicas y la polifonía. Se utilizan las mismas variables para las partes independientes y el pentagrama combinado.

```
instrumentoUno = \relative c' {
  c4 d e f
  R1
  d'4 c b a
  b4 g2 f4
  e1
}
```

```
instrumentoDos = \relative g' {
  R1
  g4 a b c
  d c b a
  g f( e) d
  e1
}
```

```
<<
  \new Staff \instrumentoUno
  \new Staff \instrumentoDos
  \new Staff \partcombine \instrumentoUno \instrumentoDos
>>
```



Las notas del tercer compás aparecen solamente una vez a pesar de que se han especificado en las dos partes. 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*) siempre 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 predeterminado “a2”.

Los dos argumentos de `\partcombine` se interpretan como contextos de *Voice*. Si se están usando octavas relativas, se debe especificar `\relative` para ambas expresiones musicales, 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*.

## 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.

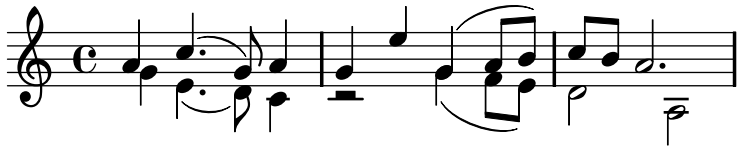


Este fragmento de código presenta las tres formas en que se pueden imprimir dos partes sobre un solo pentagrama: polifonía estándar, `\partcombine` sin textos, y `\partcombine` con textos.

```
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 {
        \set Staff.instrumentName = "Standard polyphony  "
        << \musicUp \\\musicDown >>
      }
      \new Staff \with { printPartCombineTexts = ##f } {
        \set Staff.instrumentName = "PartCombine without texts  "
        \partcombine \musicUp \musicDown
      }
      \new Staff {
        \set Staff.instrumentName = "PartCombine with texts  "
        \partcombine \musicUp \musicDown
      }
    >>
  >>
  \layout {
    indent = 6.0\cm
    \context {
      \Score
      \override SystemStartBar #'collapse-height = #30
    }
  }
}
```

```
}
}
```

Standard polyphony	
PartCombine without texts	
PartCombine with texts	

### *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.soloIIText = #"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” in *Glosario Musical*, Sección “parte” in *Glosario Musical*.

Referencia de la notación: Sección 1.6.2 [Escritura de las particellas], página 101.

Fragmentos de código: Sección “Simultaneous notes” in *Fragmentos de código*.

Referencia de funcionamiento interno: PartCombineMusic, Voice.

## Advertencias y problemas conocidos

Si `\printPartCombineTexts` está establecido, cuando las dos voces tocan y terminan las mismas notas, el combinador de partes puede tipografiar **a2** más de una vez en el mismo compás.

`\partcombine` no puede estar dentro de `\times`.

`\partcombine` no puede estar dentro de `\relative`.

Internamente, el `\partcombine` interpreta los dos argumentos como Voces llamadas **one** y **two**, y después decide cuándo se pueden combinar las partes. Como consecuencia, si los argumentos cambian a contextos de **Voice** con nombres diferentes, los eventos que contienen se ignorarán.

## Escribir música en paralelo

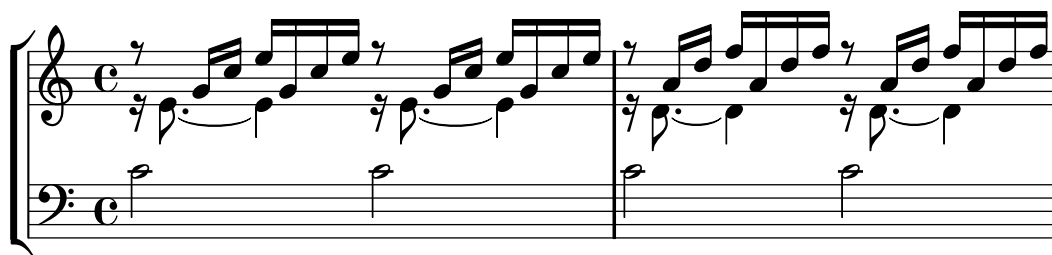
La música para varias partes se puede intercalar dentro del código de entrada. La función `\parallelMusic` admite una lista que contiene los nombres de las variables que se van a crear, y una expresión musical. El contenido de los compases alternativos extraídos de la expresión se convierten en el valor 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) {
  % Compás 1
  r8 g'16 c'' e'' g' c'' e'' r8 g'16 c'' e'' g' c'' e'' |
  r16 e'8.~ e'4 r16 e'8.~ e'4 |
  c'2 c'2 |

  % Compás 2
  r8 a'16 d'' f'' a' d'' f'' r8 a'16 d'' f'' a' d'' f'' |
  r16 d'8.~ d'4 r16 d'8.~ d'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) {
```

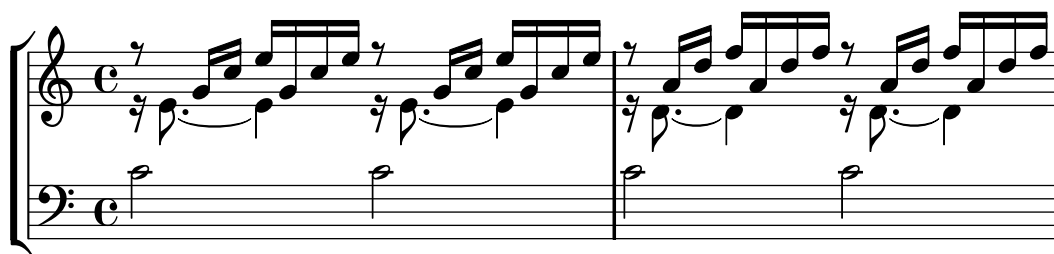
```

% Compás 1
r8 g16 c e g, c e r8 g,16 c e g, c e |
r16 e8.~ e4          r16 e8.~ e4          |
c2                  c                    |

% Compás 2
r8 a,16 d f a, d f r8 a,16 d f a, d f |
r16 d8.~ d4          r16 d8.~ d4          |
c2                  c                    |

}
\new StaffGroup <<
  \new Staff << \relative c'' \voiceA \\ \relative c' \voiceB >>
  \new Staff \relative c' { \clef bass \voiceC }
>>

```



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) {
  % Compás 1
  a8    b    c    d    |
  d4          e    |
  c16 d e fis d e fis g |
  a4          a    |

  % Compás 2
  e8    fis g    a    |
  fis4          g    |
  e16 fis g a fis g a b |
  a4          a    |

  % Compás 3 ...
}

\score {
  \new PianoStaff <<
    \new Staff {
      \global

```

```

    <<
      \relative c'' \voiceA
      \\
      \relative c' \voiceB
    >>
  }
  \new Staff {
    \global \clef bass
    <<
      \relative c \voiceC
      \\
      \relative c \voiceD
    >>
  }
  >>
}

```

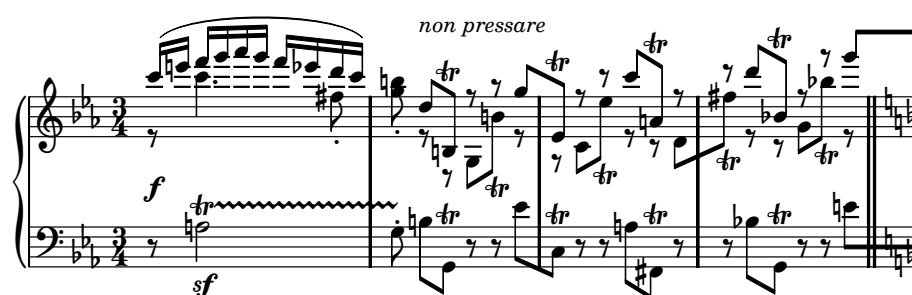


## Véase también

Manual de aprendizaje: [Sección “Organizing pieces with variables”](#) in *Manual de Aprendizaje*

Fragmentos de código: [Sección “Simultaneous notes”](#) in *Fragmentos de código*.

## 1.6 Notación de los pentagramas





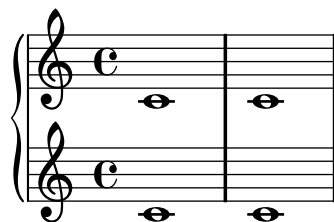
### 1.6.1 Impresión de los pentagramas

#### 1.6.1.1 Delimitadores del comienzo de un sistema

Muchas partituras constan de más de un pentagrama. Estos pentagramas se pueden unir de cuatro formas distintas:

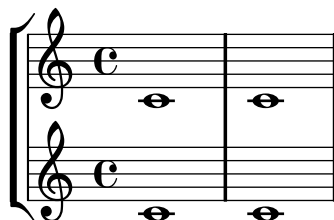
- El grupo comienza con una llave a la izquierda y las líneas de compás están conectadas. Esto se hace con el contexto `GrandStaff`.

```
\new GrandStaff
\relative <<
  \new Staff { c1 c }
  \new Staff { c c }
>>
```



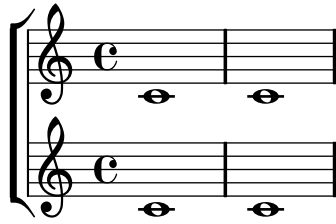
- El grupo comienza con un corchete recto y las líneas de compás están conectadas. Esto se hace con el contexto `StaffGroup`

```
\new StaffGroup
\relative <<
  \new Staff { c1 c }
  \new Staff { c c }
>>
```



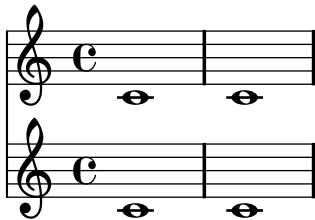
- El grupo comienza con un corchete recto pero las líneas de compás no están conectadas. Esto se hace con el contexto `ChoirStaff`.

```
\new ChoirStaff
\relative <<
  \new Staff { c1 c }
  \new Staff { c c }
>>
```



- El grupo comienza con una línea vertical. Las líneas de compás no están conectadas. Ésta es la opción por defecto para la partitura.

```
\relative <<
  \new Staff { c1 c }
  \new Staff { c c }
>>
```



## Véase también

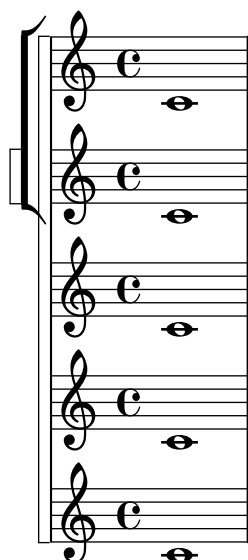
Las barras de compás al principio de cada sistema son `SystemStartBar`, `SystemStartBrace` y `SystemStartBracket`. En cada contexto solamente se crea uno de estos tipos, y ese tipo está determinado por la propiedad `systemStartDelimiter`.

## Propiedades trucadas frecuentemente

Los delimitadores del comienzo del sistema se pueden anidar muchas veces:

```
\new StaffGroup
\relative <<
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare (SystemStartBracket a (SystemStartSquare b)) d)
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
>>
```





### 1.6.1.2 El símbolo del pentagrama

Las notas, signos dinámicos, etc. están agrupados con un conjunto de líneas horizontales que reciben el nombre de pentagrama o pauta ('staff', en plural 'staves'). En LilyPond, estas líneas se dibujan utilizando un objeto de presentación separado que se llama el **staff symbol** (el símbolo del pentagrama).

El símbolo del pentagrama se puede ajustar en el número, grosor y separación de las líneas, usando las propiedades.

Además, los pentagramas se pueden iniciar e interrumpir libremente. Esto se hace con `\startStaff` y `\stopStaff`.

```
b4 b
\override Staff.StaffSymbol #'line-count = 2
\stopStaff \startStaff
b b
\revert Staff.StaffSymbol #'line-count
\stopStaff \startStaff
b b
```



En combinación con los pentagramas ocultos o franceses, esto se puede usar para componer tipográficamente secciones de *ossia*. He aquí un ejemplo:

## Véase también

Referencia del programa: `StaffSymbol`.

Fragmentos de código: [Sección “Staff notation”](#) in *Fragmentos de código*.

### 1.6.1.3 Ocultar pentagramas

En las partituras orquestales, generalmente se eliminan las líneas de pauta que contienen sólo silencios; así se ahorra espacio. Este estilo se llama ‘Partitura francesa’. Para `Lyrics`, `ChordNames` y `FiguredBass`, esta opción está activada por defecto. Cuando las líneas de estos contextos resultan vacías después del proceso de cálculo de los saltos de línea, se eliminan.

Para las pautas normales hay un contexto especializado `Staff`, que hace lo mismo: las pautas que no contienen nada (o solamente silencios multicompas) se quitan. La definición de contexto se almacena dentro de la variable `\RemoveEmptyStaffContext`. Observe cómo el segundo pentagrama del siguiente ejemplo desaparece en la segunda línea

```
\layout {
  \context { \RemoveEmptyStaffContext }
}

{
  \relative c' <<
    \new Staff { e4 f g a \break c1 }
    \new Staff { c4 d e f \break R1 }
  >>
}
```



El primer sistema muestra todos los pentagramas al completo. Si queremos que los pentagramas vacíos se quiten también del primer sistema, podemos establecer `remove-first` al valor `true` en `VerticalAxisGroup`.

```
\override Score.VerticalAxisGroup #'remove-first = ##t
```

Para quitar otros tipos de contextos, utilice `\AncientRemoveEmptyStaffContext` o `\RemoveEmptyRhythmicStaffContext`.

Otra aplicación es hacer secciones de ossia, es decir, melodías alternativas sobre un trozo de pentagrama separado, con la ayuda de los pentagramas a la francesa.

### 1.6.2 Escritura de las particellas

#### Indicaciones metronómicas

Las indicaciones metronómicas se pueden escribir como sigue

```
\tempo duración = pulsos_por_minuto
```

En la salida MIDI, se interpretan como un cambio de tempo. En la salida visual, se imprime una indicación metronómica

```
\tempo 8.=120 c' '1
```



## Propiedades trucadas frecuentemente

Para cambiar el tempo en la salida MIDI sin imprimir nada en la salida visual, haga invisible la indicación metronómica

```
\once \override Score.MetronomeMark #'transparent = ##t
```

Para imprimir otras indicaciones metronómicas, use estas instrucciones de marcado

```
c4^\markup {
  (
    \smaller \general-align #Y #DOWN \note #"16." #1
  =
    \smaller \general-align #Y #DOWN \note #"8" #1
  ) }

```



Consulte [Sección 1.8.2 \[Formatear el texto\]](#), página 125 para ver más detalles.

## Véase también

Referencia del programa: `MetronomeMark`.

## Advertencias y problemas conocidos

No se comprueba la existencia de colisiones. Si tenemos notas por encima de la línea superior del pentagrama (o notas con articulaciones, ligaduras, texto, etc), la indicación metronómica se podría imprimir encima de los símbolos musicales. Si esto ocurre, aumente el relleno adicional de la indicación metronómica para colocarla más lejos del pentagrama.

```
\override Score.MetronomeMark #'padding = #2.5
```

## Nombres de instrumentos

En una partitura orquestal, los nombres de los instrumentos se imprimen en el lado izquierdo de los pentagramas.

Esto se puede conseguir estableciendo `Staff.instrumentName` y `Staff.shortInstrumentName`, o bien `PianoStaff.instrumentName` y `PianoStaff.shortInstrumentName`. Esto imprimirá el texto antes del principio del pentagrama. Para el primer pentagrama se usa `instrumentName`, y para los siguientes se usa `shortInstrumentName`.

```
\set Staff.instrumentName = "Ploink "
\set Staff.shortInstrumentName = "Plk "
c1
\break
c''
```





También podemos usar elementos de marcado de texto para construir nombres de instrumento más complicados, por ejemplo

```
\set Staff.instrumentName = \markup {
  \column { "Clarinetti"
    \line { "in B" \smaller \flat } } }
c''1
```



Si queremos centrar los nombres de instrumento, hemos de centrarlos todos

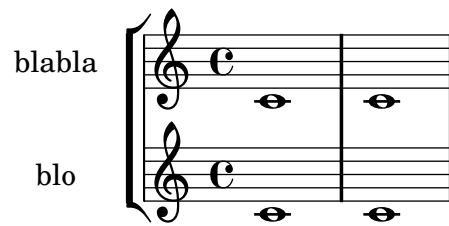
```
{ <<
\new Staff {
  \set Staff.instrumentName = \markup {
    \center-column { "Clarinetti"
      \line { "in B" \smaller \flat } } }
  c''1
}
\new Staff {
  \set Staff.instrumentName = \markup{ \center-align { Vibraphone } }
  c''1
}
>>
}
```



Para nombres de instrumento más largos, puede ser útil aumentar el valor de `indent` dentro del bloque `\layout`.

Para centrar los nombres de instrumento y al mismo tiempo dejar espacio adicional a la derecha,

```
\new StaffGroup \relative
<<
  \new Staff {
    \set Staff.instrumentName = \markup { \hcenter-in #10 "blabla" }
    c1 c1
  }
  \new Staff {
    \set Staff.instrumentName = \markup { \hcenter-in #10 "blo" }
    c1 c1
  }
>>
```



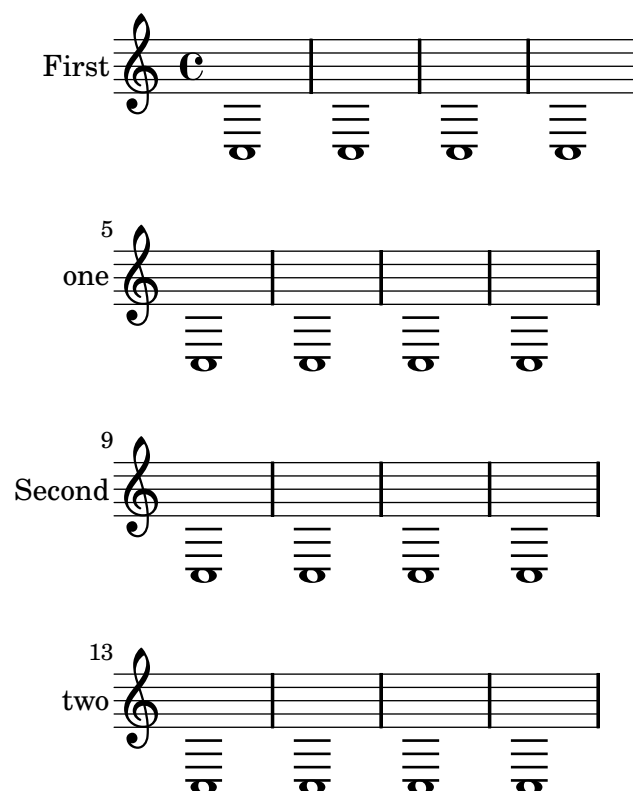
Para añadir nombres de instrumento a otros contextos (como `GrandStaff`, `ChoirStaff` o `StaffGroup`), debemos añadir el grabador a dicho contexto.

```
\layout{
  \context {\GrandStaff \consists "Instrument_name_engraver"}
}
```

Se puede encontrar más información sobre cómo añadir y suprimir grabadores en [Sección 5.1.3 \[Modificar los complementos \(plug-ins\) de contexto\]](#), página 216.

Los nombres de instrumento se pueden cambiar en mitad de una pieza,

```
\set Staff.instrumentName = "First"
\set Staff.shortInstrumentName = "one"
c1 c c c \break
c1 c c c \break
\set Staff.instrumentName = "Second"
\set Staff.shortInstrumentName = "two"
c1 c c c \break
c1 c c c \break
```



Véase también

Referencia del programa: `InstrumentName`.

## Citar otras voces

Con las citas, se pueden insertar fragmentos de otras partes directamente dentro de una parte. Antes de que una parte pueda ser citada, se debe marcar especialmente como citable. Esto se hace con la instrucción `\addQuote`.

```
\addQuote nombre música
```

Aquí, *nombre* es una cadena identificativa. La *música* es cualquier tipo de música. He aquí un ejemplo de `\addQuote`

```
\addQuote clarinete \relative c' {
  f4 fis g gis
}
```

Esta orden se debe escribir en el nivel más alto, es decir, fuera de cualquier bloque de música.

Después de la llamada a `\addQuote`, la cita se puede hacer con `\quoteDuring` o con `\cueDuring`,

```
\quoteDuring #nombre música
```

En el transcurso de una parte, un fragmento de música se puede citar con la instrucción `\quoteDuring`.

```
\quoteDuring #"clarinete" { s2. }
```

Esto citaría tres negras (la duración de la blanca con puntillo `s2.`) de la voz `clarinete` que se escribió previamente.

De manera más precisa, toma el paso de tiempo actual de la parte que se está imprimiendo, y extrae las notas en el punto correspondiente de la voz `\addQuote`ada. Por tanto, el argumento de `\addQuote` debe ser la parte completa de la voz que se va a citar, incluidos los silencios del principio, si los hay.

Las citas tienen en cuenta tanto la transposición del instrumento origen como la del de destino, si se han especificado utilizando la instrucción `\transposition`.

```
\addQuote clarinet \relative c' {
  \transposition bes
  f4 fis g gis
}

{
  e'8 f'8 \quoteDuring #"clarinet" { s2 }
}
```



Los tipos de eventos que aparecen en las notas guía se pueden recortar con la propiedad `quotedEventTypes`. El valor por defecto es `(note-event rest-event)`, lo que significa que sólo las notas y los silencios de la voz citada acaban en el `\quoteDuring`. Al establecer

```
\set Staff.quotedEventTypes =
  #'(note-event articulation-event dynamic-event)
```

se citarán las notas (pero no los silencios), además de las inscripciones textuales y de dinámica.

## Advertencias y problemas conocidos

Sólo los contenidos de la primera *Voice* que aparece dentro de una instrucción `\addQuote` se tendrán en consideración para su cita, así que *música* no puede contener sentencias `\new` y `\context` *Voice* que podrían producir el cambio a una Voz diferente.

La cita de notas de adorno es defectuosa e incluso puede causar una caída del programa LilyPond.

La cita de tresillos anidados puede dar como resultado una notación pobre.

## Véase también

En este manual: [\[Transposición de los instrumentos\]](#), página 17.

Fragmentos de código: [Sección “Staff notation”](#) in *Fragmentos de código*.

Referencia del programa: `QuoteMusic`.

## Formateo de las notas de aviso

La sección anterior trata de la inserción de notas que proceden de otra voz. Hay una función musical más avanzada que se llama `\cueDuring`, que facilita el formateo de las notas guía.

La sintaxis es

```
\cueDuring #nombre #arriba_abajo música
```

Esto insertará las notas de la parte *nombre* en una *Voice* llamada *cue*. Esto ocurre de forma simultánea con *música*, que por lo general es un silencio. Cuando comienzan las notas guía, el pentagrama se hace polifónico por un momento. El argumento *arriba\_abajo* determina si las notas guía se deben tipografiar como primera o como segunda voz.

```
smaller = {
  \set fontSize = #-2
  \override Stem #'length-fraction = #0.8
  \override Beam #'thickness = #0.384
  \override Beam #'length-fraction = #0.8
}

\addQuote clarinet \relative {
  R1*20
  r2 r8 c' f f
}

\new Staff \relative <<

% setup a context for cue notes.
\new Voice = "cue" { \smaller \skip 1*21 }

\set Score.skipBars = ##t

\new Voice {
  R1*20
  \cueDuring #"clarinet" #UP {
    R1
  }
  g4 g2.
```

```
}
>>
```



Presentamos a continuación un par de consejos para culminar con éxito la escritura de notas guía.

- Las notas guía tienen un tamaño tipográfico menor.
- la parte citada se marca con el instrumento que toca la cita.
- cuando la parte original vuelve, se debe marcar con el nombre del instrumento original.

Además cualquier otro cambio introducido por la parte citada se debe deshacer. Por ejemplo, si el instrumento citado toca en una clave distinta, la clave original se debe hacer constar de nuevo.

La macroinstrucción `\transposedCueDuring` es útil para añadir notas guía a instrumentos que utilizan un rango de octava completamente distinto (por ejemplo, tener una cita del piccolo dentro de la parte de contrafagot).

```
picc = \relative c''' {
  \clef "treble^8"
  R1 |
  c8 c c e g2 |
  a4 g g2 |
}
\addQuote "picc" { \picc }

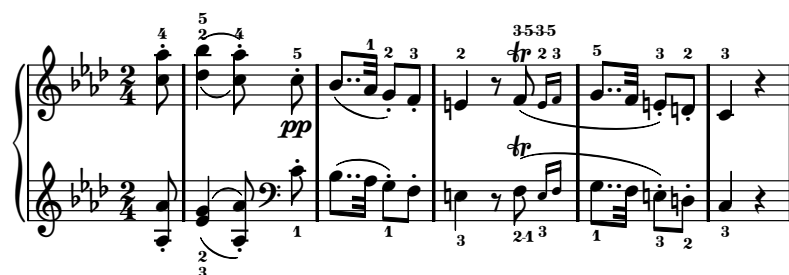
cbsn = \relative c, {
  \clef "bass_8"
  c4 r g r
  \transposedCueDuring #"picc" #UP c,, { R1 } |
  c4 r g r |
}

<<
  \context Staff = "picc" \picc
  \context Staff = "cbsn" \cbsn
>>
```





## 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

Se puede alterar el tamaño de la fuente tipográfica de los elementos de notación. Esto no cambia el tamaño de los símbolos variables, como las barras de corchea o las ligaduras.

**Nota:** Para los tamaños de tipografía del **texto**, consulte [Sección B.7 \[Text markup list commands\]](#), página 294.

```
\huge
c4.-> d8---3
\large
c4.-> d8---3
\normalsize
c4.-> d8---3
\small
c4.-> d8---3
\tiny
c4.-> d8---3
\teeny
c4.-> d8---3
```



Internamente, esto establece un valor para la propiedad `fontSize`. Esto, a su vez, hace que se establezca el valor de la propiedad `font-size` de todos los objetos de presentación. El valor de `font-size` es un número que indica el tamaño con relación al tamaño estándar para la altura vigente del pentagrama. Cada unidad hacia arriba es un incremento de un 12% del tamaño de la fuente, aproximadamente. Seis unidades son exactamente un factor de dos. La función de Scheme `magstep` convierte un número de tamaño `font-size` a un factor de escalado. La propiedad `font-size` también se puede establecer directamente, de forma que queden afectados solamente determinados objetos gráficos.

```
\set fontSize = #3
c4.-> d8---3
```

```
\override NoteHead #'font-size = #-4
c4.-> d8---3
\override Script #'font-size = #2
c4.-> d8---3
\override Stem #'font-size = #-5
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 10 puntos.

La propiedad `font-size` sólo se puede establecer en objetos gráficos que utilicesn fuentes tipográficas. Éstos son los que contemplan el interfaz de presentación `font-interface`.

## Instrucciones predefinidas

`\teeny` (enano), `\tiny` (muy pequeño), `\small` (pequeño), `\normalsize` (normal), `\large` (grande), `\huge` (enorme).

## Véase también

Fragmentos de código: [Sección “Editorial annotations” in \*Fragmentos de código\*](#).

Referencia de funcionamiento interno: `font-interface`.

## Indicaciones de digitación

Las instrucciones de digitación se pueden introducir usando *nota-dígito*:

```
c4-1 d-2 f-4 e-3
```



Para los cambios de dedo se pueden usar elementos de marcado de texto.

```
c4-1 d-2 f-4 c^\markup { \finger "2 - 3" }
```



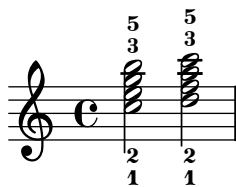
Puede usar la articulación de pulgar para indicar que una nota se debe tocar con el pulgar (p.ej. en música de cello).

```
<a_\thumb a'-3>8 <b_\thumb b'-3>
```



Las digitaciones para los acordes también se pueden añadir a las notas individuales del acorde escribiéndolas después de las alturas.

```
<c-1 e-2 g-3 b-5>2 <d-1 f-2 a-3 c-5>
```

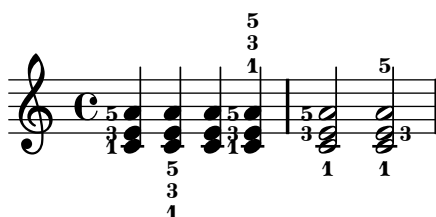


Las indicaciones de digitación se pueden situar manualmente encima o debajo del pentagrama, véase [Sección 5.4.2 \[Dirección y posición\]](#), página 230.

## Fragmentos de código seleccionados

Se puede controlar con precisión la colocación de los números de digitación.

```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(right)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left down)
  <c-1 e-3 a-5>2
  \set fingeringOrientations = #'(up right down)
  <c-1 e-3 a-5>2
}
```



## Véase también

Referencia de la notación: [Sección 5.4.2 \[Dirección y posición\]](#), página 230

Fragmentos de código: [Sección “Editorial annotations”](#) in *Fragmentos de código*.

Referencia de funcionamiento interno: `FingeringEvent`, `fingering-event`, `Fingering_engraver`, `New_fingering_engraver`, `Fingering`.

## Notas ocultas

Las notas ocultas (o invisibles, o transparentes) pueden ser de utilidad en la preparación de ejercicios de teoría o de composición.

```
c4 d
\hideNotes
e4 f
\unHideNotes
g a
\hideNotes
b
\unHideNotes
c
```



## Instrucciones predefinidas

`\hideNotes`, `\unHideNotes`

## Véase también

Fragmentos de código: [Sección “Editorial annotations”](#) in *Fragmentos de código*.

Referencia de funcionamiento interno: `Note_spacing_engraver`, `NoteSpacing`.

## Colorear objetos

Se pueden asignar colores a los objetos individuales. Los nombres de color válidos se encuentran relacionados en la [Sección B.3 \[Lista de colores\]](#), página 242.

```
\override NoteHead #'color = #red
c4 c
\override NoteHead #'color = #(x11-color 'LimeGreen)
d
\override Stem #'color = #blue
e
```



Se puede acceder al espectro completo de colores definido para X11 usando la función de Scheme `x11-color`. Esta función acepta un único argumento; éste puede ser un símbolo de la forma `'FulanoMengano` o una cadena de la forma `"FulanoMengano"`. La primera forma es más rápida de escribir y también más eficiente. Sin embargo, al usar la segunda forma es posible acceder a los colores de X11 por medio de la forma del nombre que tiene varias palabras.

Si `x11-color` no entiende el parámetro, el color predeterminado que se devuelve es el negro.

```
\override Staff.StaffSymbol #'color = #(x11-color 'SlateBlue2)
\set Staff.instrumentName = \markup {
  \with-color #(x11-color 'navy) "Clarinet"
}
```

```
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
% esto no tiene sentido, intencionalmente. Observe que las plicas siguen negras
\override Stem #'color = #(x11-color 'Boggle)
b2 cis
```



Se pueden especificar colores RGB exactos utilizando la función de Scheme `rgb-color`.

```
\override Staff.StaffSymbol #'color = #(x11-color 'SlateBlue2)
\set Staff.instrumentName = \markup {
  \with-color #(x11-color 'navy) "Clarinet"
}
```

```
% black
\override Stem #'color = #(rgb-color 0 0 0)
gis8 a
% white
\override Stem #'color = #(rgb-color 1 1 1)
gis8 a
% dark blue
\override Stem #'color = #(rgb-color 0 0 0.5)
gis4 a
```



Véase también

Referencia de la notación: Sección B.3 [Lista de colores], página 242, Sección 5.3.5 [La instrucción `tweak`], página 227.

Fragmentos de código: Sección “Editorial annotations” in *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 normales (o sea: `#blue`, azul, `#green`, verde, `#red`, rojo).

Las notas de un acorde no se pueden colorear con `\override`; en su lugar utilice `\tweak`. Consulte Sección 5.3.5 [La instrucción `tweak`], página 227 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.

```
c2 \parenthesize d
c2 \parenthesize <c e g>
```



Los objetos que no son notas también se pueden poner entre paréntesis.

```
c2-\parenthesize -. d
c2 \parenthesize r
```



## Véase también

Fragmentos de código: Sección “Editorial annotations” in *Fragmentos de código*.

Referencia de funcionamiento interno: `Parenthesis_engraver`, `ParenthesesItem`, `parentheses-interface`.

## 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.

## Instrucciones predefinidas

`\stemUp`, `\stemDown`, `\stemNeutral`.

## Fragmentos de código seleccionados

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
}
```



## Véase también

Referencia de la notación: [Sección 5.4.2 \[Dirección y posición\]](#), página 230.

Fragmentos de código: [Sección “Editorial annotations”](#) in *Fragmentos de código*.

Referencia de funcionamiento interno: `Stem_engraver`, `Stem`, `stem-interface`.

## 1.7.2 Fuera del pentagrama

Esta sección trata sobre cómo enfatizar elementos que están dentro del pentagrama, desde fuera del pentagrama.

## 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" }
{
  \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, `balloonText` y `balloonGrobText`; 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.

## Véase también

Fragmentos de código: [Sección “Editorial annotations”](#) in *Fragmentos de código*.

Referencia de funcionamiento interno: `Balloon_engraver`, `BalloonTextItem`, `balloon-interface`.

## Líneas de rejilla

Se pueden dibujar líneas verticales entre los pentagramas sincronizadas con las notas.

```
\layout {
  \context {
    \Staff
    % preparar la rejilla:
    \consists "Grid_point_engraver"
    % ajustar el intervalo de la cuadrícula a una negra:
    gridInterval = #(ly:make-moment 1 4)
  }
  \context {
    \Score
    % por omisión las líneas de rejilla están centradas horizontalmente
    % debajo y a la izquierda de la cabeza; las líneas parten de la tercera
    % línea del pentagrama
    \consists "Grid_line_span_engraver"
  }
}

\score {
  \new ChoirStaff <<
    \new Staff \relative c' {
      \stemUp
      c4. d8 e8 f g4
    }
    \new Staff \relative c {
      \clef bass
      \stemDown
      c4 g' f e
    }
  >>
}
```



## Fragmentos de código seleccionados

Se puede cambiar el aspecto de las líneas de rejilla sobrescribiendo algunas de sus propiedades.



```

\layout {
  \context {
    \Staff
    % set up grids
    \consists "Grid_point_engraver"
    % set the grid interval to one quarter note
    gridInterval = #(ly:make-moment 1 4)
  }
}

\new Score \with {
  \consists "Grid_line_span_engraver"
  % this moves them to the right half a staff space
  \override NoteColumn #'X-offset = #-0.5
}

\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'
      \once \override Score.GridLine #'thickness = #3.0
      f
      \once \override Score.GridLine #'thickness = #5.0
      e
    }
  }
}
>>

```



## Véase también

Fragmentos de código: [Sección “Editorial annotations”](#) in *Fragmentos de código*.

Referencia de funcionamiento interno: `Grid_line_span_engraver`, `Grid_point_engraver`, `GridLine`, `GridPoint`, `grid-line-interface`, `grid-point-interface`.

## 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 {
    \Staff
    \consists "Horizontal_bracket_engraver"
  }
}
\relative c'' {
  c2\startGroup
  d\stopGroup
}
```



Los corchetes de análisis se pueden anidar.

```
\layout {
  \context {
    \Staff
    \consists "Horizontal_bracket_engraver"
  }
}
\relative c'' {
  c4\startGroup\startGroup
  d4\stopGroup
  e4\startGroup
  d4\stopGroup\stopGroup
}
```



## Véase también

Fragmentos de código: [Sección “Editorial annotations”](#) in *Fragmentos de código*.

Referencia de funcionamiento interno: `Horizontal_bracket_engraver`, `HorizontalBracket`, `horizontal-bracket-interface`, `Staff`.

## 1.8 Texto

The image shows a musical score in 3/4 time, key of B-flat major. The score includes several text annotations: *p con amabilità*, *ten.*, *tranqu.*, *ten. dolce*, *cantabile, con intimissimo sentimento, ma sempre molto dolce e semplice*, *non staccato*, *molto p, sempre tranquillo ed egualmente, non rubato*, and *Red.* (likely a typo for *Red.* or *Red.*).

Esta sección explica cómo incluir texto (con diversos estilos de formateo) en las partituras. Ciertos elementos de texto de los que no nos ocupamos aquí se tratan en otras secciones específicas: [Sección 2.1 \[Música vocal\]](#), página 130, [Sección 3.2 \[Títulos y cabeceras\]](#), página 199.

**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 [Sección 3.3.3 \[Codificación del texto\]](#), página 206.

### 1.8.1 Escritura del texto

Esta sección presenta las distintas formas de añadir texto a una partitura.

#### 1.8.1.1 Guiones de texto

Es posible añadir cadenas de texto arbitrarias a una partitura, como se muestra en el ejemplo siguiente. Tales indicaciones se pueden colocar también manulamente encima o debajo del pentagrama utilizando la sintaxis simple que se describe en [Sección 5.4.2 \[Dirección y posición\]](#), página 230.

`d8~"pizz." e f g a4-"scherz." f`

The image shows a musical score in 4/4 time, key of C major. The score includes the text annotations *pizz.* and *scherz.* placed above and below the staff respectively.

En LilyPond, estas cadenas de texto se llaman objetos de `markup` o de marcado. 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 125.

```
d8^\markup { \italic pizz. } e f g
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.

```
d8^"pizz." e f g \textLengthOn a4_"scherzando" f
```



## Instrucciones predefinidas

`\textLengthOn`, `\textLengthOff`

## Véase también

Referencia de la notación: [Sección 1.8.2 \[Formatear el texto\]](#), página 125, [Sección 5.4.2 \[Dirección y posición\]](#), página 230.

Fragmentos de código: [Sección “Text” in Fragmentos de código](#).

Referencia de funcionamiento interno: `TextScript`.

## Advertencias y problemas conocidos

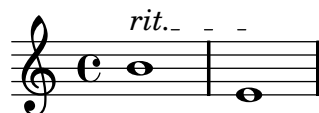
La comprobación necesaria para asegurar que las inscripciones de texto y la letra de las canciones se mantienen dentro de los márgenes, es una tarea computacional relativamente grande. Para acelerar el procesado, LilyPond no realiza estos cálculos de forma predeterminada; para habilitarlos, utilice

```
\override Score.PaperColumn #'keep-inside-line = ##t
```

### 1.8.1.2 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:

```
\override TextSpanner #'bound-details #'left #'text = "rit."
b1\startTextSpan
e,\stopTextSpan
```



La cadena de texto que imprimir se establece a través de propiedades de objeto. De forma pre-determinada 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 125.

```
\override TextSpanner #'bound-details #'left #'text =
  \markup { \upright "rit." }
b1\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 5.5.2 \[Estilos de línea\]](#), página 231.

## Instrucciones predefinidas

```
\textSpannerUp, \textSpannerDown, \textSpannerNeutral
```

## Véase también

Referencia de la notación: [Sección 5.5.2 \[Estilos de línea\]](#), página 231.

Fragmentos de código: [Sección “Text” in Fragmentos de código](#).

Referencia de funcionamiento interno: `TextSpanner`.

### 1.8.1.3 Indicaciones de texto

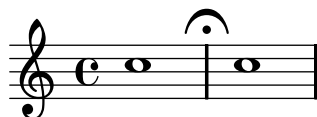
Se pueden añadir a la partitura varios elementos de texto con la sintaxis que se describe en [Sección 1.2.5.4 \[Llamadas de ensayo\]](#), página 57:

```
c4\mark "Allegro" c c c
```



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 125. Esto se puede usar para imprimir letreros como coda, segno o calderón, especificando el nombre del símbolo correspondiente:

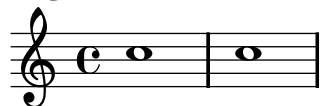
```
c1 \mark \markup { \musicglyph #"scripts.ufermata" }
c1
```



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 entra las notas. Si se especifican al principio de la partitura o en un salto de línea, las llamadas se imprimen al principio de la línea (o de la línea siguiente, en el caso de un salto de línea).

```
\mark "Allegro" c1
c\mark "assai" \break
c c
```

### Allegro



### assai



## Fragmentos de código seleccionados

### *Printing marks at the end of a line or a score*

Marks can be printed at the end of the current line, instead of the beginning of the following line. This is particularly useful when a mark has to be added at the end of a score – when there is no next line.

In such cases, the right end of the mark has to be aligned with the final bar line, as demonstrated on the second line of this example.

```
\relative c' ' {
  \override Score.RehearsalMark #'break-visibility = #begin-of-line-invisible
  g2 c
  d,2 a'
  \mark \default
  \break

  \override Score.RehearsalMark #'self-alignment-X = #RIGHT
  g2 b,
  c1 \bar "||"
  \mark "D.C. al Fine"
}
```



### *Alinear marcas con varios objetos de notación*

Si están especificadas, las marcas de texto se pueden alinear con objetos de notación distintos a las líneas divisorias. Entre estos objetos se encuentran **ambitus**, **breathing-sign**, **clef**, **custos**, **staff-bar**, **left-edge**, **key-cancellation**, **key-signature** y **time-signature**.

En estos casos, las marcas de texto se centran horizontalmente sobre el objeto, aunque esto se puede cambiar, como se muestra en la segunda línea de este ejemplo (en una partitura con varios pentagramas, se debe hacer este ajuste para todos los pentagramas).

```
\relative c' {
  e1

  % the RehearsalMark will be centered above the Clef
  \override Score.RehearsalMark #'break-align-symbols = #'(clef)
  \key a \major
  \clef treble
  \mark ""
  e

  % the RehearsalMark will be centered above the TimeSignature
  \override Score.RehearsalMark #'break-align-symbols = #'(time-signature)
  \key a \major
  \clef treble
  \time 3/4
  \mark ""
  e2.

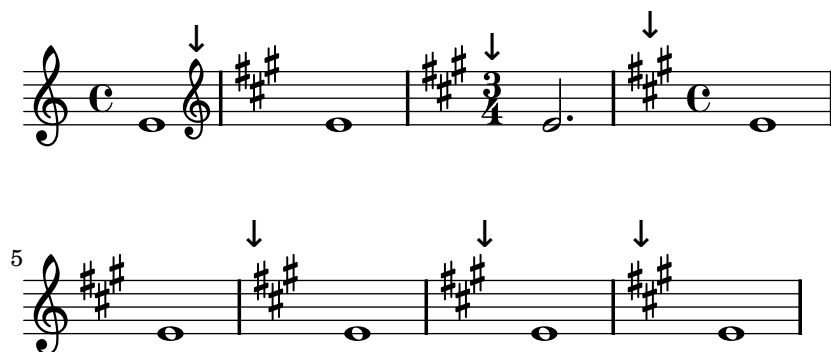
  % the RehearsalMark will be centered above the KeySignature
  \override Score.RehearsalMark #'break-align-symbols = #'(key-signature)
  \key a \major
  \clef treble
  \time 4/4
  \mark ""
  e1

  \break
  e

  % the RehearsalMark will be aligned with the left edge of the KeySignature
  \once \override Score.KeySignature #'break-align-anchor-alignment = #LEFT
  \mark ""
  \key a \major
  e

  % the RehearsalMark will be aligned with the right edge of the KeySignature
  \once \override Score.KeySignature #'break-align-anchor-alignment = #RIGHT
  \key a \major
  \mark ""
  e

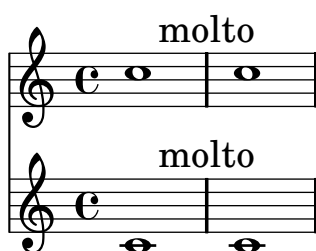
  % the RehearsalMark will be aligned with the left edge of the KeySignature
  % and then shifted right by 1 unit.
  \once \override Score.KeySignature #'break-align-anchor = #1
  \key a \major
  \mark ""
  e1
}
```



*Printing marks on every staff*

Although text marks are normally only printed above the topmost staff, they may also be printed on every staff.

```
{
  \new Score \with {
    \remove "Mark_engraver"
  }
  <<
    \new Staff \with {
      \consists "Mark_engraver"
    }
    { c'1 \mark "molto" c'1 }
    \new Staff \with {
      \consists "Mark_engraver"
    }
    { c'1 \mark "molto" c' }
  >>
}
```



## Véase también

Referencia de la notación: [Sección 1.2.5.4 \[Llamadas de ensayo\]](#), página 57, [Sección 1.8.2 \[Formatear el texto\]](#), página 125, [Sección B.4 \[La tipografía Feta\]](#), página 244.

Fragmentos de código: [Sección “Text” in Fragmentos de código](#).

Referencia de funcionamiento interno: [RehearsalMark](#).

## Advertencias y problemas conocidos

Si se introduce una llamada al final del último compás de la partitura (donde no hay línea siguiente), la llamada no se imprime en absoluto.



### 1.8.1.4 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 3.1.3 \[Estructura del archivo\]](#), página 199.

```
\markup {
  Tomorrow, and tomorrow, and tomorrow...
}
```

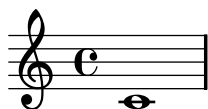
Tomorrow, and tomorrow, and tomorrow...

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 199.

```
\score {
  c'1
}
\markup {
  Tomorrow, and tomorrow, and tomorrow...
}
\score {
  c'1
}
```



Tomorrow, and tomorrow, and tomorrow...



Usando una sintaxis específica, los bloques de texto se pueden repartir entre varias páginas, posibilitando la impresión de documentos de texto o libros (y por tanto para usar LilyPond como un editor de textos). Esta sintaxis se describe en [Sección 1.8.2.6 \[Elementos de marcado de varias páginas\]](#), página 127.

## Instrucciones predefinidas

`\markup`, `\markuplines`

## Véase también

Referencia de la notación: [Sección 1.8.2 \[Formatear el texto\]](#), página 125, [Sección 3.1.3 \[Estructura del archivo\]](#), página 199, [Sección 3.1.2 \[Varias partituras en un libro\]](#), página 199, [Sección 1.8.2.6 \[Elementos de marcado de varias páginas\]](#), página 127.

Fragmentos de código: [Sección “Text” in Fragmentos de código](#).

Referencia de funcionamiento interno: `TextScript`.

## 1.8.2 Formatear el texto

Esta sección presenta los formateados básico y avanzado de texto, usando la sintaxis específica del modo de `\markup`.

### 1.8.2.1 Introducción al marcado de texto

Se usa un bloque `\markup` para tipografiar texto con una sintaxis ampliable específica que se denomina “modo de marcado”.

La sintaxis del marcado es similar a la sintaxis usual de LilyPond: una expresión `\markup` se encierra entre llaves `{ ... }`.

En el modo de marcado, las instrucciones específicas se introducen mediante la barra invertida `\`. Estas instrucciones sólo afectan a la primera expresión que venga a continuación.

Las expresiones de marcado también se pueden encerrar entre comillas `"..."`. Estas expresiones se escriben como cadenas de texto y no pueden contener expresiones anidadas ni otras instrucciones. Por tanto, generalmente se prefieren las llaves a las comillas; el ejemplo siguiente muestra las dos sintaxis.

```
e1-\markup "intenso"
a2^\markup { poco \italic più forte }
c e1
d2_\markup { \italic "string. assai" }
e
b1^\markup { \bold { molto \italic agitato } }
c
```



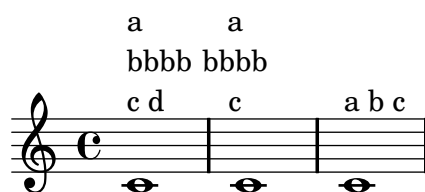
Los caracteres especiales como `\` y `#` se pueden imprimir fácilmente utilizando comillas. Las propias comillas sólo se imprimen cuando se preceden por una barra invertida:

```
\clef bass
a^\markup "##\ LEPORELLO \##"
a_\markup "Bravi! \"Cosa rara\"!"
r a8 d
cis a r4 r2
```



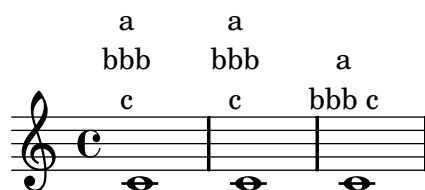
La forma en que están definidas las expresiones de marcado afectan a la manera en que se apilan, se centran o se alinean al usar las instrucciones descritas en [Sección 1.8.2.3 \[Alineación de texto\]](#), página 126.

```
c1^\markup { \column { a bbbb \line { c d } } }
c1^\markup { \center-column { a bbbb c } }
c1^\markup { \line { a b c } }
```



No se conservan como independientes aquellas listas de palabras que no estén encerradas entre comillas o precedidas de una instrucción previa. En el ejemplo siguiente, las dos primeras expresiones `\markup` son equivalentes:

```
c1^\markup { \center-column { a bbb c } }
c1^\markup { \center-column { a { bbb c } } }
c1^\markup { \center-column { a \line { 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 B.6 \[Text markup commands\]](#), página 258.

## Véase también

En este manual: [Sección B.6 \[Text markup commands\]](#), página 258.

Fragmentos de código: [Sección “Text” in Fragmentos de código](#).

Referencia de funcionamiento interno: `TextScript`.

Archivos de inicio: `'scm/new-markup.scm'`.

## Advertencias y problemas conocidos

Los errores de sintaxis para el modo de marcado a menudo producen confusión.

### 1.8.2.2 Instrucciones de marcado de texto más usuales

### 1.8.2.3 Alineación de texto

### 1.8.2.4 Notación gráfica dentro de elementos de marcado

### 1.8.2.5 Notación musical dentro de elementos de marcado

### 1.8.2.6 Elementos de marcado de varias páginas

Así como `\markup` se utiliza para introducir un bloque de texto no divisible en líneas, se puede usar `\markuplines` en el nivel más alto del código para introducir líneas de texto que pueden abarcar varias páginas:

```
\markuplines {
  \justified-lines {
    Un texto muy largo de líneas justificadas.
    ...
  }
  \justified-lines {
    Otro párrafo muy largo.
    ...
  }
  ...
}
```

`\markuplines` acepta una lista de marcado, esto es: o bien el resultado de una instrucción de lista de marcado, o bien una lista de elementos de marcado o de listas de marcado. Las instrucciones de lista de marcado predefinidas se explican en [Sección B.7 \[Text markup list commands\]](#), [página 294](#).

## Véase también

En este manual: [Sección B.7 \[Text markup list commands\]](#), [página 294](#), [Sección 6.4.4 \[Definición de nuevas instrucciones de lista de marcado\]](#), [página 238](#).

Fragmentos de código: [Sección “Text” in Fragmentos de código](#).

## Instrucciones predefinidas

`\markuplines`

## 1.8.3 Tipografías

### 1.8.3.1 Fuentes tipográficas del documento completo

También es posible cambiar la familia predeterminada de la fuente tipográfica para todo el documento. Se hace llamando a la función `make-pango-font-tree` desde dentro del bloque `\paper`. Esta función toma los nombres de las familias de fuente que se quieren usar para el texto en tipo romano, sans-serif y monoespaciado. Por ejemplo:

```
\paper {
  myStaffSize = #20

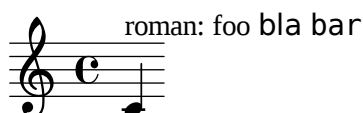
  #(define fonts
    (make-pango-font-tree "Times New Roman"
                          "Nimbus Sans"
                          "Luxi Mono")
  )
```

```

        (/ myStaffSize 20)))
}

{
  c'^\markup { roman: foo \sans bla \typewriter bar }
}

```



### 1.8.3.2 Fuentes de un solo elemento

Mediante el ajuste de las propiedades de objeto descritas a continuación, podemos elegir una fuente de entre las familias preconfiguradas. LilyPond contempla de forma predeterminada las fuentes de música «Feta». Las fuentes del texto se eligen a través de Pango/FontConfig. La fuente de tipo serif predeterminada es New Century Schoolbook, y las fuentes de tipo sans y máquina de escribir a lo que tenga por predeterminado la instalación de Pango.

- **font-encoding** es un símbolo que establece la presentación de los glifos. Sólo se debería ajustar para seleccionar distintos tipos de fuente que no sean para texto, p. ej.:  
**fetaBraces** para las llaves de los pentagramas de piano, **fetaMusic** es la fuente estándar para música, incluidos los glifos de la música antigua, **fetaDynamic** para los matices dinámicos y **fetaNumber** para los números.
- **font-family** es un símbolo que indica la clase general del tipo. Están contempladas **roman** (Computer Modern), **sans** y **typewriter**.
- **font-shape** es un símbolo que indica la forma de la fuente. Están disponibles normalmente varias formas de tipografía para cada una de las familias de fuente. se puede elegir entre **italic** (cursiva), **caps** (versalitas) y **upright** (recta).
- **font-series** es un símbolo que indica la serie de la tipografía. En general existen varias series de fuente para cada familia y forma de una tipografía. Se puede elegir entre **medium** (normal) y **bold** (negrita).

Las fuentes que se eligen de esta manera provienen de una hoja de estilos predefinida. Si quiere usar una tipografía de fuera de la hoja de estilos establezca el valor de la propiedad **font-name**:

```

{
  \override Staff.TimeSignature #'font-name = #"Charter"
  \override Staff.TimeSignature #'font-size = #2
  \time 3/4
  c'1_\markup {
    \override #'(font-name . "Vera Bold")
    { This text is in Vera Bold }
  }
}

```



Se puede usar cualquier fuente, en la medida en que esté disponible para Pango/FontConfig. Para obtener una lista completa de todas las fuentes disponibles, ejecute la instrucción

`lilypond -dshow-available-fonts blabla`

(el último argumento puede ser `blabla` o cualquier otra cosa, pero debe estar presente).

El tamaño de la fuente se puede ajustar con la propiedad `font-size`. El tamaño que resulta se considera en relación al tamaño `text-font-size` tal y como está definido en el bloque `\paper`.

## Véase también

Fragmentos de código: *Sección “Text” in Fragmentos de código.*

## 2 Notación especializada

Este capítulo explica cómo crear notación musical para instrumentos de un tipo específico o dentro de estilos específicos.

### 2.1 Música vocal

Puesto que los archivos de entrada de LilyPond son de texto, hay un par de cuestiones dignas de consideración cuando se trabaja con música vocal:

- Los textos de las canciones se deben introducir como texto, no como notas. Por ejemplo, la entrada `d` se debe interpretar como una sílaba de una sola letra, no como la nota *Re*.
- Los textos de las canciones se deben alinear con las notas respectivas de la melodía.

Existen varias formas diferentes de definir la letra; comenzaremos examinando el método más simple, y poco a poco iremos aumentando la complejidad.

#### Fragmentos de código seleccionados

Las comprobaciones que aseguran que las indicaciones textuales y las letras se encuentran dentro de los márgenes es una tarea computacional relativamente grande. Para acelerar el procesado, lilypond no realiza estos cálculos por defecto; para activarlos, utilice

```
\override Score.PaperColumn #'keep-inside-line = ##t
```

Para hacer que la letra evite las líneas divisorias también, utilice

```
\layout {
  \context {
    \Lyrics
    \consists "Bar_engraver"
    \consists "Separating_line_group_engraver"
    \override BarLine #'transparent = ##t
  }
}
```

#### 2.1.1 Notación usual para voces

##### 2.1.1.1 Referencias para música vocal

‘Parlato’ es texto hablado sin notas pero con ritmo; su notación se realiza mediante cabezas de nota en forma de cruz. Esto queda demostrado en [\[Cabezas de nota especiales\]](#), página 26.

##### 2.1.1.2 Elaborar canciones sencillas

La manera más fácil de poner letra a una melodía es añadir

```
\addlyrics { la letra }
```

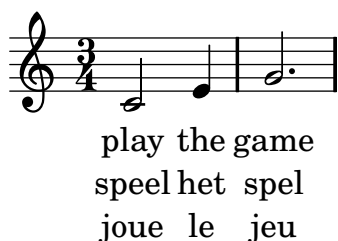
a la melodía. He aquí un ejemplo:

```
\time 3/4
\relative c' { c2 e4 g2. }
\addlyrics { play the game }
```



Se pueden añadir más versos mediante la adición de más secciones `\addlyrics`

```
\time 3/4
\relative c' { c2 e4 g2. }
\addlyrics { play the game }
\addlyrics { speel het spel }
\addlyrics { joue le jeu }
```



La instrucción `\addlyrics` no puede manejar situaciones de polifonía. Para estos casos tendrá que usar `\lyricsto` y `\lyricmode`, tal y como se explicará en [Sección 2.1.1.3 \[Introducir la letra\]](#), página 131.

### 2.1.1.3 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 puede introducir la letra, con puntuación y acentos, y el carácter `d` no se analiza como una nota, sino más bien como una sílaba de una sola letra. Las sílabas se introducen como las notas, pero con texto en lugar de alturas de nota. Por ejemplo,

```
\lyricmode { Cam-4 pa-4 ni- ta del- lu- gar2 }
```

Existen dos métodos principales para especificar la colocación horizontal de las sílabas, bien especificando la duración de cada sílaba de forma explícita, como en el ejemplo anterior, o bien alineando automáticamente la letra a la melodía o a cualquier otra voz, usando `\addlyrics` o `\lyricsto`.

Una palabra o sílaba de la letra comienza con un carácter alfabético, y termina con cualquier espacio o dígito. Los caracteres que siguen pueden ser cualesquiera excepto un dígito o un espacio.

Cualquier carácter excepto un dígito o un espacio en blanco se considera parte de una sílaba; esto tiene una importante consecuencia, y es que una palabra puede terminar en `}`, lo que con frecuencia conduce al siguiente error:

```
\lyricmode { 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.

De forma similar, un punto que sigue a una secuencia alfabética queda incluido dentro de la cadena resultante. Como consecuencia, se deben insertar espacios antes y después de los comandos de propiedades: *no escriba*

```
\override Score.LyricText #'font-shape = #'italic
```

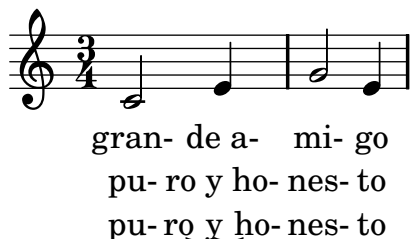
sino lo siguiente:

```
\override Score . LyricText #'font-shape = #'italic
```

Para asignar más de una sílaba a una única nota, puede rodearlas por comillas o usar un carácter `_` (guión bajo), para obtener espacios entre las sílabas, o usar el símbolo de tilde curva (`~`) para obtener una ligadura entre sílabas de la letra.



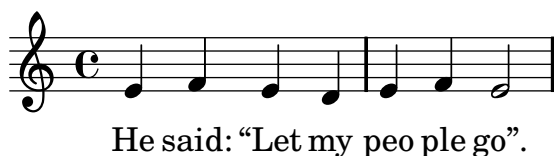
```
\time 3/4
\relative c' { c2 e4 g2 e4 }
\addlyrics { gran- de_a- mi- go }
\addlyrics { pu- "ro y ho-" nes- to }
\addlyrics { pu- ro~y~ho- nes- to }
```



La ligadura de texto se construye con el carácter Unicode U+203F, por tanto debe asegurarse de que tiene instalada una tipografía (como DejaVuLGC) que incluya este glifo.

Para escribir letra con caracteres de una lengua no inglesa, o que tenga caracteres acentuados o especiales (como el símbolo del corazón o comillas inclinadas), introduzca simplemente los caracteres directamente en el archivo de entrada y guárdelo con una codificación utf-8. Consulte [Sección 3.3.3 \[Codificación del texto\], página 206](#), para ver más información.

```
\relative c' { e4 f e d e f e2 }
\addlyrics { He said: \Let my peo ple go". }
```



Para utilizar comillas normales en la letra, escriba una barra invertida antes de las comillas. Por ejemplo,

```
\relative c' { \time 3/4 e4 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 ^.

Para definir identificadores que contengan letra, se debe usar la función `lyricmode`.

```
verseOne = \lyricmode { Joy to the world the Lord is come }
\score {
  <<
    \new Voice = "one" \relative c' {
      \autoBeamOff
```

```

        \time 2/4
        c4 b8. a16 g4. f8 e4 d c2
    }
    \addlyrics { \verseOne }
>>
}

```

## Véase también

Referencia de funcionamiento interno: `LyricText`, `LyricSpace`.

### 2.1.1.4 Trabajar con el texto y las variables

Para definir identificadores que contengan texto, se debe usar la función `\lyricmode`. Si embargo no tendrá que escribir las duraciones, si añade `\addlyrics` o `\lyricsto` en el momento de invocar el identificador.

```

verseOne = \lyricmode { Joy to the world the Lord is come }
\score {
  <<
    \new Voice = "one" \relative c' {
      \autoBeamOff
      \time 2/4
      c4 b8. a16 g4. f8 e4 d c2
    }
    \addlyrics { \verseOne }
  >>
}

```

Para arreglos distintos o más complejos, la mejor forma es preparar previamente la jerarquía de pautas y textos, p.ej.:

```

\new ChoirStaff <<
  \new Voice = "soprano" { música }
  \new Lyrics = "textoSoprano" { s1 }
  \new Lyrics = "textoTenor" { s1 }
  \new Voice = "tenor" { música }
>>

```

y después combinar las melodías y líneas de texto correspondientes:

```

\context Lyrics = textoSoprano \lyricsto "soprano"
el texto

```

El código de entrada definitivo sería algo como

```

<<\new ChoirStaff << prepare la música >>
  \lyricsto "soprano" etc
  \lyricsto "alto" etc
etc
>>

```

## Véase también

Referencia de funcionamiento interno: `LyricCombineMusic`, `Lyrics`.

### 2.1.2 Alineación de la letra a una melodía

La letra se imprime a través de su interpretación dentro del contexto llamado `Lyrics`.

`\new Lyrics \lyricmode ...`

Existen dos métodos principales para especificar la colocación horizontal de las sílabas:

- alineando automáticamente el texto a una melodía o a otra voz, utilizando `\addlyrics` o `\lyricsto`.
- o especificando la duración de cada sílaba explícitamente, utilizando `\lyricmode`

#### 2.1.2.1 Duración automática de las sílabas

La letra se puede alinear automáticamente bajo una melodía dada. Esto se consigue combinando la melodía y el texto con la expresión `\lyricsto`

`\new Lyrics \lyricsto nombre ...`

Esto alinea la letra a las notas del contexto de `Voice` llamado *nombre*, que debe existir previamente. Por ello la `Voice` normalmente se especifica en primer lugar, y después se especifica la letra con `\lyricsto`. La instrucción `\lyricsto` cambia automáticamente al modo `\lyricmode`, por lo que la palabra clave `\lyricmode` se puede omitir.

El ejemplo siguiente utiliza comandos diferentes para introducir la letra.

```
<<
\new Voice = "one" \relative c'' {
  \autoBeamOff
  \time 2/4
  c4 b8. a16 g4. f8 e4 d c2
}
\new Lyrics \lyricmode { Joy4 to8. the16 world!4. the8 Lord4 is come.2 }
\new Lyrics \lyricmode { Joy to the earth! the Sa -- viour reigns. }
\new Lyrics \lyricsto "one" { No more let sins and sor -- rows grow. }
>>
```



Joy to the world! the Lord is come.  
 Joy to the earth! the Sa - viour  
 No more let sins and sor - rows grow.

8

reigns.

El segundo verso no está bien alineado porque las duraciones no se han especificado. Una solución para eso sería usar `\lyricsto`.

El comando `\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 }`

### 2.1.2.2 Duración manual de las sílabas

El texto también se puede introducir sin `\addlyrics` ni `\lyricsto`. En este caso, las sílabas se escriben como notas (pero con las alturas sustituidas por texto) y la duración de cada sílaba se debe escribir explícitamente. Por ejemplo:

```
play2 the4 game2.
sink2 or4 swim2.
```

La alineación respecto de una melodía se puede especificar con la propiedad `associatedVoice`,

```
\set associatedVoice = #"lala"
```

El valor de la propiedad (aquí: "lala") ha de ser el nombre de un contexto de `Voice`. Sin este ajuste, las líneas de extensión no se formatearán correctamente.

Presentamos a continuación un ejemplo que muestra el uso de las duraciones manuales de las sílabas del texto:

```
<< \new Voice = "melody" {
    \time 3/4
    c2 e4 g2.
}
\new Lyrics \lyricmode {
    \set associatedVoice = #"melody"
    play2 the4 game2.
} >>
```



### Véase también

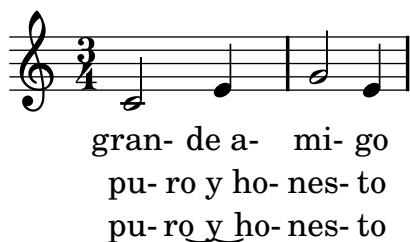
Referencia de funcionamiento interno: `Lyrics`.

### 2.1.2.3 Varias sílabas sobre una nota

Para asignar más de una sílaba a una sola nota, puede encerrarlas entre comillas o utilizar un guión bajo (`_`), para dejar espacios entre las sílabas, o usar el símbolo de tilde curva (`~`) para obtener una ligadura en el texto<sup>1</sup>.

```
\time 3/4
\relative c' { c2 e4 g2 e4 }
\addlyrics { gran- de_a- mi- go }
\addlyrics { pu- "ro y ho-" nes- to }
\addlyrics { pu- ro~y~ho- nes- to }
```

<sup>1</sup> Las ligaduras de texto están hechas con el carácter de Unicode U+203F, por tanto debe asegurarse de tener instalada una tipografía (como DejaVuLGC) que incluya este glifo.



## Véase también

Referencia de funcionamiento interno: `LyricCombineMusic`.

### 2.1.2.4 Varias notas sobre una sílaba

A veces, y sobre todo en la música medieval, varias notas se cantan sobre una sílaba única; tales vocalizaciones reciben el nombre de melismas, o melismata.

Puede definir los melismas completamente dentro de la letra, escribiendo un guión bajo `_` para cada nota que forma parte del melisma.

```
{ \set melismaBusyProperties = #'()
  c d( e) f f( e) e e }
\addlyrics
{ Ky -- _ _ ri _ _ _ _ e }
```



En este caso, también puede tener ligaduras de unión y de expresión en la melodía, si establece `melismaBusyProperties`, como se hizo en el ejemplo anterior.

Sin embargo la instrucción `\lyricsto` también puede detectar los melismas automáticamente: sólo pone una sílaba debajo de un grupo de notas unidas mediante ligadura de unión o de expresión. Si quiere forzar que un grupo de notas sin ligadura sea un melisma, inserte `\melisma` después de la primera nota del grupo, y `\melismaEnd` después de la última, p.ej.:

```
<<
  \new Voice = "lala" {
    \time 3/4
    f4 g8
    \melisma
    f e f
    \melismaEnd
    e2
  }
  \new Lyrics \lyricsto "lala" {
    la di _ _ daah
  }
>>
```



Además, las notas se consideran un melisma si están unidas manualmente mediante una barra, y el barrado automático (véase [Sección 1.2.4.2 \[Establecer el comportamiento de las barras automáticas\]](#), página 49) está desactivado.

Hay un ejemplo completo de una disposición de partitura SATB en la sección [Sección “Vocal ensembles” in \*Manual de Aprendizaje\*](#).

## Instrucciones predefinidas

`\melisma, \melismaEnd`

## Véase también

## Advertencias y problemas conocidos

Los melismas no se detectan automáticamente, y las líneas de extensión se deben insertar a mano.

### 2.1.2.5 Saltar notas

### 2.1.2.6 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).

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 la `minimum-distance` (distancia mínima entre las dos sílabas) y la `minimum-length` (umbral por debajo del cual se suprimen los guiones).

## Véase también

Referencia de funcionamiento interno: `LyricExtender`, `LyricHyphen`

### 2.1.2.7 Letras y repeticiones

### 2.1.3 Colocación de la letra

A menudo se aplican a una melodía distintos versos de una canción de formas ligeramente distintas. Tales variantes pueden aún captarse mediante `\lyricsto`.

#### 2.1.3.1 Letras en divisi

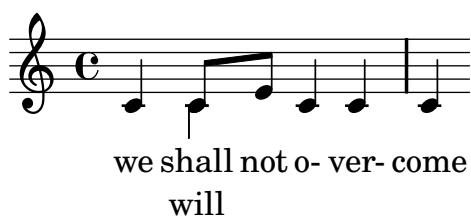
Puede presentar letras alternativas (o divisi) nombrando contextos de voz y aplicando la letra a los mencionados contextos específicos.

```
\score{ <<
  \new Voice = "melody" {
```

```

\relative c' {
  c4
  <<
    { \voiceOne c8 e }
    \new Voice = "splitpart" { \voiceTwo c4 }
  >>
  \oneVoice c4 c | c
}
}
\new Lyrics \lyricsto "melody" { we shall not o- ver- come }
\new Lyrics \lyricsto "splitpart" { will }
>> }

```

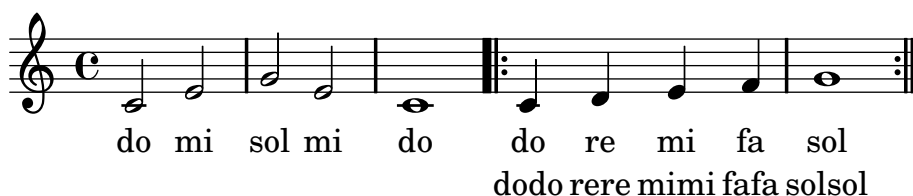


Puede usar este truco para presentar las distintas letras de una sección que se repite.

```

\score{ <<
  \new Voice = "melody" \relative c' {
    c2 e | g e | c1 |
    \new Voice = "verse" \repeat volta 2 {c4 d e f | g1 | }
    a2 b | c1}
  \new Lyrics = "mainlyrics" \lyricsto melody \lyricmode {
    do mi sol mi do
    la si do }
  \context Lyrics = "mainlyrics" \lyricsto verse \lyricmode {
    do re mi fa sol }
  \new Lyrics = "repeatlrics" \lyricsto verse \lyricmode {
    dodo rere mimi fafa solsol }
>>
}

```



### 2.1.3.2 Letra independiente de las notas

En cierta música vocal compleja, podría ser deseable colocar la letra de forma totalmente independiente de las notas. La música que se define dentro de `lyricrhythm` desaparece dentro del contexto `Devnull`, pero los valores rítmicos aún se pueden utilizar para colocar la letra.

```
voz = {
  c''2
  \tag #'music { c''2 }
  \tag #'lyricrhythm { c''4. c''8 }
  d''1
}

letra = \lyricmode { I like my cat! }

<<
  \new Staff \keepWithTag #'music \voz
  \new Devnull="nowhere" \keepWithTag #'lyricrhythm \voz
  \new Lyrics \lyricsto "nowhere" \letra
  \new Staff { c'8 c' c' c' c' c' c' c'
    c' c' c' c' c' c' c' c' }
>>
```



### 2.1.3.3 Cantos

#### 2.1.3.4 Espaciado de las sílabas

Para aumentar el espaciado entre las sílabas de la letra, establezca la propiedad `minimum-distance` de `LyricSpace`.

```
{
  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 que este cambio sea válido para la letra de la partitura completa, establezca la propiedad dentro del ‘layout’ o disposición.

```
\score {
  \relative c' {
    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
    }
  }
}
```



### 2.1.3.5 Centrado de la letra entre pentagramas

## 2.1.4 Versos

### 2.1.4.1 Adding stanza numbers

Los números de los versos se pueden añadir estableciendo `stanza`, p.ej.,

```
\new Voice {
  \time 3/4 g2 e4 a2 f4 g2.
} \addlyrics {
  \set stanza = "1. "
  Hi, my name is Bert.
} \addlyrics {
  \set stanza = "2. "
  Oh, ché -- ri, je t'aime
}
```



1. Hi, my name is Bert.
2. Oh, ché - ri, je t'aime

Estos números aparecerán inmediatamente antes de la primera sílaba.

#### 2.1.4.2 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`:

```
texto = {
  \set stanza = \markup { \dynamic "ff" "1. " }
  \lyricmode {
    Big bang
  }
}
```

```
<<
  \new Voice = "tune" {
    \time 3/4
    g'4 c'2
  }
\new Lyrics \lyricsto "tune" \texto
>>
```



***ff*** 1. Big bang

#### 2.1.4.3 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 {
  \time 3/4 g2 e4 a2 f4 g2.
} \addlyrics {
  \set vocalName = "Bert "
  Hi, my name is Bert.
} \addlyrics {
  \set vocalName = "Ernie "
  Oh, ché -- ri, je t'aime
}
```



Bert	Hi, my name is Bert.
Ernie	Oh, ché - ri, je t'aime

### 2.1.4.4 Versos con ritmos distintos

#### 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.

Hay un aspecto delicado: el valor de `ignoreMelismata` se debe establecer una sílaba *antes* de la sílaba no melismática del texto, tal y como se muestra a continuación:

```
%{
<<
  \relative c' \new Voice = "lahlah" {
    \set Staff.autoBeaming = ##f
    c4
    \slurDotted
    f8.[( g16)]
    a4
  }
  \new Lyrics \lyricsto "lahlah" {
    more slow -- ly
  }
  \new Lyrics \lyricsto "lahlah" {
    \set ignoreMelismata = ##t % se aplica al "fas"
    go fas -- ter
    \unset ignoreMelismata
    still
  }
>>
%}
```

El `ignoreMelismata` se aplica a la sílaba ‘fas’, por ello se debe introducir antes de ‘go’.

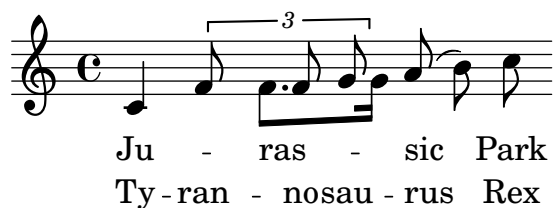
La inversa también es posible: hacer que una línea de letra sea más lenta que la normal. Esto se puede conseguir insertando `\skips` (desplazamientos) dentro de la letra. Para cada `\skip`, el texto se retrasará una nota más. Por ejemplo,

```
\relative c' { c c g' }
\addlyrics {
  twin -- \skip 4
  kle
}
```



#### Cambio a una melodía alternativa

Son posibles variaciones más complejas en la disposición del texto. Se puede cambiar la melodía para una línea de la letra durante el texto. Esto se hace estableciendo la propiedad `associatedVoice`. En el ejemplo



el texto del primer verso se aplica a una melodía llamada ‘lahlah’,

```
\new Lyrics \lyricsto "lahlah" {
  Ju -- ras -- sic Park
}
```

El segundo verso se aplica en principio al contexto de `lahlah`, pero para la sílaba ‘ran’, cambia a una melodía diferente. Esto se consigue con

```
\set associatedVoice = alternative
```

Aquí, `alternative` es el nombre del contexto de `Voice` que contiene el tresillo.

Una vez más, la instrucción debe ir una sílaba antes de la cuenta, antes de ‘Ty’ en este caso.

```
\new Lyrics \lyricsto "lahlah" {
  \set associatedVoice = alternative % se aplica al "ran"
  Ty --
  ran --
  no --
  \set associatedVoice = lahlah % se aplica al "rus"
  sau -- rus Rex
}
```

La disposición se vuelve a cambiar a la situación inicial mediante la asignación de `lahlah` a `associatedVoice`.

#### 2.1.4.5 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`.

```
melodia = \relative c' {
  e d c d | e e e e |
  d d e d | c1 |
}
```

```
texto = \lyricmode {
\set stanza = "1." Ma- ry had a lit- tle lamb,
its fleece was white as snow.
}
```

```
\score{ <<
  \new Voice = "one" { \melodia }
  \new Lyrics \lyricsto "one" \texto
>>
  \layout { }
}
\markup { \column{
  \line{ Verse 2. }
  \line{ All the children laughed and played }
  \line{ To see a lamb at school. }
}
```

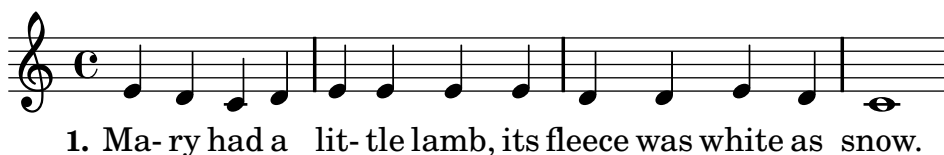
```

}
\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.

#### 2.1.4.6 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.

```

melodia = \relative c' {
  c c c c | d d d d
}

texto = \lyricmode {
  \set stanza = "1." This is verse one.
  It has two lines.
}

\score{ <<
  \new Voice = "one" { \melodia }
  \new Lyrics \lyricsto "one" \texto
  >>
  \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."
    }
}
\hspace #0.1 % añade espaciado vertical entre estrofas
\line { \bold "3."
    \column {
        "This is verse three."
        "It has two lines."
    }
}
}
\hspace #0.1 % adds horizontal spacing between columns; if they are
% still too close, add more " " pairs until the result
% looks good
\column {
    \line { \bold "4."
        \column {
            "This is verse four."
            "It has two lines."
        }
    }
}
\hspace #0.1 % añade espaciado vertical entre estrofas
\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: `LyricText`, `StanzaNumber`.

## 2.2 Instrumentos de teclado

### 2.2.1 Notación usual para instrumentos de teclado

#### Referencias para teclados

Esta sección aún no está traducida; consulte el manual en inglés.

Los sistemas de piano son dos pentagramas normales agrupados mediante una llave. Los pentagramas son hasta cierto punto independientes, pero a veces las voces pueden cruzarse entre los dos pentagramas. También se utiliza esta notación para otros instrumentos de teclado y para el arpa. El `PianoStaff` está preparado especialmente para manejar este comportamiento de cruce de pentagramas. En esta sección trataremos el `PianoStaff` y alguna que otra peculiaridad pianística.

#### Advertencias y problemas conocidos

Los matices no se colocan centrados, pero se pueden utilizar ciertas alternativas. Consulte la plantilla ‘piano con matices centrados’ que está en [Sección “Piano templates” in \*Manual de Aprendizaje\*](#).

#### Cambiar de pentagrama manualmente

Las voces se pueden cambiar de pentagrama manualmente usando la instrucción

```
\change Staff = nombre_del_pentagrama música
```

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`. Por lo común `nombre_del_pentagrama` es "up" o "down". El `Staff` referido debe existir previamente, por ello normalmente la preparación de una partitura empezará por la preparación de los pentagramas,

```
<<
\new Staff = "up" {
  \skip 1 * 10 % mantener vivo el pentagrama
}
\new Staff = "down" {
  \skip 1 * 10 % ídem
}
>>
```

y la `Voice` se inserta con posterioridad

```
\context Staff = down
\new Voice { ... \change Staff = up ... }
```

#### 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 `PianoStaff` actual, llamados `up` (arriba) y `down` (abajo). El pentagrama inferior estará en clave de Fa por omisión.

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`, como en

```
\autochange \relative ... ...
```

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. He aquí un ejemplo práctico

```
\new PianoStaff
\autochange \relative c'
{
  g4 a b c d r4 a g
}
```



## Véase también

En este manual: [Sección 1.7.1 \[Dentro del pentagrama\]](#), página 108.

Referencia del programa: `AutoChangeMusic`.

## Advertencias y problemas conocidos

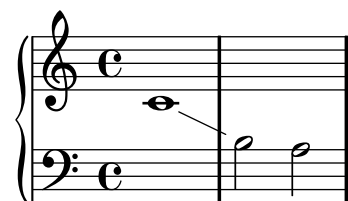
Los cambios de pentagrama pueden no acabar en lugares óptimos. Para conseguir una salida de alta calidad es mejor especificar manualmente los cambios de pentagrama.

`\autochange` no puede estar dentro de `\times`.

## Líneas de cambio de pentagrama

Cuando una voz cambia a otro pentagrama, se puede imprimir automáticamente una línea que conecte las notas. Esto se activa estableciendo `followVoice` a verdadero

```
\new PianoStaff <<
  \new Staff="one" {
    \set followVoice = ##t
    c1
    \change Staff=two
    b2 a
  }
  \new Staff="two" { \clef bass \skip 1*2 }
>>
```





## Véase también

Referencia de funcionamiento interno: `VoiceFollower`.

## Instrucciones predefinidas

`\showStaffSwitch`, `\hideStaffSwitch`.

## Plicas de pentagrama cruzado

Los acordes que cruzan pentagramas se pueden producir mediante el incremento de la longitud de la plica en el pentagrama inferior, de manera que alcance a la plica que está en el pentagrama superior, y viceversa.

```
stemExtend = {
  \once \override Stem #'length = #10
  \once \override Stem #'cross-staff = ##t
}
noFlag = \once \override Stem #'flag-style = #'no-flag
\new PianoStaff <<
  \new Staff {
    \stemDown \stemExtend
    f'4
    \stemExtend \noFlag
    f'8
  }
  \new Staff {
    \clef bass
    a4 a8
  }
>>
```



### 2.2.2 Piano

#### Pedales de piano

El piano tiene pedales que alteran la forma en que se produce el sonido. Generalmente un piano tiene tres pedales: derecho, una corda y tonal (en inglés: sustain, una corda y sostenuto).

Se puede expresar una instrucción de pedal de piano adjuntando `\sustainOn`, `\sustainOff`, `\unaCorda`, `\treCorde`, `\sostenutoOn` y `\sostenutoOff` a una nota o acorde

```
c'4\sustainOn c'4\sustainOff
```



Es posible modificar lo que se imprime estableciendo `pedalXStrings`, donde *X* es uno de los tipos de pedal: `Sustain`, `Sostenuto` o `UnaCorda`. Consulte `SustainPedal` dentro de la referencia del programa para más información.

Los pedales se pueden indicar también mediante una secuencia de corchetes, estableciendo la propiedad `pedalSustainStyle` a objetos corchete

```
\set Staff.pedalSustainStyle = #'bracket
c\sustainOn d e
b\sustainOff\sustainOn
b g \sustainOff a \sustainOn \bar "|."
```



Un tercer estilo de notación de pedal es una mezcla de texto y corchetes, que se obtiene estableciendo la propiedad `pedalSustainStyle` a `mixed`

```
\set Staff.pedalSustainStyle = #'mixed
c\sustainOn d e
b\sustainOff\sustainOn
b g \sustainOff a \sustainOn \bar "|."
```



El estilo por defecto ‘\*Ped.’ para los pedales derecho y una corda corresponde al estilo `text`. El pedal tonal usa el estilo `mixed` por defecto.

```
c\sostenutoOn d e c, f g a\sostenutoOff
```



Para hacer un ajuste fino del aspecto de un corchete de pedal, se pueden modificar las propiedades `edge-width`, `edge-height` y `shorten-pair` de los objetos `PianoPedalBracket` (véase `PianoPedalBracket` en la Referencia del programa). Por ejemplo, el corchete se puede extender hasta el borde derecho de la cabeza de la nota

```
\override Staff.PianoPedalBracket #'shorten-pair = #'(0 . -1.0)
c\sostenutoOn d e c, f g a\sostenutoOff
```



## Véase también

Referencia de la notación: [Sección 1.2.1.4 \[Ligaduras de unión\]](#), página 35 (*laissez vibrer*).

### 2.2.3 Acordeón

#### Símbolos de Discanto

Esta sección aún no está traducida; consulte el manual en inglés.

## 2.3 Instrumentos de cuerda sin trastes

### 2.3.1 Notación usual para cuerdas sin trastes

#### 2.3.1.1 Referencias para cuerdas sin trastes

#### 2.3.2 Instrumentos de arco

##### 2.3.2.1 Referencias para instrumentos de cuerda frotada

Los armónicos artificiales se tipografían con un estilo de cabeza distinto. Se introducen marcando la nota que es un armónico mediante `\harmonic`.

```
<c g'\harmonic>4
```



### 2.3.3 Instrumentos de cuerda pulsada

#### 2.3.3.1 Arpa

## 2.4 Instrumentos de cuerda con trastes

### 2.4.1 Notación usual para cuerdas con trastes

#### Referencias para cuerdas con trastes

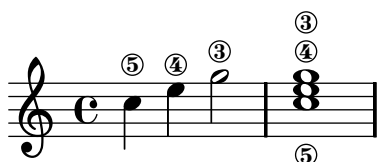
#### Indicación de los números de cuerda

Se puede indicar la cuerda en que se debe tocar una nota añadiendo `\ número` a una nota dentro de una construcción de acorde (`<>`).

**Nota:** Los números de cuerda **se deben** definir dentro de una construcción de acorde con ángulos simples, incluso si se trata de una sola nota.

```
<c\5>4 <e\4> <g\3>2
```

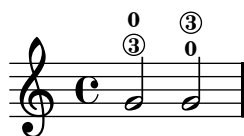
```
<c,\5 e\4 g\3>1
```



Aunque las digitaciones se explican en detalle en otro lugar, merece la pena citar aquí cómo se comportan juntas las indicaciones de digitación y de cuerda, porque la música de guitarra con frecuencia incorpora los dos tipos al mismo tiempo. La convención usual es situar la digitación más cerca de la nota, y después la indicación de la cuerda por fuera de la digitación. Esto se controla por medio del orden en que aparecen los dos elementos:

```
<g\3-0>2
```

```
<g-0\3>
```



## Fragmentos de código seleccionados

### Véase también

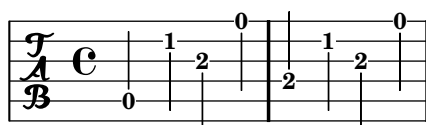
Referencia de funcionamiento interno: `StringNumber`,

## Tablaturas predeterminadas

La notación de tablatura se utiliza para la notación de música de instrumentos de cuerda pulsada. Las alturas no se denotan mediante cabezas de nota, sino mediante números que indican sobre qué cuerda y traste se debe tocar la nota. LilyPond contempla las tablaturas de forma limitada.

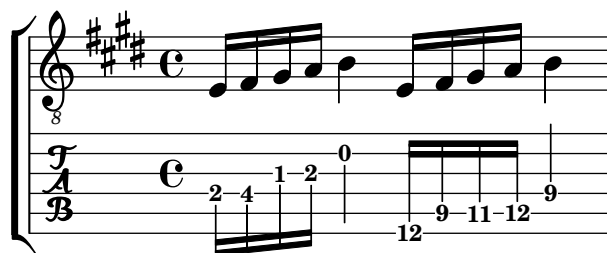
El número de cuerda asociado a una nota se proporciona en la forma de una barra invertida seguida por un número, p.ej. `c4\3` es un Do negra sobre la tercera cuerda. Por defecto, la cuerda 1 es la más aguda, y la afinación normal es la afinación estándar de guitarra (con 6 cuerdas). Las notas se imprimen como tablatura, usando los contextos `TabStaff` y `TabVoice`:

```
\new TabStaff {
  a,4\5 c'\2 a\3 e'\1
  e\4 c'\2 a\3 e'\1
}
```



Cuando no se especifica ninguna cuerda, se selecciona la primera cuerda que no da un número de traste menor de `minimumFret` (traste mínimo). El valor por defecto de `minimumFret` es cero:

```
e16 fis gis a b4
\set TabStaff.minimumFret = #8
e16 fis gis a b4
```



## Propiedades trucadas frecuentemente

Para imprimir tablaturas con las plicas hacia abajo y barras horizontales, inicialice el `TabStaff` (pentagrama de tablatura) con el código siguiente:

```
\stemDown
\override Beam #'damping = #100000
```

## Véase también

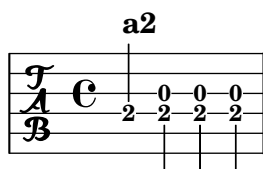
Referencia de funcionamiento interno: `TabNoteHead`, `TabStaff`, `TabVoice`.

## Advertencias y problemas conocidos

Los acordes no se tratan de una manera especial, de aquí que el selector automático de cuerdas podría fácilmente seleccionar la misma cuerda para dos notas dentro del mismo acorde.

Para manejar `\partcombine`, una pauta de tablatura `TabStaff` debe usar voces creadas especialmente:

```
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 }
  >>
>>
```

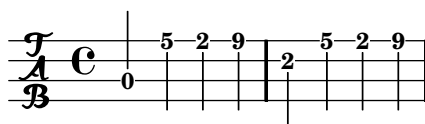


## Tablaturas personalizadas

Puede cambiar la afinación de las cuerdas. Una afinación de cuerdas se da como una lista en lenguaje Scheme con un número entero para cada cuerda, donde el número es la altura (medida en semitonos desde el Do central) de una cuerda al aire. Los números especificados para `stringTunings` son la cantidad de semitonos que se deben restar o añadir, comenzando por la altura especificada que por defecto es el Do central, y en el orden de las cuerdas. LilyPond calcula automáticamente el número de cuerdas leyendo `stringTunings`.

En el ejemplo siguiente, `stringTunings` se establece a los valores Mi, La, Re y Sol

```
\new TabStaff <<
  \set TabStaff.stringTunings = #'(-5 -10 -15 -20)
  {
    a,4 c' a e' e c' a e'
  }
>>
```



LilyPond tiene afinaciones predefinidas para el banjo, la mandolina, la guitarra y el bajo eléctrico.

```
\set TabStaff.stringTunings = #bass-tuning
```

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).

## Véase también

El archivo ‘`scm/output-lib.scm`’ contiene las afinaciones predefinidas de las cuerdas.

Referencia de funcionamiento interno: `Tab_note_heads_engraver`.

## Advertencias y problemas conocidos

No se ha implementado ningún efecto especial de guitarra.

## Marcas de diagramas de trastes

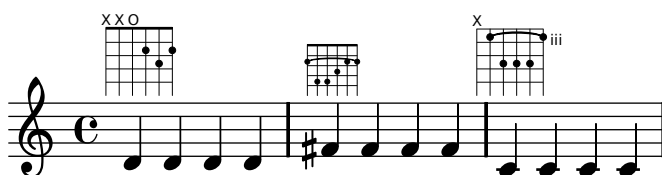
Se pueden añadir diagramas de trastes a la música como elementos de marcado sobre la nota deseada. El marcado contiene información sobre el diagrama de trastes deseado. Existen tres interfaces distintos de marcado de diagramas de trastes: `standard` (estándar), `terse` (escueto) y `verbose` (prolijo). Los tres interfaces producen marcados equivalentes, pero tienen cantidades variables de información en la cadena de marcado. Hay más detalles sobre los interfaces de marcado en [Sección B.6 \[Text markup commands\]](#), página 258.

```
\new Voice {
  d'^\markup \fret-diagram #"6-x;5-x;4-o;3-2;2-3;1-2;"
  d' d' d'
  fis'^\markup \override #'(size . 0.75) {
    \override #'(finger-code . below-string) {
      \fret-diagram-verbose #'((place-fret 6 2 1) (barre 6 1 2)
                             (place-fret 5 4 3)
                             (place-fret 4 4 4)
                             (place-fret 3 3 2)
                             (place-fret 2 2 1)
                             (place-fret 1 2 1))
    }
  }
  fis' fis' fis'
```

```

c'~\markup \override #'(dot-radius . 0.35) {
  \override #'(finger-code . in-dot) {
    \override #'(dot-color . white) {
      \fret-diagram-terse #"x;3-1-(;5-2;5-3;5-4;3-1-);"
    }
  }
}
c' c' c'

```



Puede establecer un cierto número de propiedades gráficas según su elección. Se pueden encontrar más detalles sobre el interface de las propiedades de los diagramas de trastes en `fret-diagram-interface`.

## Véase también

## Digitaciones de la mano derecha

Se pueden introducir las digitaciones de la mano derecha de los acordes, usando `<nota-\rightHandFinger dedo >`. La indicación de digitación de la mano derecha se debe producir dentro de una construcción `<>` de acorde para que se pueda imprimir en la partitura, incluso si es una sola nota. Debe haber un guión después de la nota y un espacio antes del ángulo de cierre `>`.

```

\relative c' {
  <c-\rightHandFinger #1 >4
  <e-\rightHandFinger #2 >
  <g-\rightHandFinger #3 >
  <c-\rightHandFinger #4 >
  <c,-\rightHandFinger #1 e-\rightHandFinger #2 g-\rightHandFinger #3 c-\rightHandFinger #4
}

```



Por brevedad, es posible abreviar `\rightHandFinger` como algo más corto, como por ejemplo RH:

```

\define RH rightHandFinger

```

## Fragmentos de código seleccionados

Puede ejercer un mayor control sobre las digitaciones de la mano derecha estableciendo `strokeFingerOrientations`:

```
#(define RH rightHandFinger)
{
  \set strokeFingerOrientations = #'(up down)
  <c-\RH #1 es-\RH #2 g-\RH #4 > 4
  \set strokeFingerOrientations = #'(up right down)
  <c-\RH #1 es-\RH #2 g-\RH #4 > 4
}
```



Las letras utilizadas para las digitaciones se encuentran contenidas dentro de la propiedad `digit-names` (nombres de los dedos), pero también se pueden fijar individualmente aportando a `\rightHandFinger` un argumento de cadena, como en el ejemplo siguiente:

```
#(define RH rightHandFinger)
{
  \set strokeFingerOrientations = #'(right)
  \override StrokeFinger #'digit-names = #'#"x" "y" "z" "!" "@"
  <c-\RH #5 >4
  <c-\RH "@">4
  <c-\RH #2 >4
  <c-\RH "z">4
}
```



This example combines left-hand fingering, string indication, and right-hand fingering

```
#(define RH rightHandFinger)
\relative c' {
  <c-3\5-\RH #1 >4
  <e-2\4-\RH #2 >
  <g-0\3-\RH #3 >
  <c-1\2-\RH #4 >
}
```



## Véase también

Referencia de funcionamiento interno: `StrokeFinger`

### 2.4.2 Guitarra



## Indicar la posición y la cejilla

Este ejemplo muestra cómo incluir indicaciones de posición y cejilla de guitarra:

```
\clef "G_8"
b16 d16 g16 b16 e16
\textSpannerDown
\override TextSpanner #'bound-details #'left #'text = #"XII "
  g16\startTextSpan
  b16 e16 g16 e16 b16 g16\stopTextSpan
e16 b16 g16 d16
```



Las cabezas en aspa (X) se utilizan en la música de guitarra para indicar el lugar donde el guitarrista debe tocar una determinada nota o acorde, con sus dedos apenas tocando las cuerdas en lugar de pisarlas por completo. Esto produce en el sonido un carácter percusivo y ruidoso, sin perder la altura de la nota original. Su notación se realiza con cabezas en equis; se muestra en [\[Cabezas de nota especiales\]](#), página 26.

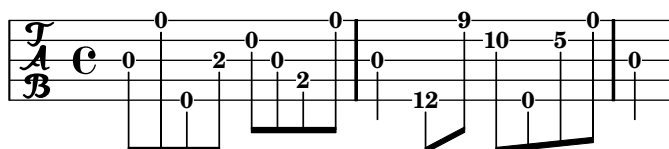
## Indicar armónicos y notas tapadas

### 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:

```
\new TabStaff <<
  \set TabStaff.tablatureFormat = #fret-number-tablature-format-banjo
  \set TabStaff.stringTunings = #banjo-open-g-tuning
  {
    \stemDown
    g8 d' g'\5 a b g e d' |
    g4 d'\8\5 b' a'\2 g'\5 e'\2 d' |
    g4
  }
>>
```



En LilyPond se encuentran predefinidas varias afinaciones comunes para el banjo: `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#-La-Re) y `banjo-open-dm-tuning` (Re menor al aire, la-Re-Fa-La-Re).

Estas afinaciones se pueden convertir a afinaciones para bajo de cuatro cuerdas utilizando la función `four-string-banjo`:

```
\set TabStaff.stringTunings = #(four-string-banjo banjo-c-tuning)
```

## Véase también

El archivo ‘scm/output-lib.scm’ contiene las afinaciones predefinidas para el banjo.

## 2.5 Percusión

### 2.5.1 Notación usual 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.

#### 2.5.1.1 Referencias para percusión

#### 2.5.1.2 Notación básica de percusión

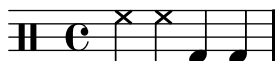
Las notas de percusión se pueden escribir en el modo `\drummode`, que es similar al modo estándar para introducir notas. Cada elemento de un set de percusión lleva un nombre completo y un nombre abreviado, y los dos se pueden usar en la entrada. La manera más fácil de escribir notas de percusión es utilizar la instrucción `\drums`, que crea el contexto y el modo de entrada apropiados para percusión:

```
\drums {
  hihat hh bassdrum bd
}
```



Esto es una abreviatura de

```
\new DrumStaff
\drummode {
  hihat hh bassdrum bd
}
```



La lista completa de nombre de instrumentos de percusión se encuentra en el archivo de inicio ‘ly/drumpitch-init.ly’.

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, `:32`, véase [\[Repeticiones de trémolo\]](#), página 83. He aquí un ejemplo de redobles de caja:

```
\new DrumStaff
\drummode {
  \time 2/4
```

```
sn16 sn8 sn16 sn8 sn8:32~ |
sn8 sn8 sn4:32~ |
sn4 sn8 sn16 sn16 |
sn4 r4 |
}
```



Los golpes de baqueta se pueden indicar mediante la colocación de `^"R"` o `^"L"` después de la nota. La propiedad `staff-padding` se puede sobrescribir para conseguir una línea de base satisfactoria.

## Véase también

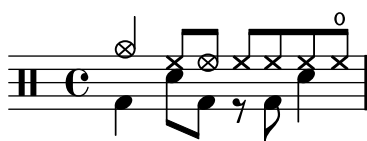
Referencia del programa: `note-event`.

### 2.5.1.3 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.

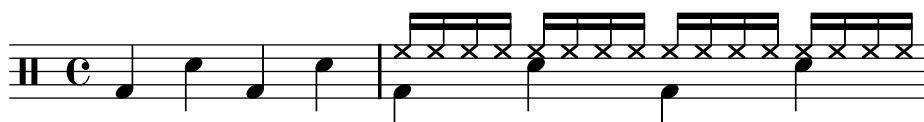
Para tipografiar la música, se deben interpretar las notas dentro de los contextos `DrumStaff` y `DrumVoice`

```
arriba = \drummode { crashcymbal4 hihat8 halfopenhihat hh hh hh openhihat }
abajo = \drummode { bassdrum4 snare8 bd r bd sn4 }
\new DrumStaff <<
  \new DrumVoice { \voiceOne \arriba }
  \new DrumVoice { \voiceTwo \abajo }
>>
```



El ejemplo anterior muestra una notación polifónica prolija. La notación polifónica abreviada, descrita en [Sección “I’m hearing Voices” in \*Manual de Aprendizaje\*](#), también se puede usar si las `DrumVoices` se instancian primero a mano. Por ejemplo,

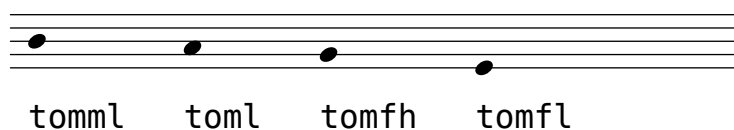
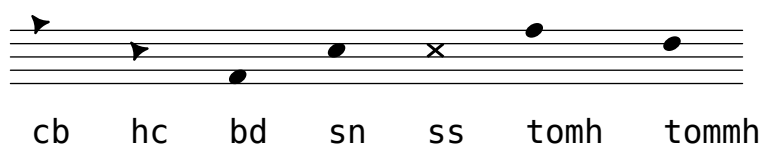
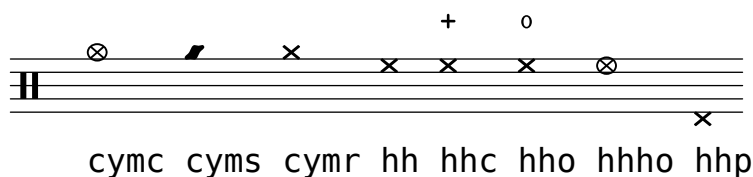
```
\new DrumStaff <<
  \new DrumVoice = "1" { s1 *2 }
  \new DrumVoice = "2" { s1 *2 }
  \drummode {
    bd4 sn4 bd4 sn4
    <<
      { \repeat unfold 16 hh16 }
      \\
      { bd4 sn4 bd4 sn4 }
    >>
  }
>>
```



También existen otras posibilidades en lo referente a la disposición. Para usarlas, establezca la propiedad `drumStyleTable` en el contexto `DrumVoice`. Se han predefinido las siguientes variables:

#### `drums-style`

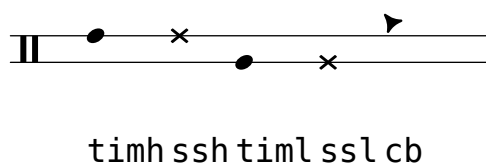
Es la opción por defecto. Tipografía un típico set de percusión sobre una pauta de cinco líneas



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`.

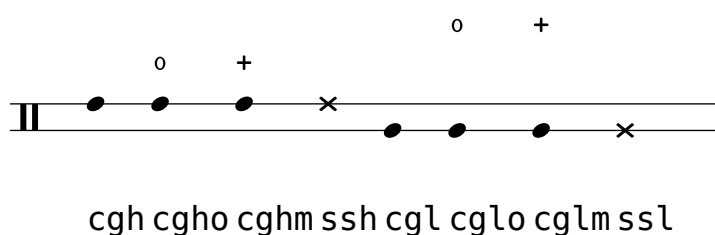
#### `timbales-style`

Esto tipografía timbales en una pauta de dos líneas:



#### `congas-style`

Esto tipografía congas en una pauta de dos líneas:



**bongos-style**

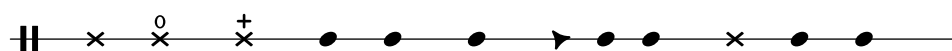
Esto tipografía bongos sobre una pauta de dos líneas:



boh boho boh mssh bol bolo bol mssh

**percussion-style**

Para tipografiar toda clase de percusiones simples sobre pautas de una línea.



tri trio trimgui guis guil cb cl tamb cab mar hc

Si no le gusta ninguna de las listas predefinidas, puede definir su propia lista al principio de su archivo

```
#(define mydrums '(
  (bassdrum      default    #f      -1)
  (snare         default    #f      0)
  (hihat         cross      #f      1)
  (pedalhihat    xcircle    "stopped" 2)
  (lowtom        diamond    #f      3)))

arriba = \drummode { hh8 hh hh hh hhp4 hhp }
abajo = \drummode { bd4 sn bd toml8 toml }

\new DrumStaff <<
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \new DrumVoice { \voiceOne \arriba }
  \new DrumVoice { \voiceTwo \abajo }
>>
```

**Véase también**

Archivos de inicio: 'ly/drumpitch-init.ly'.

Referencia de funcionamiento interno: DrumStaff, DrumVoice.

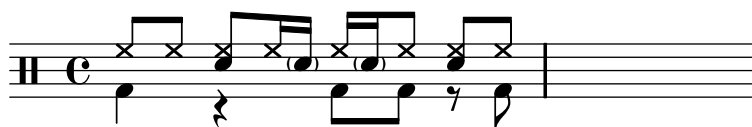
**Advertencias y problemas conocidos**

A causa de que el estándar General MIDI no contiene golpes de aro (rim shot), para este propósito se usa el golpe lateral de baqueta (sidestick) en su lugar.

### 2.5.1.4 Notas fantasma

Las notas fantasma para la batería e instrumentos de percusión se pueden crear utilizando la instrucción `\parenthesize` detallada en [Paréntesis], página 113. Sin embargo, el modo por defecto `\drummode` no incluye el complemento grabador `Parenthesis_engraver` que permite esto.

```
\new DrumStaff \with {
  \consists "Parenthesis_engraver"
} <<
\context DrumVoice = "1" { s1 *2 }
\context DrumVoice = "2" { s1 *2 }
\drummode {
  <<
    {
      hh8[ hh] <hh sn> hh16
      < \parenthesize sn > hh < \parenthesize
      sn > hh8 <hh sn> hh
    } \\\ {
      bd4 r4 bd8 bd r8 bd
    }
  >>
}
>>
```



Observe, además, que debe añadir acordes (paréntesis en ángulo < >) alrededor de cada una de las instrucciones `\parenthesize`.

## 2.6 Instrumentos de viento

### 2.6.1 Notación usual para instrumentos de viento

#### Referencias para instrumentos de viento

#### Digitaciones

### 2.6.2 Gaita

#### 2.6.2.1 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 `cflat` o `fflat`. De forma parecida, el sol agudo piobaireachd se puede escribir `gflat` cuando aparece en música ligera.

### 2.6.2.2 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.



### 2.7 Notación de acordes

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

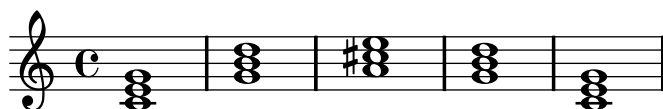
Para introducir acordes se utiliza el modo de acordes. Se usa un indicador de la estructura del acorde en lugar de la altura de sus notas.

#### Panorámica del modo de acordes

Los acordes se pueden escribir como música simultánea, como se explica en [Notas en acorde], página 85.

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 228.

```
\chordmode { c1 g a g c }
```

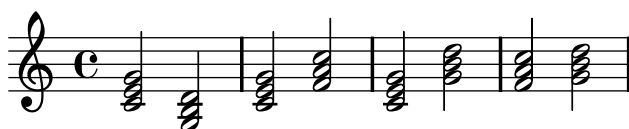




Los acordes escritos con el modo de acordes son elementos musicales, y se pueden transportar igual que los acordes escritos mediante construcciones de música simultánea.

Se pueden mezclar los modos de acorde y de nota en la música secuencial:

```
<c e g>2 <g b d>
\chordmode { c2 f}
<c e g> <g' b d>
\chordmode { f g}
```



## Véase también

Glossario musical: [Sección “acorde” in \*Glosario Musical\*](#).

Referencia de la notación: [\[Notas en acorde\]](#), página 85, [Sección 5.4.1 \[Modos de entrada\]](#), [página 228](#).

Fragmentos de código: [Sección “Chords” in \*Fragmentos de código\*](#)

## Advertencias y problemas conocidos

Cuando se mezclan los modos de acorde y de nota en música secuencial, y el modo de acordes aparece en primer lugar, el modo de notas crea un nuevo contexto de **Staff**.

```
\chordmode { c2 f}
<c e g> <g' b d>
```



Para evitar este comportamiento podemos crear explícitamente el contexto de **Staff**:

```
\new Staff {
  \chordmode { c2 f}
  <c e g> <g' b d>
}
```



## Acordes más usuales

Las tríadas mayores se introducen escribiendo la fundamental y una duración opcional:

```
\chordmode { c2 f4 g }
```



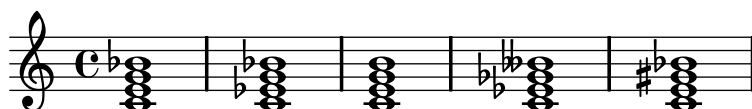
Las tríadas menores, aumentadas y disminuidas se escriben poniendo : y una cadena modificadora de variante después de la duración:

```
\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.

m	Acorde menor. Este modificador baja la tercera y, si existe, la séptima.
dim	Acorde disminuido. Este modificador baja la tercera, la quinta y, si existe, la séptima.
aug	Acorde aumentado. Este modificador eleva la quinta.
maj	Acorde de séptima mayor. Este modificador añade una séptima elevada. El 7 que sigue a <b>maj</b> es opcional. NO utilice este modificador para crear una tríada mayor.

## Véase también

Fragmentos de código: [Sección “Chords” in Fragmentos de código.](#)

## 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. 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.

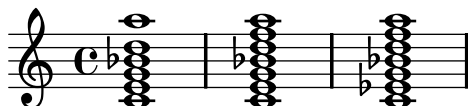
```
\chordmode {
  c:2 c:3 c:4 c:5
  c:6 c:7 c:8 c:9
  c:10 c:11 c:12 c:13
}
```



Observe que c:5 es idéntico a c (los dos producen una tríada de Do mayor).

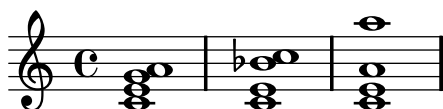
Puesto que una oncenena sin alteración no suena bien cuando se combina con una trecena sin alteración, se elimina la oncenena de los acordes de :13 (a no ser que se añada explícitamente).

```
\chordmode {
  c1:13 c:13.11 c:m13
}
```



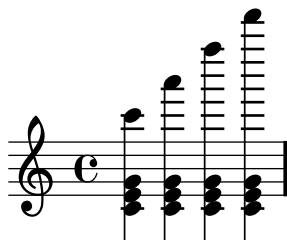
Se pueden añadir notas individuales a un acorde. Las adiciones siguen el ámbito y van prefijadas por un punto (.).

```
\chordmode {
  c1:5.6 c:3.7.8 c:3.6.13
}
```



Las notas añadidas pueden ser tan agudas como se desee.

```
\chordmode {
  c4:5.15 c:5.20 c:5.25 c: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  $\wedge$ . Sólo se permite una supresión con  $\wedge$  dentro de una cadena modificadora.

```
\chordmode {
  c1^3 c:7^5 c:9^3.5
}
```



Se puede añadir el modificador `sus` a la cadena modificadora para crear acordes suspendidos. Esto elimina la tercera del acorde. Esciba bien 2 o bien 4 para añadir la segunda o la cuarta al acorde, respectivamente. `sus` equivale a  $\wedge 3$ ; `sus4` equivale a  $\wedge 3.5$ .

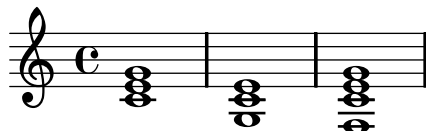
```
\chordmode {
  c1:sus c:sus2 c:sus4 c:5.4^3
}
```



Es posible eliminar la tercera con `sus` y eliminar otra nota con  $\wedge$  dentro de una cadena modificadora.

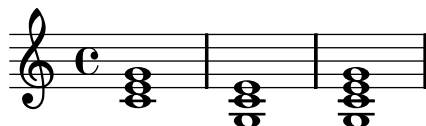
Se pueden especificar las inversiones (colocar en el bajo una nota distinta de la fundamental del acorde) y notas de bajo añadidas mediante la adición de `/nota` al acorde.

```
\chordmode {
  c1 c/g c/f
}
```



Se puede añadir una nota del bajo que forma parte del acorde, en lugar de ser movida por efecto de una inversión, mediante el uso de `/+nota`.

```
\chordmode {
  c1 c/g c/+g
}
```



## Véase también

Fragmentos de código: [Sección “Chords” in \*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.

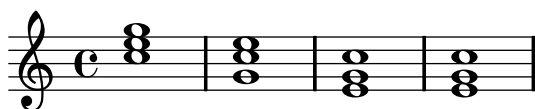
```
\chordmode { c1:5.5-.5+ }
```



Sólo se puede eliminar una nota de un acorde. Si se desea hacer un acorde con varias notas eliminadas, se debe construir a través de la adición de varias notas.

Sólo se puede crear la primera inversión mediante la adición de un bajo. La segunda inversión requiere cambiar la fundamental del acorde.

```
\chordmode {  
  c'1: c':/g e:6-3-^5 e:m6-^5  
}
```



### 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**

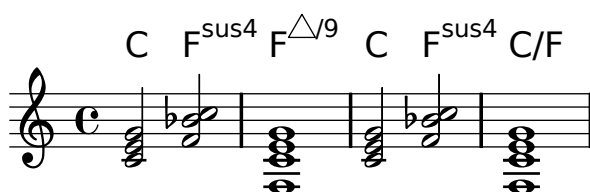
Los acordes se pueden escribir como notas simultáneas o a través del uso del modo de acordes. El nombre de acorde que se imprime es el mismo independientemente del modo de entrada, a no ser que existan inversiones o bajos añadidos:

```
<<  
  \new ChordNames {  
    <c e g>2 <f bes c>  
    <f c' e g>1
```

```

\chordmode {
  c2 f:sus4 c1:/f
}
}
{
  <c e g>2 <f bes c>
  <f, c' e g>1
  \chordmode {
    c2 f:sus4 c1:/f
  }
}
>>

```



`\chords { ... }` is a shortcut notation for `\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*

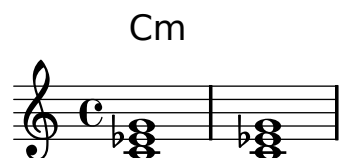
Se pueden imprimir los acordes exclusivamente al comienzo de las líneas y cuando cambia el acorde.

```

harmonies = \chordmode {
  c1:m c:m \break c:m c:m d
}
<<
\new ChordNames {
  \set chordChanges = ##t
  \harmonies
}
\new Staff {
  \relative c' { \harmonies }
}

```

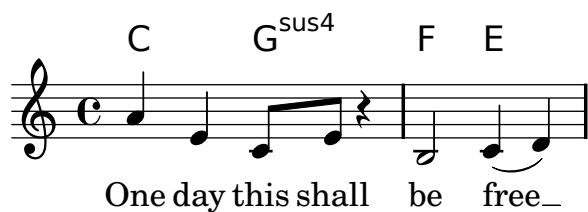
```
}
>>
```



*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 }
\relative c'' {
  a4 e c8 e r4
  b2 c4( d)
}
\addlyrics { One day this shall be free __ }
>>
```



## Véase también

Glosario musical: [Sección “acorde”](#) in *Glosario Musical*.

Referencia de la notación: [\[Escribir música en paralelo\]](#), página 95.

Fragmentos de código: [Sección “Chords”](#) in *Fragmentos de código*.

Referencia de funcionamiento interno: `ChordNames`, `Volta_engraver`, `Bar_engraver`.

## 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 [Apéndice A \[Lista bibliográfica\]](#), página 239). También funcionan otros dos esquemas de nomenclatura de acordes: una notación de acordes de Jazz alternativa y un esquema sistemático llamado Acordes de Banter. La notación de Jazz alternativa también se puede ver en el cuadro [Sección B.1 \[Carta de nombres de acordes\]](#), página 240.

Además de los distintos sistemas de nomenclatura, se usan diferentes nombres de nota para la fundamental en los distintos idiomas. Las variables predefinidas `\germanChords`, `\semiGermanChords`, `\italianChords` y `\frenchChords` establecen el valor de estas variables. El efecto se muestra aquí:

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>



Si ninguno de los ajustes predeterminados proporciona el resultado deseado, 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 cunción 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).

### `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 una barra inclinada. Mediante el ajuste de `chordNameSeparator`, podemos usar cualquier marcado que deseemos como separador.

**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.

**chordPrefixSpacer**

La ‘m’ de los acordes menores se imprime por lo general inmediatamente a la derecha de la fundamental. Estableciendo `chordPrefixSpacer`, podemos determinar un símbolo espaciador entre la fundamental y la ‘m’. El 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)" }
}

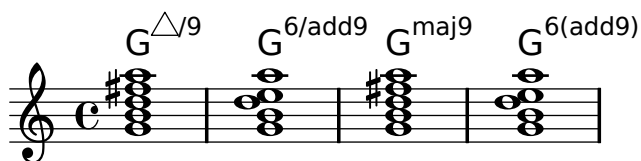
% Convert music to list and prepend to existing exceptions.
chExceptions = #( append
  ( sequential-music-to-chord-exceptions chExceptionMusic #t)
  ignatzekExceptions)

theMusic = \chordmode {
  g1:maj9 g1:6.9
  \set chordNameExceptions = #chExceptions
  g1:maj9 g1:6.9
}

\layout {
  ragged-right = ##t
}

<< \context ChordNames \theMusic
  \context Voice \theMusic
```

&gt;&gt;



The layout of the major 7 can be tuned with `majorSevenSymbol`.

```
\version "2.11.51"
\header {
  texidoc = "The layout of the major 7 can be tuned with
@code{majorSevenSymbol}."
}

\chords {
  c:7+
  \set majorSevenSymbol = \markup { "j7" }
  c:7+
}
```

$$C^{\Delta}C^{j7}$$

El separador entre las distintas partes del nombre de un acorde se puede establecer para que sea cualquier elemento de marcado.

```
\chords {
  c:7sus4
  \set chordNameSeparator
    = \markup { \typewriter "|" }
  c:7sus4
}
```

$$C^{7/sus4}C^{7|sus4}$$

## Véase también

Referencia de la notación: [Sección B.1 \[Carta de nombres de acordes\]](#), página 240.

Archivos instalados: ‘`scm/chords-ignatzek.scm`’, ‘`scm/chord-entry.scm`’, ‘`ly/chord-modifier-init.ly`’.

Fragmentos de código: [Sección “Chords”](#) in *Fragmentos de código*.

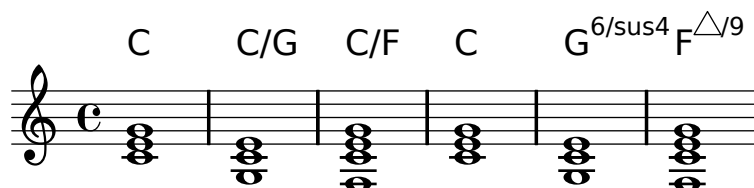
## Advertencias y problemas conocidos

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.

```

misAcordes = \relative c' {
  \chordmode{ c1 c/g c/f }
  <c e g> <g c e> <f c' e g>
}
<<
  \new ChordNames { \misAcordes }
  \new Staff { \misAcordes }
>>

```



### 2.7.3 Bajo cifrado

Se puede imprimir notación de bajo cifrado.

#### Introducción al bajo cifrado

LilyPond contempla el bajo cifrado, también conocido como basso continuo:

```

<<
  \new Voice { \clef bass dis4 c d ais g fis}
  \new FiguredBass {
    \figuremode {
      < 6 >4 < 7\+ >8 < 6+ [_!] >
      < 6 >4 <6 5 [3+] >
      < _ >4 < 6 5/>4
    }
  }
>>

```



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.

Las cifrase se crean como textos de marcado. Cualquiera de las propiedades estándar de marcado se pueden usar para modificar la impresión de las cifras. Por ejemplo, el espaciado vertical de las cifras se puede ajustar con `baseline-skip`.

## Introducir el bajo cifrado

Se usa `\figuremode` para cambiar el mod de entrada al modo de cifras. Hay más información sobre los distintos modos de entrada en [Sección 5.4.1 \[Modos de entrada\]](#), página 228.

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 añadir alteraciones accidentales (incluso becuadros) a las cifras:

```
\figures {
  <7! 6+ 4-> <5++> <3-->
}
```

**b7 x5 b3**  
**#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>
% FIXME: broken.
  %<6\\>
}
```

**6**

Se pueden incluir espacios verticales y corchetes en las cifras:

```
\figures {
  <[12 _!] 8 [6 4]>
}
```

**[12]**  
**[b]**  
**8**  
**[6]**  
**4]**



## 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**                **4b**                **4b**                **4b**

## Véase también

Fragmentos de código: [Sección “Chords” in Fragmentos de código](#).

Referencia de funcionamiento interno: `BassFigure`, `BassFigureAlignment`, `BassFigureLine`, `BassFigureBracket`, `BassFigureContinuation`, `FiguredBass`.

## 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' {
    c4 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.

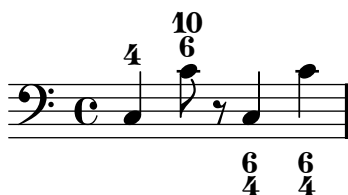
El bajo cifrado se puede añadir también a contextos de **Staff** directamente. En este caso, la posición vertical de las cifras se ajusta automáticamente.

```
<<
  \new Staff = myStaff
  \figuremode {
    <4>4 <10 6>8 s8
    <6 4>4 <6 4>
  }
  %% Poner las notas en el mismo pentagrama que los números
  \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>
  }
  %% Poner las notas en el mismo pentagrama que los números
  \context Staff = myStaff
  {
    \clef bass
    c4 c'8 r8 c4 c'
  }
>>
```



## Instrucciones predefinidas

`\bassFigureStaffAlignmentDown,` `\bassFigureStaffAlignmentUp,`  
`\bassFigureStaffAlignmentNeutral.`

## Fragmentos de código seleccionados

### Véase también

Sección “Chords” in *Fragmentos de código*.

Referencia de funcionamiento interno: `BassFigure`, `BassFigureAlignment`, `BassFigureLine`, `BassFigureBracket`, `BassFigureContinuation`, `FiguredBass`.

## 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
  % Los extensores aquí son correctos, con el mismo ritmo que el bajo
  \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
  % Los extensores aquí son incorrectos, aunque la medida es la misma
  <6 4->4 <6 4->4
  %\repeat unfold 4 {<6 4->16. <6 4->32}
  <5>8. r16 <6>8 <6\! 5->
}
>>
```







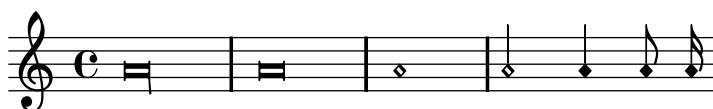
## 2.8.1 Antiguos

### 2.8.1.1 Cabezas de nota antiguas

Para la notación antigua, se puede elegir un estilo de cabezas de notas distinto al `default` (estilo por defecto). Esto se consigue estableciendo la propiedad `style` del objeto `NoteHead` a los valores `baroque`, `neomensural`, `mensural` o `petrucci`. El estilo `baroque` se diferencia del estilo por defecto `default` únicamente en que usa una forma cuadrada para las cabezas de las `\breve`. El estilo `neomensural` se diferencia del estilo `baroque` en que usa cabezas romboidales para las redondas y todas las figuras de duración menor. Las plicas aparecen centradas sobre las cabezas. Este estilo es particularmente útil al transcribir música mensural, p.ej. para el incipit. El estilo `mensural` produce cabezas que imitan el aspecto de aquellas que aparecen en las ediciones impresas del siglo XVI. Finalmente, el estilo `petrucci` también imita las ediciones históricas, pero utiliza unas cabezas mayores.

El ejemplo siguiente muestra el estilo `neomensural`

```
\set Score.skipBars = ##t
\override NoteHead #'style = #'neomensural
a'\longa a'\breve a'1 a'2 a'4 a'8 a'16
```



Cuando se tipografía una pieza en notación de Canto Gregoriano, el grabador `Vaticana_ligature_engraver` selecciona automáticamente las cabezas adecuadas, de manera que no es necesario establecer el estilo de las cabezas explícitamente. Aún así, el estilo de las cabezas se puede establecer, p.ej., a `vaticana_punctum` para producir neumas punctum. De forma similar, se utiliza un grabador `Mensural_ligature_engraver` para construir automáticamente ligaduras mensurales. Consulte [Sección 2.8.1.10 \[Ligaduras\]](#), página 188 para ver cómo funcionan los grabadores de las ligaduras.

## Véase también

[Sección B.5 \[Estilos de cabezas de nota\]](#), página 257 ofrece una visión de conjunto sobre la totalidad de los estilos de cabeza disponibles.

### 2.8.1.2 Alteraciones accidentales antiguos

Utilice la propiedad `glyph-name-alist` del elemento gráfico `Accidental` y `KeySignature` para seleccionar los símbolos de alteración de la notación antigua.

**vaticana medicaea hufnagel mensural**



Como se muestra aquí, no todas las alteraciones están contempladas por cada uno de los estilos. Cuando trate de acceder a una alteración no contemplada, LilyPond cambiará a un estilo diferente.

De forma similar a las alteraciones accidentales, el estilo de la armadura de la tonalidad se puede controlar a través de la propiedad `glyph-name-alist` del elemento gráfico `KeySignature`.

## Véase también

En el presente manual: [Sección 1.1 \[Alturas\]](#), página 1, [\[Alteraciones accidentales\]](#), página 4 y [\[Alteraciones accidentales automáticas\]](#), página 18 proporcionan una introducción general al uso de las alteraciones. [\[Armadura de la tonalidad\]](#), página 14 ofrece una introducción general al uso de las armaduras de tonalidad.

Referencia del programa: `KeySignature`.

Ejemplos: [Sección “Ancient notation” in \*Fragmentos de código\*](#).

### 2.8.1.3 Silencios antiguos

Utilice la propiedad `style` del elemento gráfico `Rest` para seleccionar silencios antiguos. Los estilos contemplados son `classical`, `neomensural` y `mensural`. `classical` se diferencia del estilo por defecto `default` solamente en que el silencio de negra parece un silencio de corchea invertido horizontalmente. El estilo `neomensural` es adecuado para, p.ej., el incipit de una pieza transcrita de música mensural. Finalmente, el estilo `mensural` reproduce el aspecto de los silencios de las ediciones impresas históricas del siglo XVI.

El ejemplo siguiente muestra el estilo `neomensural`

```
\set Score.skipBars = ##t
\override Rest #'style = #'neomensural
r\longa r\breve r1 r2 r4 r8 r16
```



No existen silencios de fusa y semifusa específicos para los estilos mensural ni neo-mensural. En su lugar se tomarán los silencios del estilo por defecto.

No hay silencios en la notación del Canto Gregoriano; en su lugar, se utilizan [Sección 2.8.1.9 \[Divisiones\]](#), página 188.

## Véase también








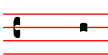

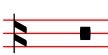
En este manual: [Sección 1.2.2.1 \[Silencios\]](#), página 37 ofrece una introducción general al uso de los silencios.

### 2.8.1.4 Claves antiguas

LilyPond contempla una amplia variedad de claves, muchas de ellas antiguas.

La siguiente tabla presenta todas las claves antiguas que están contempladas a través de la instrucción `\clef`. Algunas de las claves usan el mismo glifo, pero se diferencian sólo en la línea en que se imprimen. En tales casos, se usa un número añadido al nombre para enumerar dichas claves. Aún así, puede forzar manualmente que se escriba un glifo de clave sobre una línea arbitraria, como se encuentra descrito en [\[Clave\]](#), página 11. La nota impresa a la derecha de cada clave en la columna de los ejemplos denota la situación del Do central (c') con respecto a dicha clave.

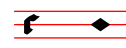
Descripción	Claves posibles	Ejemplo
-------------	-----------------	---------

clave de Do en estilo mensural moderno	neomensural-c1, neomensural-c2, neomensural-c3, neomensural-c4	
claves de Do mensurales en estilo de Petrucci, para su uso sobre diferentes líneas de la pauta (los ejemplos muestran la clave de Do en segunda línea)	petrucci-c1, petrucci-c2, petrucci-c3, petrucci-c4, petrucci-c5	
clave de Fa mensural en estilo de Petrucci	petrucci-f	
clave de Sol mensural en estilo de Petrucci	petrucci-g	
clave de Do mensural en estilo histórico	mensural-c1, mensural-c2, mensural-c3, mensural-c4	
clave de Fa mensural en estilo histórico	mensural-f	
clave de Sol mensural de estilo histórico	mensural-g	
clave de Do del estilo de la Editio Vaticana	vaticana-do1, vaticana-do2, vaticana-do3	
clave de Fa del estilo de la Editio Vaticana	vaticana-fa1, vaticana-fa2	
clave de Do del estilo de la Editio Medicaea	medicaea-do1, medicaea-do2, medicaea-do3	

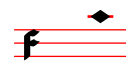
clave de Fa del estilo de la Editio Medicaea `medicaea-fa1, medicaea-fa2`



clave de Do hufnagel de estilo histórico `hufnagel-do1, hufnagel-do2, hufnagel-do3`



clave de Fa hufnagel de estilo histórico `hufnagel-fa1, hufnagel-fa2`



clave hufnagel de Do y Fa combinados de estilo histórico `hufnagel-do-fa`



*Estilo Moderno* significa “como se encuentra tipografiado en las ediciones contemporáneas de música mensural transcrita.”

*Estilo de Petrucci* significa “inspirado por las ediciones impresas del famoso grabador Petrucci (1466-1539).”

*Estilo Histórico* significa “como se tipografiaba o se escribía en las ediciones históricas (aparte de las de Petrucci).”

*Estilo de la Editio XXX* significa “como se imprime o imprimía en la Editio XXX.”

Petrucci usaba claves de Do que tenían barras verticales en el lado izquierdo en un equilibrio variable, dependiendo de la línea de la pauta en que se imprimían.

## Véase también

En este manual: consulte [\[Clave\]](#), página 11.

## Advertencias y problemas conocidos

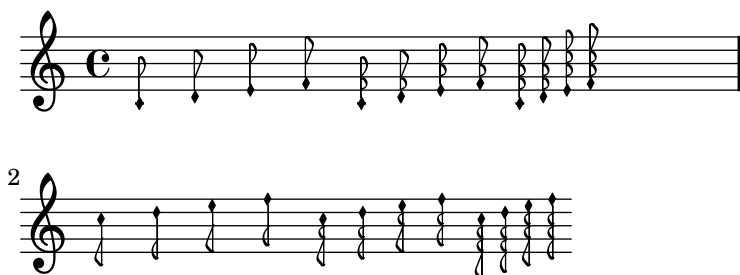
La clave mensural de Sol está reasignada a la clave de Sol de Petrucci.

### 2.8.1.5 Corchetes antiguos

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`

```
\override Stem #'flag-style = #'mensural
\override Stem #'thickness = #1.0
\override NoteHead #'style = #'mensural
\autoBeamOff
c'8 d'8 e'8 f'8 c'16 d'16 e'16 f'16 c'32 d'32 e'32 f'32 s8
```

c''8 d''8 e''8 f''8 c''16 d''16 e''16 f''16 c''32 d''32 e''32 f''32



Dese cuenta de que la bandera más interna de cada uno de los corchetes mensurales siempre se alinea verticalmente con una línea de la pauta.

No existe un estilo particular de corchete para la notación neo-mensural. Por ello, cuando tipografíe el incipit de una pieza transcrita de música mensural, debe usar el estilo de corchete por defecto. No existen corchetes en la notación del Canto Gregoriano.

## Advertencias y problemas conocidos

La unión de los corchetes antiguos a las plicas está ligeramente desviada a causa de un cambio en la antigua versión 2.3.x.

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).

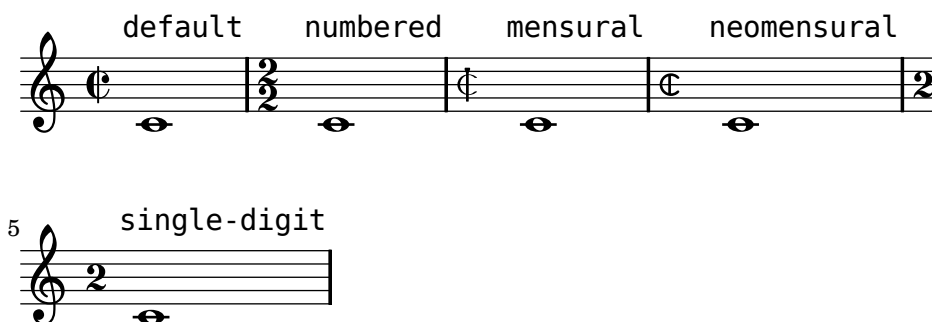
### 2.8.1.6 Indicaciones antiguas de compás

Las indicaciones de compás mensurales están contempladas de forma limitada. Los glifos están configurados de forma fija para determinadas fracciones de tiempo. En otras palabras, para obtener un glifo de compás mensural determinado con la instrucción `\time n/m`, se tienen que escoger  $n$  y  $m$  según la siguiente tabla

$\text{C}$	$\text{C}$	$\text{C}$	$\text{C}$
<code>\time 4/4</code>	<code>\time 6/4</code>	<code>\time 2/2</code>	<code>\time 6/8</code>
$\text{O}$	$\text{O}$	$\text{O}$	$\text{O}$
<code>\time 3/2</code>	<code>\time 3/4</code>	<code>\time 9/4</code>	<code>\time 9/8</code>
$\text{O}$	$\text{O}$		
<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`. Este estilo es adecuado para el incipit de las transcripciones de piezas mensurales. El estilo `mensural` reproduce el aspecto de las ediciones impresas históricas del s. XVI.

Los ejemplos siguientes muestran las diferencias que existen entre los estilos:



## Véase también

En el presente manual: [Sección 1.2.3.1 \[Indicación de compás\]](#), [página 43](#) ofrece una introducción general al uso de las indicaciones de compás.

## Advertencias y problemas conocidos

Las relaciones de duración de las notas no se modifican cuando cambia la indicación de compás. Por ejemplo, la razón de 1 breve = 3 semibreves (tiempo perfecto) se debe hacer a mano mediante el establecimiento de

```
breveTP = #(ly:make-duration -1 0 3 2)
...
{ c\breveTP f1 }
```

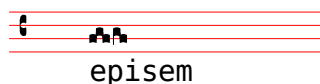
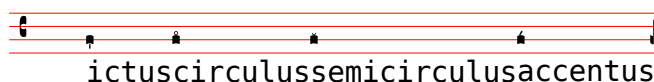
Esto fija un valor para `breveTP` de  $3/2$  multiplicado por  $2 = 3$  veces una redonda.

El símbolo `old6/8alt` (alternativo al símbolo de  $6/8$ ) no se puede seleccionar directamente con `\time`. En su lugar, utilice un elemento de marcado `\markup`.

### 2.8.1.7 Articulaciones antiguas

Además de los signos de articulación estándar que se hallan descritos en la sección [\[Articulaciones y ornamentos\]](#), [página 64](#), se proveen signos de articulación para la notación antigua. Éstos están pensados específicamente para utilizarlos con la notación del estilo Editio Vaticana.

```
\include "gregorian-init.ly"
\score {
  \new VaticanaVoice {
    \override TextScript #'font-family = #'typewriter
    \override TextScript #'font-shape = #'upright
    \override Script #'padding = #-0.1
    a\ictus_"ictus" \break
    a\circulus_"circulus" \break
    a\semicirculus_"semicirculus" \break
    a\accentus_"accentus" \break
    \[ a_"episem" \episemInitium \pes b \flexa a b \episemFinis \flexa a \]
  }
}
```



## Advertencias y problemas conocidos

Ciertas articulaciones se colocan demasiado cerca verticalmente de las cabezas de nota correspondientes.

La línea episemática no se muestra en muchos casos. Si se presenta, el extremo derecho de la línea episemática se encuentra a menudo demasiado lejos hacia la derecha.

### 2.8.1.8 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), como se muestra en el ejemplo siguiente:

```
\layout {
  \context {
    \Staff
    \consists Custos_engraver
    Custos \override #'style = #'mensural
  }
}
```

El resultado tendrá el aspecto siguiente:



El glifo del custos se selecciona mediante la propiedad `style`. Los estilos contemplados son *vaticana*, *medicaea*, *hufnagel* y *mensural*. Se muestran en el siguiente fragmento:

`vaticana medicaea hufnagel mensural`





## Véase también

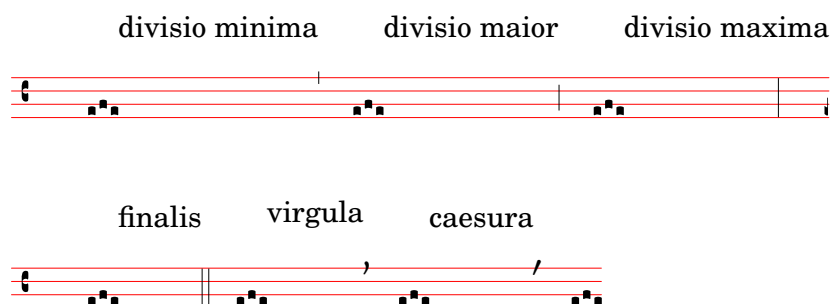
Referencia del programa: `Custos`.

Ejemplos: Sección “Ancient notation” in *Fragmentos de código*.

### 2.8.1.9 Divisiones

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 72. 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-init.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-init.ly’ define también las instrucciones `\virgula` y `\caesura`



## Instrucciones predefinidas

`\virgula`, `\caesura`, `\divisioMinima`, `\divisioMaior`, `\divisioMaxima`, `\finalis`.

## Véase también

En el presente manual: [Marcas de respiración], página 72.

Referencia del programa: `BreathingSign`.

Ejemplos: Sección “Winds” in *Fragmentos de código*.

### 2.8.1.10 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.

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. Por defecto, el grabador `LigatureBracket` se limita a colocar un corchete recto sobre la ligadura

```
\transpose c c' {
  \[ g c a f d' \]
  a g f
  \[ e f a g \]
}
```



Para seleccionar un estilo de ligaduras específico, se debe añadir el grabador adecuado al contexto `Voice`, tal y como se explica en los sub-apartados siguientes. Solamente están contempladas las ligaduras mensurales blancas con ciertas limitaciones.

## Advertencias y problemas conocidos

Las ligaduras necesitan un espaciado especial que aún no se ha desarrollado. Como consecuencia, la mayor parte del tiempo existe una separación excesiva entre las ligaduras, y a menudo los cortes de línea no son satisfactorios. Además, la letra de los cantos no se alinea correctamente con las ligaduras.

Las alteraciones accidentales no se deben imprimir dentro de las ligaduras, sino que deben reunirse e imprimirse juntas delante de ellas.

La sintaxis todavía utiliza el obsoleto estilo infijo `\[ expresión_musical \]`. Por motivos de consistencia, esto cambiará algún día al estilo postfijo `nota\[ ... nota\]`. De forma alternativa, puede incluirse el archivo `'gregorian-init.ly'`; proporciona una función de Scheme

```
\ligature expresión_musical
```

que causa el mismo efecto y se supone que es lo bastante estable.

### 2.8.1.11 Ligaduras mensurales blancas

Las ligaduras mensurales blancas están contempladas de forma limitada.

Para grabar ligaduras mensurales blancas, en el bloque layout o de disposición de la página coloque el grabador `Mensural_ligature_engraver` dentro del contexto `Voice`, y elimine el grabador `Ligature_bracket_engraver`, de la siguiente manera:

```
\layout {
  \context {
    \Voice
    \remove Ligature_bracket_engraver
    \consists Mensural_ligature_engraver
  }
}
```

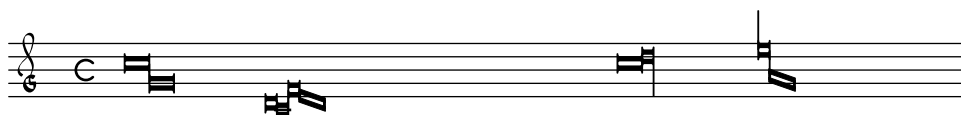
No existe ningún lenguaje de entrada adicional para describir la forma de una ligadura mensural blanca. Más bien la forma se determina exclusivamente a partir de la altura y duración de las notas comprendidas. Si bien este enfoque podría hacer que un usuario con poca experiencia tardase un tiempo en 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.

Por ejemplo,

```

\set Score.timing = ##f
\set Score.defaultBarType = "empty"
\override NoteHead #'style = #'neomensural
\override Staff.TimeSignature #'style = #'neomensural
\clef "petrucci-g"
\[ c'\maxima g \]
\[ d\longa c\breve f e d \]
\[ c'\maxima d'\longa \]
\[ e'1 a g\breve \]

```



Si no se sustituye el grabador `Ligature_bracket_engraver` por el `Mensural_ligature_engraver`, la misma música se transcribe de la siguiente manera:



## Advertencias y problemas conocidos

El espaciado horizontal es pobre.

### 2.8.1.12 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.

La tabla siguiente contiene los neumas extendidos del segundo volumen del Antiphonale Romanum (*Liber Hymnarius*), publicado en el año 1983 por los monjes de Solesmes.

Neuma aut Neumarum Elementa	Figurae Rectae	Figurae Liquescentes Auctae	Figurae Liquescentes Deminutae
1. Punctum	a b ■ ◆	c d e ■ ■ ◆	f ◆
2. Virga	g ■		

### 3. Apostropha vel Stropha

h	i
◊	◊

### 4. Oriscus

j
~

### 5. Clivis vel Flexa

	l m	
h	h h	h
k		n

### 6. Podatus vel Pes

	p q	
h	h h	h
o		r

### 7. Pes Quassus

h	h
s	t

### 8. Quilisma Pes

h	h
u	v

9. Podatus Initio Debilis



W



X

10. Torculus



y



z



A

11. Torculus Initio Debilis



B



C



D

12. Porrectus



E



F



G

13. Climacus



H

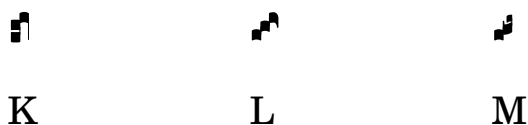


I



J

## 14. Scandicus



## 15. Salicus



## 16. Trigonus



Al contrario que la mayor parte de los otros sistemas de notación de neumas, el lenguaje de entrada para los neumas no refleja la apariencia tipográfica, sino que está pensado para mantener la atención sobre el significado musical. Por ejemplo, `\[ a \pes b \flexa g \]` produce un Torculus que consiste en tres cabezas de Punctum, mientras que `\[ a \flexa g \pes b \]` 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.

La tabla siguiente muestra los fragmentos de código que producen las ligaduras en la anterior tabla de neumas. La letra de la primera columna de cada línea en la tabla siguiente, indica a qué ligadura de la tabla anterior se refiere. La segunda columna da el nombre de la ligadura. La tercera columna muestra el fragmento de código que produce esta ligadura, utilizando `g`, `a` `b` (Sol, La y Si) como notas de ejemplo.

#	Nombre	Frase de la entrada
a	Punctum	<code>\[ b \]</code>
b	Punctum Inclinatorum	<code>\[ \inclinatorum b \]</code>
c	Punctum Auctum Ascendens	<code>\[ \auctum \ascendens b \]</code>
d	Punctum Auctum Descendens	<code>\[ \auctum \descendens b \]</code>

e	Punctum Inclinatorum Auctum	<code>\[ \inclinatorum \auctum b \]</code>
f	Punctum Inclinatorum Parvum	<code>\[ \inclinatorum \deminutum b \]</code>
g	Virga	<code>\[ \virga b \]</code>
h	Stropha	<code>\[ \stropha b \]</code>
i	Stropha Aucta	<code>\[ \stropha \auctum b \]</code>
j	Oriscus	<code>\[ \oriscus b \]</code>
k	Clivis vel Flexa	<code>\[ b \flexa g \]</code>
l	Clivis Aucta Descendens	<code>\[ b \flexa \auctum \descendens g \]</code>
m	Clivis Aucta Ascendens	<code>\[ b \flexa \auctum \ascendens g \]</code>
n	Cephalicus	<code>\[ b \flexa \deminutum g \]</code>
o	Podatus vel Pes	<code>\[ g \pes b \]</code>
p	Pes Auctus Descendens	<code>\[ g \pes \auctum \descendens b \]</code>
q	Pes Auctus Ascendens	<code>\[ g \pes \auctum \ascendens b \]</code>
r	Epiphonus	<code>\[ g \pes \deminutum b \]</code>
s	Pes Quassus	<code>\[ \oriscus g \pes \virga b \]</code>
t	Pes Quassus Auctus Descendens	<code>\[ \oriscus g \pes \auctum \descendens b \]</code>
u	Quilisma Pes	<code>\[ \quilisma g \pes b \]</code>
v	Quilisma Pes Auctus Descendens	<code>\[ \quilisma g \pes \auctum \descendens b \]</code>
w	Pes Initio Debilis	<code>\[ \deminutum g \pes b \]</code>
x	Pes Auctus Descendens Initio Debilis	<code>\[ \deminutum g \pes \auctum \descendens b \]</code>
y	Torculus	<code>\[ a \pes b \flexa g \]</code>
z	Torculus Auctus Descendens	<code>\[ a \pes b \flexa \auctum \descendens g \]</code>
A	Torculus Deminutus	<code>\[ a \pes b \flexa \deminutum g \]</code>
B	Torculus Initio Debilis	<code>\[ \deminutum a \pes b \flexa g \]</code>
C	Torculus Auctus Descendens Initio Debilis	<code>\[ \deminutum a \pes b \flexa \auctum \descendens g \]</code>
D	Torculus Deminutus Initio Debilis	<code>\[ \deminutum a \pes b \flexa \deminutum g \]</code>

E	Porrectus	<code>\[ a \flexa g \pes b \]</code>
F	Porrectus Auctus Descendens	<code>\[ a \flexa g \pes \auctum \descendens b \]</code>
G	Porrectus Deminutus	<code>\[ a \flexa g \pes \deminutum b \]</code>
H	Climacus	<code>\[ \virga b \inclinatum a \inclinatum g \]</code>
I	Climacus Auctus	<code>\[ \virga b \inclinatum a \inclinatum \auctum g \]</code>
J	Climacus Deminutus	<code>\[ \virga b \inclinatum a \inclinatum \deminutum g \]</code>
K	Scandicus	<code>\[ g \pes a \virga b \]</code>
L	Scandicus Auctus Descendens	<code>\[ g \pes a \pes \auctum \descendens b \]</code>
M	Scandicus Deminutus	<code>\[ g \pes a \pes \deminutum b \]</code>
N	Salicus	<code>\[ g \oriscus a \pes \virga b \]</code>
O	Salicus Auctus Descendens	<code>\[ g \oriscus a \pes \auctum \descendens b \]</code>
P	Trigonus	<code>\[ \stroph a b \stroph a b \stroph a \]</code>

Las ligaduras que se acaban de relacionar sirven principalmente como un conjunto limitado, pero representativo, de ejemplos de ligaduras gregorianas. Dentro de los delimitadores de ligadura `\[` y `\]`, prácticamente cualquier cantidad de notas se pueden acumular para formar una única ligadura, y los prefijos de cabeza como `\pes`, `\flexa`, `\virga`, `\inclinatum`, etc. se pueden intermezclar como se desee. La utilización del conjunto de reglas que subyace a la construcción de las ligaduras en la tabla anterior se extrapola de la forma correspondiente. De esta manera se pueden crear una cantidad infinita de ligaduras diferentes.

Los puntos de aumentación, llamados también *morae*, se escriben con la función musical `\augmentum`. Dese cuenta de que `\augmentum` está programado como una función musical unaria y no como un prefijo principal. 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-init.ly"
\score {
  \new VaticanaVoice {
    \[ \augmentum a \flexa \augmentum g \]
    \augmentum g
  }
}
```



## Instrucciones predefinidas

Están contemplados los siguientes prefijos de cabeza:





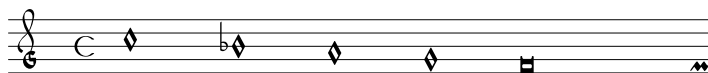


San- ctus

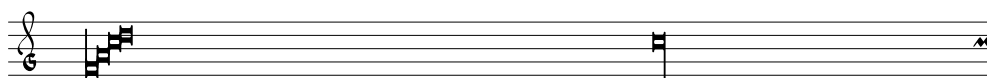
### 2.8.1.14 Contextos de la música mensural

Los contextos predefinidos de voz `MensuralVoiceContext` y de pauta `MensuralStaffContext` 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

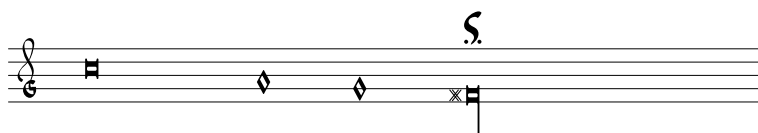
```
\score {
  <<
    \new MensuralVoice = "discantus" \transpose c c' {
      \override Score.BarNumber #'transparent = ##t {
        c'1\melisma bes a g\melismaEnd
        f\breve
        \[ f1\melisma a c'\breve d'\melismaEnd \]
        c'\longa
        c'\breve\melisma a1 g1\melismaEnd
        fis\longa^\signumcongruentiae
      }
    }
    \new Lyrics \lyricsto "discantus" {
      San -- ctus, San -- ctus, San -- ctus
    }
  >>
}
```



San - - ctus,



San - - ctus,



San - - ctus

### 2.8.1.15 Alteraciones de la Musica Ficta

En la música europea de hasta aprox. 1600, se esperaba de los cantantes que alterasen cromáticamente ciertas notas de acuerdo con su propia iniciativa. 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 cierto.

```
fis gis  
\set suggestAccidentals = ##t  
ais bis
```



## Véase también

Referencia del programa: el grabador `Accidental_engraver` y el objeto `AccidentalSuggestion`.

## 2.9 Músicas del mundo

### 2.9.1 Música árabe

References for Arabic music

Arabic note names

Arabic key signatures

Arabic time signatures

Arabic music example

Lecturas adicionales

## 3 Entrada y salida generales

### 3.1 Estructura del código de entrada

#### 3.1.1 Estructura de una partitura

#### 3.1.2 Varias partituras en un libro

#### 3.1.3 Estructura del archivo

### 3.2 Títulos y cabeceras

Casi toda la música impresa tiene un título y el nombre del compositor; ciertas piezas tienen mucha más información.

#### 3.2.1 Crear títulos

Se crean títulos para cada uno de los bloques `\score` así como para todo el archivo de entrada (o bloque `\book` (libro)).

El contenido de cada título se extrae a partir de los bloques `\header` (cabecera). El bloque de cabecera de un libro contempla lo siguiente:

`dedication`

El dedicatario de la música, centrado en lo alto de la primera página.

`title` El título de la música, centrado justo debajo de la dedicatoria.

`subtitle` Subtítulo, centrado debajo del título.

`subsubtitle`

Sub-subtítulo, centrado debajo del subtítulo.

`poet` Nombre del poeta, a la izquierda, debajo del subtítulo.

`composer` Nombre del compositor, a la derecha, debajo del subtítulo.

`meter` Texto de la medida, a la izquierda, debajo del poeta.

`opus` Nombre del Opus, a la derecha, debajo del compositor.

`arranger` Nombre del arreglista, a la derecha, debajo del Opus.

`instrument`

Nombre del instrumento, centrado, debajo del arreglista. También aparece centrado en lo alto de cada página (aparte de la primera).

`piece` Nombre de la pieza, a la izquierda, debajo del instrumento.

`breakbefore`

Esto fuerza que el título comience sobre una página nueva (establézcalo a `##t` o a `##f`).

`copyright`

Aviso de Copyright, centrado en la parte inferior de la primera página. Para insertar el símbolo de copyright symbol, consulte [Sección 3.3.3 \[Codificación del texto\]](#), [página 206](#).

`tagline` Cartel de propósito general en el pie de página, centrado al final de la última.

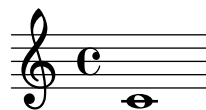
A continuación presentamos una demostración de todos los campos que están disponibles. Fíjese en que puede utilizar en la cabecera cualquier instrucción de marcado de texto (Véase [Sección 1.8.2 \[Formatear el texto\]](#), página 125).

```
\paper {
  line-width = 9.0\cm
  paper-height = 10.0\cm
}

\book {
  \header {
    dedication = "dedicated to me"
    title = \markup \center-column { "Title first line" "Title second line,
longer" }
    subtitle = "the subtitle,"
    subsubtitle = #(string-append "subsubtitle LilyPond version "
(lilypond-version))
    poet = "Poet"
    composer = \markup \center-column { "composer" \small "(1847-1973)" }
    texttranslator = "Text Translator"
    meter = \markup { \teeny "m" \tiny "e" \normalsize "t" \large "e" \huge
"r" }
    arranger = \markup { \fontsize #8.5 "a" \fontsize #2.5 "r" \fontsize
#-2.5 "r" \fontsize #-5.3 "a" \fontsize #7.5 "nger" }
    instrument = \markup \bold \italic "instrument"
    piece = "Piece"
  }

  \score {
    { c'1 }
    \header {
      piece = "piece1"
      opus = "opus1"
    }
  }
  \markup {
    and now...
  }
  \score {
    { c'1 }
    \header {
      piece = "piece2"
      opus = "opus2"
    }
  }
}
```

dedicated to me  
**Title first line**  
**Title second line, longer**  
the subtitle,  
subsubtitle LilyPond version 2.11.57  
Poet *instrument* composer  
meter *a r r a nger* (1847-1973)  
piece1 opus1



2 *instrument*

and now...

piece2 opus2



Music engraving by LilyPond 2.11.57—[www.lilypond.org](http://www.lilypond.org)

Como se mostró anteriormente, puede usar varios bloques `\header`. Cuando los mismos campos aparecen en distintos bloques, tiene validez el último en definirse. A continuación un breve ejemplo.

```
\header {
  composer = "Compositor"
}
\header {
  piece = "Pieza"
}
\score {
  \new Staff { c'4 }
  \header {
    piece = "Otra pieza" % sobreescribe a la anterior
  }
}
```

Si define la cabecera `\header` dentro del bloque `\score`, por lo general se imprimirán solamente las cabeceras `piece` y `opus`. Fíjese en que la expresión musical debe aparecer antes del `\header`.

```
\score {
  { c'4 }
  \header {
    title = "title" % no impreso
    piece = "piece"
    opus = "opus"
  }
}
```

**piece**

**opus**



Se puede cambiar este comportamiento (e imprimir todas las cabeceras cuando la cabecera `\header` está definida dentro del bloque `\score`) mediante la utilización de

```
\paper{
  printallheaders=##t
}
```

El pie de página por omisión se encuentra vacío, excepto para la primera página, en la que se inserta el campo de `copyright` extraído del `\header`, y para la última página, en la que se escribe el cartel `tagline` sacado del bloque `\header`. La línea por omisión para este cartel es “Music engraving by LilyPond (*version*)”.<sup>1</sup>

Las cabeceras se pueden quitar completamente estableciendo su valor a falso.

```
\header {
  tagline = ##f
  composer = ##f
}
```

### 3.2.2 Títulos personalizados

Una opción más avanzada es cambiar las definiciones de las variables siguientes dentro del bloque `\paper` block. El archivo de inicio ‘`ly/titling-init.ly`’ ofrece un listado de la disposición predeterminada.

**bookTitleMarkup**

Es el título que se escribe en lo alto del documento de salida completo. Normalmente contiene el compositor y el título de la pieza.

**scoreTitleMarkup**

Es el título que se coloca por encima de un bloque `\score`. normalmente contiene el nombre del movimiento (campo `piece`).

**oddHeaderMarkup**

Es el encabezamiento de las páginas de numeración impar.

<sup>1</sup> Las partituras bien tipografiadas son una magnífica promoción para nosotros, así que por favor, si puede, conserve intacta la línea del letrero.

**evenHeaderMarkup**

Es el encabezamiento de las páginas de numeración par. Si se deja sin especificar, se usará el encabezamiento de las páginas impares.

De forma predeterminada, los encabezamientos se definen de tal forma que el número de página está en el borde exterior, y el instrumento está centrado.

**oddFooterMarkup**

Es el pie de las páginas de numeración impar.

**evenFooterMarkup**

Es el pie de las páginas de numeración par. Si se deja sin especificar, se usará en su lugar el encabezamiento de las páginas impares.

De forma predeterminada, el pie de página tiene el aviso de copyright en la primera página, y el cartel **tagline** en la última.

La definición siguiente pone el título en el extremo izquierdo, y el compositor en el extremo derecho sobre una única línea.

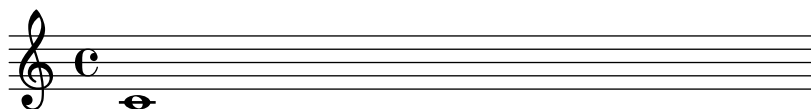
```
\paper {
  bookTitleMarkup = \markup {
    \fill-line {
      \fromproperty #'header:title
      \fromproperty #'header:composer
    }
  }
}
```

### 3.2.3 Referencias a números de página

Se puede marcar un lugar determinado de una partitura utilizando la instrucción `\label` (etiqueta), bien en lo alto de la estructura o bien dentro de la música. Posteriormente se puede hacer referencia a esta etiqueta dentro de un elemento de marcado, para obtener el número de la página en que se encuentra la marca, usando la instrucción de marcado `\page-ref`.

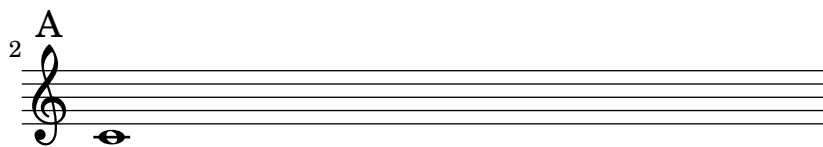
```
\header { tagline = ##f }
\book {
  \label #'firstScore
  \score {
    {
      c'1
      \pageBreak \mark A \label #'markA
      c'
    }
  }
}

\markup { The first score begins on page \page-ref #'firstScore "0" "?" }
\markup { Mark A is on page \page-ref #'markA "0" "?" }
}
```





2



The first score begins on page 1

Mark A is on page 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`

### 3.2.4 Índice general

Se puede insertar un índice general o tabla de contenidos utilizando la instrucción `\markuplines \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.

`\markuplines \table-of-contents`

`\pageBreak`

`\tocItem \markup "Primera partitura"`

`\score {`

`{`

`c' % ...`

`\tocItem \markup "Un punto concreto dentro de la primera partitura"`

`d' % ...`

`}`

`}`

`\tocItem \markup "Segunda partitura"`

`\score {`

`{`

`e' % ...`

`}`

`}`

Los elementos de marcado que se usan para dar formato al índice general se encuentran definidos dentro del bloque `\paper`. Los elementos predeterminados son `tocTitleMarkup`, para dar formato al título de la tabla, y `tocItemMarkup`, para aplicar formato a los elementos del índice, que consisten en el título del elemento y el número de página. Estas variables se pueden cambiar por parte del usuario:

```
\paper {
  %% Traducir el título del índice al español:
  tocTitleMarkup = \markup \huge \column {
    \fill-line { \null "Índice general" \null }
    \hspace #1
  }
  %% usar una fuente mayor
  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`.

También se pueden definir nuevas instrucciones y elementos de marcado para crear índices generales más elaborados:

- en primer lugar, defina una variable de marcado nueva dentro del bloque `\paper`
- luego defina una función de música cuyo propósito es añadir un elemento al índice general utilizando esta variable de marcado del bloque `\paper`.

En el ejemplo siguiente se define un estilo nuevo para introducir los nombres de los actos de una ópera en el índice general:

```
\paper {
  tocActMarkup = \markup \large \column {
    \hspace #1
    \fill-line { \null \italic \fromproperty #'toc:text \null }
    \hspace #1
  }
}

tocAct =
#(define-music-function (parser location text) (markup?)
  (add-toc-item! 'tocActMarkup text))
```

## Table of Contents

### *Atto Primo*

Coro. Viva il nostro Alcide	1
Cesare. Presti omai l'Egizzia terra	1

### *Atto Secondo*

Sinfonia	1
Cleopatra. V'adoro, pupille, saette d'Amore	1

## Véase también

Archivos de inicio: ‘ly/toc-init.ly’.

## Instrucciones predefinidas

`\table-of-contents \tocItem`

## 3.3 Trabajar sobre los archivos de entrada

### 3.3.1 Inclusión de archivos de LilyPond

### 3.3.2 Distintas ediciones a partir de una misma fuente

### Uso de las variables

### Uso de etiquetas

### 3.3.3 Codificación del texto

### 3.3.4 Presentar la notación de LilyPond

## 3.4 Controlar la salida

### 3.4.1 Extracción de fragmentos de música

### 3.4.2 Saltar la música corregida

## 3.5 Salida MIDI

El MIDI (Musical Instrument Digital Interface, Interfase Digital para Instrumentos Musicales) es un estándar para interconectar y controlar instrumentos musicales electrónicos. Un archivo o secuencia MIDI es una serie de notas dentro de un conjunto de pistas. No es un archivo de sonidos reales; se necesita un programa reproductor especial para traducir la serie de notas en sonidos de verdad.

Cualquier música se puede convertir a archivos MIDI, de manera que podamos escuchar lo que hayamos introducido. Esto es muy conveniente para comprobar la corrección de la música; las octavas equivocadas o las alteraciones erróneas se ponen de relieve muy claramente al escuchar la salida MIDI.

## Advertencias y problemas conocidos

Muchhos efectos con interés musical como el swing, la articulación, el fraseo, etc., no se traducen al midi.

La salida midi reserva un canal para cada pentagrama, y uno más para los ajustes globales. Por ello, el archivo de entrada no debería tener más de 15 pentagramas (o 14 si no usa percusión). Los otros pentagramas permanecerán en silencio.

No todos los reproductores de midi manejan correctamente los cambios de tempo en la salida midi. Entre los reproductores que se sabe que funcionan, se encuentra **timidity**.

### 3.5.1 Crear archivos MIDI

Para crear un MIDI a partir de una pieza de música, escriba un bloque `\midi` en la partitura, por ejemplo:

```
\score {
  ...música...
  \midi {
    \context {
      \Score
      tempoWholesPerMinute = #(ly:make-moment 72 4)
    }
  }
}
```

El tempo se puede especificar utilizando la instrucción `\tempo` dentro de la propia música, véase **[Indicaciones metronómicas]**, página 101. Más abajo se muestra una alternativa que no da lugar a una indicación metronómica en la partitura impresa. En este ejemplo, el tempo de negras se establece en 72 pulsos por minuto. Esta clase de especificación del tempo no puede tomar notas con puntillo como argumento. En este caso, divida las notas con puntillo en unidades más pequeñas. Por ejemplo, un tempo de 90 negras con puntillo por minuto se puede especificar como 270 corcheas por minuto:

```
tempoWholesPerMinute = #(ly:make-moment 270 8)
```

Si hay una instrucción `\midi` dentro de un `\score`, solamente se producirá MIDI. Si se necesita también una notación impresa, se debe escribir un bloque `\layout`

```
\score {
  ...música...
  \midi { }
  \layout { }
}
```

Se interpretan las ligaduras de unión, los matices dinámicos y los cambios de tempo. Las marcas dinámicas, *crescendi* y *decrescendi* se traducen en niveles de volumen MIDI. Las marcas de dinámica se traducen a una fracción fija del rango de volumen MIDI disponible, los *crescendi* y *decrescendi* hacen que el volumen varíe de forma lineal entre sus dos extremos. Las fracciones se pueden ajustar mediante `dynamicAbsoluteVolumeFunction` dentro de un contexto `Voice`. Para cada tipo de instrumento MIDI se puede definir un rango de volumen. Esto proporciona un control de ecualización básico, que puede realzar significativamente la calidad de la salida

MIDI. El ecualizador se puede controlar estableciendo un valor para `instrumentEqualizer`, o fijando

```
\set Staff.midiMinimumVolume = #0.2
\set Staff.midiMaximumVolume = #0.8
```

Para quitar los matices dinámicos de la salida MIDI, escriba las siguientes líneas dentro de la sección `\midi{}`.

```
\midi {
  ...
  \context {
    \Voice
    \remove "Dynamic_performer"
  }
}
```

## Advertencias y problemas conocidos

Los (de)crescendos sin terminar no se procesarán adecuadamente en el archivo midi, dando como resultado pasajes de música en silencio. Una forma de sortearlo es terminar el (de)crescendo explícitamente. Por ejemplo,

```
{ a\< b c d\f }
```

no funcionará bien, pero

```
{ a\< b c d\!\f }
```

sí lo hará.

Solamente se crea una salida MIDI cuando la instrucción `\midi` está dentro de un bloque `\score`. Si lo escribe dentro de un contexto instanciado explícitamente (es decir: `\new Score`) el archivo no se producirá. Para solucionarlo, encierre el `\new Score` y el `\midi` dentro de un bloque `\score`.

```
\score {
  \new Score { ...notas... }
  \midi
}
```

## Nombres de instrumentos

El nombre del instrumento MIDI se establece mediante la propiedad `Staff.midiInstrument`. El nombre del instrumento se debe elegir de entre los que están en la lista que aparece en [Sección B.2 \[Instrumentos MIDI\], página 241](#).

```
\set Staff.midiInstrument = "glockenspiel"
...notes...
```

Si el instrumento elegido no coincide exactamente con uno de los instrumentos de la lista de instrumentos MIDI, se usará el instrumento Piano de Cola (`"acoustic grand"`).

### 3.5.2 El bloque MIDI

El bloque `\midi` es similar al bloque `\layout`, pero algo más simple. Puede contener definiciones de contexto.

Las definiciones de contexto siguen con exactitud la misma sintaxis que dentro del bloque `\layout`. Los módulos de traducción para el sonido reciben el nombre de «performers» o intérpretes. Los contextos para la salida MIDI se encuentran definidos dentro de `'ly/performer-init.ly'`.

### 3.5.3 ¿Qué sale por el MIDI?

Contemplado en el MIDI

No contemplado en el MIDI

### 3.5.4 Repeticiones y MIDI

Con un poco de trucaje, se puede hacer que cualquier tipo de repetición esté presente en la salida MIDI. Esto se consigue mediante la aplicación de la función musical `\unfoldRepeats`. Esta función cambia todas las repeticiones a repeticiones desplegadas.

```
\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' }
  }
}
\bar "|."
```



Al crear un archivo de partitura que use `\unfoldRepeats` para el MIDI, es necesario hacer dos bloques `\score`: uno para el MIDI (con repeticiones desplegadas) y otro para la notación impresa (con repeticiones de volta –primera y segunda vez–, tremolo –trémolo–, y percent –repetición de compases–). Por ejemplo,

```
\score {
  ..música..
  \layout { .. }
}
\score {
  \unfoldRepeats ..música..
  \midi { .. }
}
```

### 3.5.5 Control de los matices en el MIDI

Indicaciones dinámicas

Volumen maestro MIDI

**Balance entre instrumentos (i)**

**Balance entre instrumentos (i)**

## 4 Problemas de espaciado

### 4.1 El papel y las páginas

#### 4.1.1 Tamaño del papel

#### 4.1.2 Formateo de las páginas

### 4.2 Disposición de la música

#### 4.2.1 Establecer el tamaño del pentagrama

#### 4.2.2 Disposición de la partitura

### 4.3 Saltos

#### 4.3.1 Saltos de línea

#### 4.3.2 Saltos de página

#### 4.3.3 Saltos de página óptimos

#### 4.3.4 Paso de página óptimo

#### 4.3.5 Saltos de página mínimos

#### 4.3.6 Saltos de línea explícitos

#### 4.3.7 Utilizar una voz adicional para los saltos de línea

### 4.4 Espaciado vertical

#### 4.4.1 Espaciado vertical dentro de un sistema

#### 4.4.2 Espaciado vertical entre sistemas

#### 4.4.3 Posicionamiento explícito de los pentagramas y los sistemas

#### 4.4.4 Espaciado vertical en dos fases

#### 4.4.5 Evitar las colisiones verticales

### 4.5 Espaciado horizontal

#### 4.5.1 Panorámica del espaciado horizontal

#### 4.5.2 Área de espaciado nueva

#### 4.5.3 Cambiar el espaciado horizontal

#### 4.5.4 Longitud de la línea



#### 4.5.5 Notación proporcional

### 4.6 Encajar la música en menos páginas

#### 4.6.1 Mostrar el espaciado

#### 4.6.2 Cambiar el espaciado

## 5 Cambiar los valores por omisión

El objetivo del diseño de LilyPond es proporcionar por defecto la más alta calidad de los resultados. A pesar de ello, podría tener que cambiar este resultado por defecto. La disposición sobre el papel se controla a través de un amplio número de proverbiales ‘botones e interruptores’. Este capítulo no relaciona todos y cada uno de los botones. Más bien da una visión de conjunto sobre qué grupos de controles se encuentran disponibles y explica la forma de hallar el botón exacto que se debe utilizar para conseguir un determinado efecto.

Los controles disponibles para los ajustes finos se describen en un documento separado, el Manual de 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 [on-line](#), pero que también va incluido en el paquete de la documentación de LilyPond.

Hay cuatro áreas en las que se pueden cambiar los valores por defecto:

- Notación automática: cambiar la creación automática de los elementos de notación. Por ejemplo, cambiar las reglas de barrado de las figuras.
- Salida: cambiar el aspecto de los objetos individuales. Por ejemplo, cambiar las direcciones de las plicas o la situación exacta de los subíndices.
- Contexto: modificar aspectos de la traducción de los eventos musicales en notación. Por ejemplo, dar a cada pentagrama una indicación de compás distinta.
- Disposición de la página: cambiar el aspecto visual del espaciado, los saltos de línea y las dimensiones de la página. Estas modificaciones se discuten en [Capítulo 3 \[Entrada y salida generales\]](#), página 199 y [Capítulo 4 \[Problemas de espaciado\]](#), página 211.

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 cuadradillo `#`.<sup>1</sup>

### 5.1 Contextos de interpretación

Esta sección explica qué son los contextos y cómo modificarlos.

#### 5.1.1 Explicación de los contextos

Cuando se imprime la música, se tienen que añadir a la salida una gran cantidad de elementos notacionales. Por ejemplo, compare la entrada y la salida del siguiente ejemplo:

```
cis4 cis2. g4
```

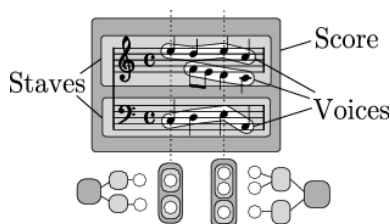


La entrada es bastante escueta, pero en la salida se añaden líneas divisorias, alteraciones accidentales, la clave y la indicación de compás. LilyPond *interpreta* la entrada. En esta fase se inspecciona la información musical en orden temporal, de forma parecida a la lectura de una partitura de izquierda a derecha. Mientras se lee la entrada, el programa recuerda dónde se encuentran los límites de los compases, y qué notas requieren alteraciones explícitas. Esta información se puede presentar sobre varios niveles. Por ejemplo, el efecto de una alteración

<sup>1</sup> Sección “Scheme tutorial” in *Manual de Aprendizaje* contiene un breve tutorial sobre la introducción de números, listas, cadenas y símbolos en Scheme.

accidental se encuentra limitada a un solo pentagrama, mientras que una barra divisoria debe estar sincronizada a través de la partitura de arriba a abajo.

Dentro de LilyPond, estas reglas y pequeñas porciones de información se agrupan en *Contexts*. Algunos ejemplos de contextos son **Voice** (Voz), **Staff** (Pauta o pentagrama) y **Score** (Partitura). Los contextos son jerárquicos, por ejemplo: un **Staff** contener muchas **Voices**, y una **Score** puede contener muchos contextos de **Staff**.



Cada contexto asume la responsabilidad de imponer algunas reglas de notación, creando ciertos objetos de notación y manteniendo las propiedades asociadas. Por ejemplo, el contexto **Voice** puede introducir una alteración accidental y entonces el contexto **Staff** mantiene la regla de mostrar o suprimir la alteración para el resto del compás. La sincronización de las líneas divisorias se gestiona dentro del contexto de la partitura, **Score**.

Sin embargo, en algunas músicas posiblemente no queramos que las líneas divisorias estén sincronizada (pensemos en una partitura polimétrica en compases de 4/4 y de 3/4). En tales casos, debemos modificar los ajustes por omisión de los contextos **Score** y **Staff**.

Para partituras muy sencillas, los contextos se crean implícitamente y no debemos preocuparnos por ellos. Para piezas mayores, como por ejemplo cualquiera que tenga más de un pentagrama, los contextos se deben crear explícitamente para asegurarnos de que tendremos la cantidad exacta de pentagramas que necesitamos, y que están en el orden correcto. Para tipografiar piezas con notación especializada, puede ser útil modificar contextos existentes o definir unos nuevos.

En la referencia del programa se encuentra una descripción completa de todos los contextos que están disponibles, consulte Traducción  $\mapsto$  Contexto.

### 5.1.2 Crear contextos

Para partituras que sólo tienen una voz y un pentagrama, los contextos se crean automáticamente. Para partituras más complejas, es necesario crearlos a mano. Existen tres instrucciones que hacen esto.

- La instrucción más fácil es `\new`, y es también la más rápida de escribir. Se antepone a una expresión musical, por ejemplo

```
\new tipo expresión_musical
```

donde *tipo* es el nombre de un contexto (como **Staff** o **Voice**). Esta instrucción crea un contexto nuevo, y empieza a interpretar la *expresión\_musical* con él.

Una aplicación práctica de `\new` es una partitura con muchos pentagramas. Cada parte que debe ir en su propio pentagrama, va precedida de `\new Staff`.

```
<<
  \new Staff { c4 c }
  \new Staff { d4 d }
>>
```



La instrucción `\new` puede también dar nombre al contexto,

```
\new tipo = identificador música
```

Sin embargo, este nombre especificado por el usuario sólo se utiliza si no hay ya otro contexto anterior con el mismo nombre.

- Como `\new`, la instrucción `\context` también dirige una expresión musical a un objeto de contexto, pero da al contexto un nombre explícito. La sintaxis es

```
\context tipo = identificador música
```

En esta forma, la instrucción buscará un contexto existente del *tipo* especificado que tenga el nombre *identificador*. Si ese contexto aún no existe, se crea un contexto nuevo con el nombre especificado. Esto es útil si nos vamos a referir más tarde al contexto. Por ejemplo, cuando se escribe la letra, la melodía está dentro de un contexto con nombre

```
\context Voice = "tenor" música
```

de forma que los textos se puedan alinear correctamente con sus notas,

```
\new Lyrics \lyricsto "tenor" letra
```

Otro uso posible de los contextos con nombre es la fusión de dos expresiones musicales distintas en un solo contexto. En el siguiente ejemplo, se introducen por separado las articulaciones y las notas,

```
musica = { c4 c4 }
```

```
decoracion = { s4-. s4-> }
```

se combinan enviando los dos al mismo contexto `Voice`,

```
<<
```

```
  \new Staff \context Voice = "A" \musica
```

```
  \context Voice = "A" \decoracion
```

```
>>
```



Con este mecanismo, es posible definir un Urtext (una edición original), con la posibilidad de poner articulaciones distintas sobre las mismas notas.

- La tercera instrucción para crear contextos es

```
\context tipo música
```

Esto es similar a `\context` con `= identificador`, pero se corresponde con cualquier contexto del tipo *tipo*, sin importar qué nombre se le ha dado.

Esta variante se usa con expresiones musicales que se pueden interpretar en varios niveles. Por ejemplo, la instrucción `\applyOutput` (véase [Sección 6.5.2 \[Ejecutar una función sobre todos los objetos de la presentación\]](#), página 238). Sin una instrucción `\context` explícita, normalmente se aplicaría a `Voice`

```
\applyOutput #'contexto #función % aplicar al contexto Voice
```

Para que se interprete dentro de los niveles de `Score` o `Staff`, utilice las siguientes formas:

```
\applyOutput #'Score #función
```

```
\applyOutput #'Staff #función
```

### 5.1.3 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_head_engraver` que crea las cabezas de nota y el contexto `Staff` contiene un grabador `Key_signature_engraver` que crea la indicación de compás.

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 los ... debe 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 {
    f2 g
  }
  \new Staff \with {
    \remove "Time_signature_engraver"
    \remove "Clef_engraver"
  } {
    f2 g2
  }
>>
```

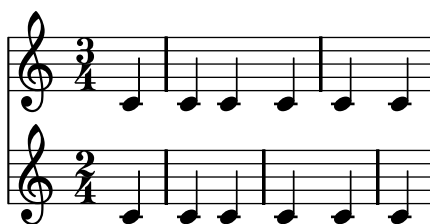


En el segundo pentagrama no hay indicación de compás ni clave. Éste es un método bastante rudimentario de hacer que desaparezcan los objetos porque afecta a todo el pentagrama. Este método también afecta al espaciado, lo que puede ser deseable o no serlo. Se muestran métodos más sofisticados para quitar objetos en [Sección “Visibility and color of objects” in \*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.

```
\new Score \with {
  \remove "Timing_translator"
  \remove "Default_bar_line_engraver"
} <<
  \new Staff \with {
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  } {
    \time 3/4
    c4 c c c c c
  }
  \new Staff \with {
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  } {
    \time 2/4
    c4 c c c c c
  }
}>>
```



#### 5.1.4 Cambiar los valores por omisión de los contextos

Los ajustes de las secciones previas ( [Sección 5.3.2 \[La instrucción set\]](#), página 225, [Sección 5.1.3 \[Modificar los complementos \(plug-ins\) de contexto\]](#), página 216 y [Sección 5.3.1 \[Panorámica de la modificación de las propiedades\]](#), página 223) también se pueden escribir separados de la música dentro del bloque `\layout`:

```
\layout {
  ...
  \context {
    \Staff

    \set fontSize = #-2
    \override Stem #'thickness = #4.0
    \remove "Time_signature_engraver"
  }
}
```

La instrucción `\Staff` recupera la definición existente del contexto de pentagrama de manera que se pueda modificar.

Los enunciados

```
\set fontSize = #-2
\override Stem #'thickness = #4.0
```

```
\remove "Time_signature_engraver"
```

afectan a todos los pentagramas de la partitura. Otros contextos se pueden modificar de forma análoga.

La palabra clave `\set` es opcional dentro del bloque `\layout`, y así

```
\context {
  ...
  fontSize = #-2
}
```

también funciona.

## Advertencias y problemas conocidos

No es posible recolectar cambios de contexto dentro de una variable y aplicarlos a una definición de `\context` por referencia a dicha variable.

La instrucción `\RemoveEmptyStaffContext` sobrescribe nuestros ajustes en curso para `\Staff`. Si queremos cambiar los valores predeterminados para un pentagrama que utilice `\RemoveEmptyStaffContext`, debe hacerlo después de llamar a `\RemoveEmptyStaffContext`, o sea

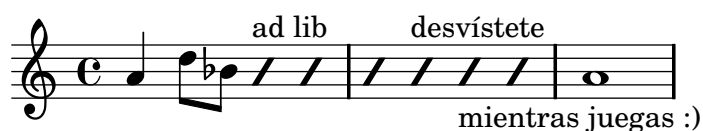
```
\layout {
  \context {
    \RemoveEmptyStaffContext

    \override Stem #'thickness = #4.0
  }
}
```

### 5.1.5 Definir contextos nuevos

Los contextos específicos, como `Staff` y `Voice`, están contruidos a base 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 sobre contextos de `Voice` (existentes) para que siga funcionando. Esto se consigue dando al contexto nuevo un alias `Voice`,

```
\alias Voice
```

El contexto imprimirá notas y textos explicativos, por ello tenemos que añadir los grabadores que aportan esta funcionalidad,

```
\consists Note_heads_engraver
```

```
\consists Text_engraver
```

Pero sólo necesitamos esto en 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
```

```
\override Stem #'transparent = ##t
```

Todos estos añadidos tienen que cooperar, y esto se consigue con un añadido especial, que se debe marcar con la palabra clave `\type`. Este será siempre `Engraver_group`,

```
\type "Engraver_group"
```

Al juntarlo todo, obtenemos

```
\context {
  \name ImproVoice
  \type "Engraver_group"
  \consists "Note_heads_engraver"
  \consists "Text_engraver"
  \consists Pitch_squash_engraver
  squashedPosition = #0
  \override NoteHead #'style = #'slash
  \override Stem #'transparent = ##t
  \alias Voice
}
```

Los contextos dan lugar a jerarquías. Queremos colgar el contexto `ImproVoice` bajo el contexto `Staff`, como simples `Voices` normales. Por tanto, modificamos la definición de `Staff` con la instrucción `\accepts` (acepta),

```
\context {
  \Staff
  \accepts ImproVoice
}
```

Lo opuesto a `\accepts` (acepta) es `\denies` (deniega), lo que a veces se necesita cuando se están reutilizando definiciones de contexto existentes.

Ponemos ambos dentro de un bloque `\layout`, como

```
\layout {
  \context {
    \name ImproVoice
    ...
  }
}
```



```

\context {
  \Staff
  \accepts "ImproVoice"
}
}

```

Así pues, la salida que aparece al comienzo de esta sub-sección se puede escribir como

```

\relative c'' {
  a4 d8 bes8
  \new ImproVoice {
    c4^"ad lib" c
    c4 c^"undress"
    c c_"while playing :)"
  }
  a1
}

```

### 5.1.6 Alinear los contextos

Los contextos nuevos se pueden alinear por encima o por debajo de otros contextos existentes. Esto podría ser de utilidad al preparar un pentagrama vocal ([Sección “Vocal ensembles” in \*Manual de Aprendizaje\*](#)) y un ossia,



## 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
\stemUp
f

```



Si hace una visita a la documentación en busca de instrucciones de digitación (en [\[Indicaciones de digitación\]](#), página 109), encontrará:

### Véase también

Referencia de funcionamiento interno: **Fingering**.

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 **Fingering**. Al principio de la página, puede ver Los objetos de digitación se crean por parte de: **Fingering\_engraver** y **New\_fingering\_engraver**.

Siguiendo los enlaces relacionados dentro de la referencia del programa, podemos seguir el flujo de información dentro del programa:

- **Fingering**: los objetos **Fingering** se crean por parte de: **Fingering\_engraver**
- **Fingering\_engraver**: Tipos de música aceptados: **fingering-event**
- **fingering-event**: El tipo de evento musical **fingering-event** está descrito en Expresiones musicales con el nombre de **FingeringEvent**

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 **Translation**, y del **Backend**. 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 **Fingering**. 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 **grob-interface**.

La página dedicada a **Fingering** relaciona las definiciones del objeto **Fingering**. Por ejemplo, la página dice

**relleno** (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 **Fingering**

Este objeto contempla los siguientes interfaces: `item-interface`, `self-alignment-interface`, `side-position-interface`, `text-interface`, `text-script-interface`, `font-interface`, `finger-interface` y `grob-interface`.

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 3.1.3 \[Estructura del archivo\]](#), página 199) '`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 . fetaNumber)
    (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
\stemUp
f
```



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 3 espacios de pentagrama entre la nota y la digitación:

```
\once \override Voice.Fingering #'padding = #3
```

Al insertar esta instrucción antes de que se haya creado el objeto **Fingering**, es decir, antes del **c2**, llegamos al siguiente resultado:

```
\once \override Voice.Fingering #'padding = #3
```

```
c-2
```

```
\stemUp
```

```
f
```



En este caso, el contexto de este truco es **Voice**. Este hecho se puede deducir también a partir de la referencia del programa, ya que la página dedicada al añadido **Fingering\_engraver** dice

El grabador **Fingering\_engraver** es parte de los contextos: ... **Voice**

## 5.2.4 Convenciones de nombres

## 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.

La sintaxis de esto es

```
\override contexto.nombre #'propiedad = #valor
```

Aquí *nombre* es el nombre de un objeto gráfico, como **Stem** o **NoteHead**, y *propiedad* es una variable interna del sistema de formateo ('propiedad del grob' o 'propiedad de disposición'). Este último es un símbolo, y por ello debe ir precedido de un apóstrofe. La subsección [Sección 5.3 \[Modificar las propiedades\]](#), página 223 explica cómo se deben cumplimentar los conceptos *nombre*, *propiedad* y *valor*. Aquí sólo nos ocuparemos de la funcionalidad des esta instrucción.

La instrucción

```
\override Staff.Stem #'thickness = #4.0
```

hace más gruesas las plicas (el valor predeterminado es 1.3, con el grosor de una línea del pentagrama como unidad). Puesto que la instrucción especifica como contexto a **Staff**, sólo se aplica al pentagrama actual. Otros pentagramas mantienen su aspecto normal. Aquí vemos la instrucción en pleno funcionamiento:

```
c4
\override Staff.Stem #'thickness = #4.0
c4
c4
c4
```



La instrucción `\override` modifica la definición de la plica `Stem` dentro del pentagrama en curso `Staff`. Después de que la instrucción se ha interpretado, todas las plicas se engrosan.

DE manera análoga a `\set`, el argumento *contexto* se puede omitir, ocasionando que se utilice el contexto predeterminado `Voice`. Al añadir `\once` se aplica el cambio durante un solo paso de tiempo.

```
c4
\once \override Stem #'thickness = #4.0
c4
c4
```



El `\override` se debe hacer antes de que el objeto se inicia. Por tanto, al alterar objetos *Spanner* «de extensión» como ligaduras o barras, la instrucción `\override` se debe ejecutar en el momento en que se crea el objeto. En este ejemplo:

```
\override Slur #'thickness = #3.0
c8[( c
\override Beam #'thickness = #0.6
c8 c])
```



la ligadura es más gruesa pero la barra no lo es. Esto es así porque la instrucción para `Beam`, la barra, va después de que la barra se ha iniciado, y por ello no tiene ningún efecto.

De forma análoga a `\unset`, la instrucción `\revert` para un contexto deshace una instrucción `\override`; como con `\unset`, solamente afecta a los ajustes que se hicieron dentro del mismo contexto. En otras palabras, el `\revert` del siguiente ejemplo no hace nada.

```
\override Voice.Stem #'thickness = #4.0
\revert Staff.Stem #'thickness
```

Salgunas opciones «trucables» se llaman ‘subpropiedades’ y residen dentro de las propiedades normales. Para trucarlas, utilice instrucciones de la forma

```
\override context.name #'property #'subproperty = #value
```

tales como

```
\override Stem #'details #'beamed-lengths = #'(4 4 3)
```

## Véase también

Referencia de funcionamiento interno: `OverrideProperty`, `RevertProperty`, `PropertySet`, `Backend` y `All layout objects`.

## 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:

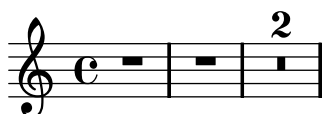
```
\set contexto.propiedad = #valor
```

Por ejemplo:

```
R1*2
```

```
\set Score.skipBars = ##t
```

```
R1*2
```



Esta instrucción salta los compases que no tienen notas. El resultado es que los silencios multicompaés se condensan. El valor asignado es un objeto de Scheme. En este caso, es `#t`, el valor booleano `True` o verdadero.

Si se omite el argumento *contexto*, entonces se utiliza el contexto actual de nivel más bajo (normalmente `ChordNames`, `Voice` o `Lyrics`). En este ejemplo:

```
c8 c c c
```

```
\set autoBeaming = ##f
```

```
c8 c c c
```



el argumento *contexto* de la instrucción `\set` se omite, así pues el barrado automático se desactiva en la voz actual. Observe que el contexto de nivel más bajo no siempre contiene la propiedad que queremos cambiar (por ejemplo, no tendrá ningún efecto intentar establecer un valor para la propiedad `skipBars` del contexto de nivel más bajo, en este caso `Voice`).

```
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. El cambio se aplica ‘al vuelo’, durante la música, de manera que el ajuste sólo afecta al segundo grupo de corcheas.

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*, de manera que

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

introduce un ajuste de la propiedad en el nivel de **Staff**. El ajuste también se aplica a la **Voice** actual. Sin embargo:

```
\unset Voice.autoBeaming
```

no tiene ningún efecto. Para cancelar este ajuste, se debe especificar el `\unset` en el mismo nivel que el `\set` original. Dicho de otra forma, para deshacer el efecto de `Staff.autoBeaming = ##f` se necesita

```
\unset Staff.autoBeaming
```

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.

Los ajustes que se aplican solamente a un único paso de tiempo se pueden escribir con `\once`, por ejemplo en

```
c4
```

```
\once \set fontSize = #4.7
```

```
c4
```

```
c4
```



la propiedad `fontSize` se deshace automáticamente después de la segunda nota.

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.

### 5.3.3 La instrucción `\override`

Las instrucciones que modifican la salida tienen por lo general un aspecto como

```
\override Voice.Stem #'thickness = #3.0
```

Para construir este truco debemos determinar los siguientes datos:

- el contexto: aquí **Voice**.
- el objeto de presentación: aquí **Stem**.
- la propiedad de presentación: aquí **thickness**.
- un valor adecuado: aquí **3.0**.

Ciertas opciones ajustables se denominan ‘subpropiedades’ y residen en el interior de las propiedades normales. Para modificarlas utilice instrucciones de la forma

```
\override Stem #'details #'beamed-lengths = #'(4 4 3)
```

Para muchas propiedades, independientemente del tipo de datos de la propiedad, si se establece el valor de la propiedad a falso (**##f**) se producirá su desactivación, ocasionando que LilyPond ignore por completo dicha propiedad. Esto es especialmente útil para «apagar» propiedades de grobs (objetos gráficos) que de otra manera causarían problemas.

Mostraremos a continuación cómo localizar esta información en el manual de notación y en la referencia de funcionamiento interno.

### 5.3.4 `\set` frente a `\override`

Hemos visto dos formas de cambiar las propiedades: `\set` y `\override`. De hecho, en realidad existen dos clases diferentes de propiedades.

Los contextos pueden tener propiedades, que por lo general reciben nombres en mayúsculasDeCamello. Principalmente controlan la traducción de la música a la notación, p.ej. `localKeySignature` (para determinar si hay que imprimir las alteraciones accidentales), `measurePosition` (para determinar cuándo imprimir una línea divisoria). Las propiedades de contexto pueden ver modificado su valor con el tiempo según se interpreta una pieza de música; `measurePosition` es un ejemplo obvio de esto. Las propiedades de contexto se modifican con `\set`.

Hay un tipo especial de propiedad de contexto: la descripción del elemento. Estas propiedades reciben nombres en MayúsculasDeCamello (comenzando en letra mayúscula). Contienen los ‘ajustes por defecto’ para dicho elemento gráfico como una lista asociativa. Consulte ‘`scm/define-grobs.scm`’ para ver qué tipos de ajustes hay. Las descripciones de los elementos se pueden modificar con `\override`.

Realmente, `\override` es un atajo;

```
\override contexto.nombre #'propiedad = #valor
```

es más o menos equivalente a

```
\set contexto.nombre #'propiedad = #(cons (cons 'propiedad valor) <valor previo de contexto>)
```

El valor de `contexto` (la lista-a) se usa para dar un valor inicial a las propiedades de los objetos gráficos individuales. Los objetos gráficos también tienen propiedades, que reciben nombres en el estilo de Scheme, con palabras-con-guiones. Los valores de las propiedades de objetos gráficos cambian durante el proceso de formateo: el formateo básicamente consiste en calcular las propiedades utilizando funciones de callback.

`fontSize` es una propiedad especial: equivale a escribir `\override ... #'font-size` para todos los objetos pertinentes. Al ser éste un cambio muy común, se creó la propiedad especial (modificada con `\set`).

### 5.3.5 La instrucción `\tweak`

En ocasiones es posible tomar un atajo para realizar el ajuste fino de los objetos gráficos. Para objetos que reusultan directamente de un elemento de código de la entrada, puede usar la función `\tweak`, por ejemplo

```
<
c
\tweak #'color #red d
g
\tweak #'duration-log #1 a
>4-\tweak #'padding #10 -.
```





Como podemos ver, las propiedades se establecen directamente dentro de los objetos, sin hacer mención al nombre del grob ni al contexto en el que se tendría que aplicar.

Esta técnica sólo funciona para los objetos que están conectados directamente a un **Event** (evento) del texto de entrada, por ejemplo

- cabezas de nota, producidas por chord-pitch (p. ej.: las notas de un acorde)
- signos de articulación, producidos por instrucciones de articulación

Posiblemente no funcione para plicas y alteraciones accidentales (éstos están producidos por parte de las cabezas de nota, no por los eventos musicales) ni para las claves (éstos no están producidos por ninguna entrada musical, sino más bien a causa de un cambio en el valor de una propiedad).

Existen muy pocos objetos que estén conectados *directamente* al resultado de salida. Una nota normal (como `c4`) no está conectada directamente a la salida, así que

```
\tweak #'color #red c4
```

no cambia el color. Consulte [Sección 6.3.1 \[Presentación de expresiones musicales\]](#), página 238 para ver más detalles.

## 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.

#### Modo de acordes

Se activa con la instrucción `\chordmode` y produce que la entrada se interprete con la sintaxis de la notación de acordes, véase [Sección 2.7 \[Notación de acordes\]](#), página 163. Los acordes se imprimen como notas sobre un pentagrama.

El modo de acordes se activa también con la instrucción `\chords`. Esto crea también un contexto **ChordNames** nuevo y produce que el código que sigue se interprete con la sintaxis de la notación de acordes y se imprima como nombres de acorde dentro del contexto **ChordNames**, véase [\[Impresión de los nombres de acorde\]](#), página 168.

#### 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 [Sección 2.5.1.2 \[Notación básica de percusión\]](#), página 157.

El modo de percusión también se activa con la instrucción `\drums`. También crea un contexto **DrumStaff** nuevo y hace que el código que sigue se interprete con la sintaxis de la notación de percusión y se imprima como símbolos de percusión sobre un pentagrama de percusión, véase [Sección 2.5.1.2 \[Notación básica de percusión\]](#), página 157.

#### 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 175.

El modo de cifrase también se activa con la instrucción `\figures`. También crea un contexto de **FiguredBass** nuevo y hace que el código que viene a continuación se interprete con la sintaxis

del bajo cifrado y se imprima como símbolos de bajo cifrado dentro del contexto `FiguredBass`, véase [\[Introducción al bajo cifrado\]](#), página 174.

### 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 151.

Para crear diagramas de trastes encima de un pentagrama, escríbalos como elementos de marcado encima de las notas utilizando la instrucción `\fret-diagram`, véase [\[Marcas de diagramas de trastes\]](#), página 153.

### 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 130.

El modo de letra también se habilita con la instrucción `\addlyrics`. Esto también crea un contexto `Lyrics` nuevo y una instrucción `\lyricsto` implícita que asocia la letra que viene a continuación con la música precedente.

### Modo de marcado

Se activa con la instrucción `\markup`, y hace que la entrada se interprete con la sintaxis del marcado, véase [Sección B.6 \[Text markup commands\]](#), página 258.

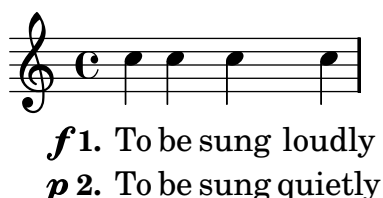
### Modo de notas

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.

Por ejemplo, para insertar indicaciones dinámicas para las estrofas de una pieza coral es necesario entrar en el modo de notas para poder interpretar dichas indicaciones:

```
{ c4 c4 c4 c4 }
\addlyrics {
  \notemode{\set stanza = \markup{ \dynamic f 1. } }
  To be sung loudly
}
\addlyrics {
  \notemode{\set stanza = \markup{ \dynamic p 2. } }
  To be sung quietly
}
```



### 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 lestra, 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.

#### Acciones predeterminadas

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).

#### Disposición de contexto

Los contextos se colocan dentro de un sistema de arriba a abajo en el orden en que se encuentran. Sin embargo, observe que se crea un contexto implícitamente si se encuentra una instrucción cuando no está disponible un contexto apropiado para contenerla.

Se puede cambiar el orden predeterminado en que los contextos se presentan, véase [Sección 5.1.6 \[Alinear los contextos\]](#), página 220

#### Indicadores de dirección de las articulaciones

Al añadir articulaciones a notas se puede omitir normalmente el indicador de dirección, `^` (que significa “arriba”), `_` (que significa “abajo”) o `-` (que significa “usar la dirección predeterminada”), en cuyo caso se supone el predeterminado `-`. Pero se necesita un indicador de dirección **always** 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. `-. , -> , --`

#### 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`, `xxxNeutral`

donde `xxxNeutral` significa “utilizar la dirección predeterminada”. Véase [Sección “Within-staff objects” in \*Manual de Aprendizaje\*](#).

En alguna que otra ocasión, con el único ejemplo usual del arpeggio, el valor de la propiedad `direction` especifica 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”, como antes.

### 5.4.3 Distancias y medidas

### 5.4.4 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.

## 5.5 Propiedades más usuales

### 5.5.1 Control de la visibilidad de los objetos

### 5.5.2 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-interface::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.

```
d2 \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
```



La información que determina los puntos extremos se calcula al vuelo para cada objeto gráfico, pero es posible sobrescribirlos.

```
e2 \glissando f
\once \override Glissando #'bound-details #'right #'Y = #-2
e2 \glissando f
```



El objeto *Glissando*, como cualquier otro que use la rutina `ly:line-interface::print`, conlleva una lista asociativa anidada. En la instrucción anterior, el valor de *Y* se establece a `-2` para la lista asociativa correspondiente al extremo final de la derecha. Por supuesto, también es posible ajustar el extremo de la izquierda con `left` en vez de `right`.

Si no se establece el valor de Y, se calcula a partir de la posición vertical del punto de anclaje derecho del objeto de extensión.

En el caso de un salto de línea, los valores de los puntos extremos se extienden con los contenidos de las sublistas `left-broken` y `right-broken`, por ejemplo

```
\override Glissando #'breakable = ##T
\override Glissando #'bound-details #'right-broken #'Y = #-3
c1 \glissando \break
f1
```



Las siguientes propiedades se pueden usar para el ajuste fino de los objetos de extensión:

**Y** Establece la coordenada Y del punto extremo, en espacios de pentagrama. Por omisión, es el centro del objeto con que enlaza. Así, para un glissando apuntaría al centro vertical de la cabeza de la nota.

Para objetos de extensión horizontales, como el de texto y los trinos, tiene el valor fijo de cero.

**attach-dir**

Determina el lugar donde la línea comienza y acaba en la dirección X, relativo al objeto al que se enlaza. Así, un valor de `-1` (o `LEFT`) hace que la línea comience o acabe en el lado izquierdo de la cabeza de la nota a la que el objeto se encuentra adjuntado.

**X** Es la coordenada absoluta del extremo final. Normalmente se calcula al vuelo y no tiene mucho sentido sobreescribirlo.

**stencil** Los objetos extensores de línea pueden tener símbolos al principio o al final, que se encuentra contenido dentro de esta subpropiedad. Es para uso interno y se recomienda usar `text`.

**text** Es un elemento de marcado que se evalúa para generar la propiedad `stencil` anterior. Se usa para poner *cresc.* y *tr* sobre los objetos de extensión horizontales.

```
\override TextSpanner #'bound-details #'left #'text
= \markup { \small \bold Slower }
c2\startTextSpan b c a\stopTextSpan
```



**stencil-align-dir-y**

**stencil-offset**

Sin un valor establecido para esto, el stencil se coloca simplemente sobre el extremo del final, tal y como se encuentra definido por las subpropiedades **X** e **Y**. La asignación de un valor para **stencil-align-dir-y** o para **stencil-offset** desplaza el símbolo del extremo de forma relativa al extremo de la línea

```

\override TextSpanner #'bound-details #'left #'stencil-align-dir-y = #DOWN
\override TextSpanner #'bound-details #'right #'stencil-align-dir-y = #UP

\override TextSpanner #'bound-details #'left #'text = #"gggg"
\override TextSpanner #'bound-details #'right #'text = #"hhhh"
c4~\startTextSpan c c c \stopTextSpan

```



**arrow (flecha)**

El establecimiento de esta subpropiedad al valor de **#t** produce una cabeza de flecha al final de la línea.

**padding (relleno)**

Esta subpropiedad controla el espacio entre el punto extremo de la línea especificado y el extremo real. Sin relleno, un glissando empezaría y acabaría en el centro de las cabezas de las notas.

La función musical `\endSpanners` termina los objetos de extensión y los reguladores después de una nota exactamente.

```

\endSpanners
c2 \startTextSpan c2
c2 \< c2

```



Al utilizar `\endSpanners` no es necesario cerrar `\startTextSpan` con `\stopTextSpan`, ni tampoco cerrar los reguladores con `\!`.

## Véase también

Referencia del programa: `TextSpanner`, `Glissando`, `VoiceFollower`, `TrillSpanner`, `line-spanner-interface`.

### 5.5.3 Rotación de objetos

### 5.5.4 Alineación de objetos

## 5.6 Trucos avanzados

### 5.6.1 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`.

### **5.6.2 Modificación de los extremos de los extensores**

### **5.6.3 Modificación de los sellos**

## 6 Interfaces para programadores

### 6.1 Funciones musicales

#### 6.1.1 Panorámica de las funciones musicales

#### 6.1.2 Funciones de sustitución sencillas

#### 6.1.3 Funciones de sustitución en parejas

#### 6.1.4 Matemáticas dentro de las funciones

#### 6.1.5 Funciones vacías

#### 6.1.6 Funciones sin argumentos

#### 6.1.7 Panorámica de las funciones musicales disponibles

`acciaccatura` - *music* (music)  
(undocumented; fixme)

`addChordShape` - *key-symbol* (symbol) *shape-string* (string)  
(undocumented; fixme)

`addInstrumentDefinition` - *name* (string) *lst* (list)  
(undocumented; fixme)

`addQuote` - *name* (string) *music* (music)  
(undocumented; fixme)

`afterGrace` - *main* (music) *grace* (music)  
(undocumented; fixme)

`allowPageTurn`  
(undocumented; fixme)

`applyContext` - *proc* (procedure)  
(undocumented; fixme)

`applyMusic` - *func* (procedure) *music* (music)  
(undocumented; fixme)

`applyOutput` - *ctx* (symbol) *proc* (procedure)  
(undocumented; fixme)

`appoggiatura` - *music* (music)  
(undocumented; fixme)

`assertBeamQuant` - *l* (pair) *r* (pair)  
(undocumented; fixme)

`assertBeamSlope` - *comp* (procedure)  
(undocumented; fixme)

`autochange` - *music* (music)  
(undocumented; fixme)

`balloonGrobText` - *grob-name* (symbol) *offset* (pair of numbers) *text* (markup)  
(undocumented; fixme)



**balloonText** - *offset* (pair of numbers) *text* (markup)  
 (undocumented; fixme)

**bar** - *type* (string)  
 (undocumented; fixme)

**barNumberCheck** - *n* (integer)  
 (undocumented; fixme)

**bendAfter** - *delta* (unknown)  
 (undocumented; fixme)

**breathe** (undocumented; fixme)

**clef** - *type* (string)  
 (undocumented; fixme)

**cueDuring** - *what* (string) *dir* (direction) *main-music* (music)  
 (undocumented; fixme)

**displayLilyMusic** - *music* (music)  
 (undocumented; fixme)

**displayMusic** - *music* (music)  
 (undocumented; fixme)

**endSpanners** - *music* (music)  
 (undocumented; fixme)

**featherDurations** - *factor* (moment) *argument* (music)  
 (undocumented; fixme)

**grace** - *music* (music)  
 (undocumented; fixme)

**includePageLayoutFile**  
 (undocumented; fixme)

**instrumentSwitch** - *name* (string)  
 (undocumented; fixme)

**keepWithTag** - *tag* (symbol) *music* (music)  
 (undocumented; fixme)

**killCues** - *music* (music)  
 (undocumented; fixme)

**label** - *label* (symbol)  
 (undocumented; fixme)

**makeClusters** - *arg* (music)  
 (undocumented; fixme)

**musicMap** - *proc* (procedure) *mus* (music)  
 (undocumented; fixme)

**noPageBreak**  
 (undocumented; fixme)

**noPageTurn**  
 (undocumented; fixme)

**octaveCheck** - *pitch-note* (music)  
 (undocumented; fixme)

`oldaddyrics` - *music* (music) *lyrics* (music)  
 (undocumented; fixme)  
`ottava` - *octave* (number)  
 (undocumented; fixme)  
`overrideProperty` - *name* (string) *property* (symbol) *value* (any type)  
 (undocumented; fixme)  
`pageBreak`  
 (undocumented; fixme)  
`pageTurn` (undocumented; fixme)  
`parallelMusic` - *voice-ids* (list) *music* (music)  
 (undocumented; fixme)  
`parenthesize` - *arg* (music)  
 (undocumented; fixme)  
`partcombine` - *part1* (music) *part2* (music)  
 (undocumented; fixme)  
`pitchedTrill` - *main-note* (music) *secondary-note* (music)  
 (undocumented; fixme)  
`pointAndClickOff`  
 (undocumented; fixme)  
`pointAndClickOn`  
 (undocumented; fixme)  
`quoteDuring` - *what* (string) *main-music* (music)  
 (undocumented; fixme)  
`removeWithTag` - *tag* (symbol) *music* (music)  
 (undocumented; fixme)  
`resetRelativeOctave` - *reference-note* (music)  
 (undocumented; fixme)  
`rightHandFinger` - *finger* (number or string)  
 (undocumented; fixme)  
`scaleDurations` - *fraction* (pair of numbers) *music* (music)  
 (undocumented; fixme)  
`scoreTweak` - *name* (string)  
 (undocumented; fixme)  
`shiftDurations` - *dur* (integer) *dots* (integer) *arg* (music)  
 (undocumented; fixme)  
`spacingTweaks` - *parameters* (list)  
 (undocumented; fixme)  
`storePredefinedDiagram` - *chord* (music) *tuning* (list) *terse-definition* (string)  
 (undocumented; fixme)  
`tag` - *tag* (symbol) *arg* (music)  
 (undocumented; fixme)  
`tocItem` - *text* (markup)  
 Add a line to the table of content, using the `tocItemMarkup` paper variable markup

`transposedCueDuring` - *what* (string) *dir* (direction) *pitch-note* (music) *main-music* (music)  
(undocumented; fixme)

`transposition` - *pitch-note* (music)  
(undocumented; fixme)

`tweak` - *sym* (symbol) *val* (any type) *arg* (music)  
(undocumented; fixme)

`unfoldRepeats` - *music* (music)  
(undocumented; fixme)

`withMusicProperty` - *sym* (symbol) *val* (any type) *music* (music)  
(undocumented; fixme)

## 6.2 Interfaces para el programador

### 6.2.1 Variables de entrada y Scheme

### 6.2.2 Representación interna de la música

## 6.3 Construcción de funciones complejas

### 6.3.1 Presentación de expresiones musicales

### 6.3.2 Propiedades de la música

### 6.3.3 Doblar una nota con ligaduras (ejemplo)

### 6.3.4 Añadir articulación a las notas (ejemplo)

## 6.4 Interfaz de marcado para el programador

### 6.4.1 Construcción del marcado en Scheme

### 6.4.2 Cómo funciona internamente el marcado

### 6.4.3 Definición de una instrucción de marcado nueva

### 6.4.4 Definición de nuevas instrucciones de lista de marcado

## 6.5 Contextos para programadores

### 6.5.1 Evaluación de contextos

### 6.5.2 Ejecutar una función sobre todos los objetos de la presentación

## 6.6 Procedimientos de Scheme como propiedades

## 6.7 HACER trasladado al apartado sobre scheme

### 6.7.1 Usar código de Scheme en lugar de `\tweak`

### 6.7.2 Trucos difíciles

## Apéndice A Lista bibliográfica

Si tiene necesidad de aprender más acerca de la notación musical, le presentamos a continuación algunos títulos interesantes que puede leer.

### *Ignatzek 1995*

Klaus Ignatzek, *Die Jazzmethode für Klavier*. Schott's Söhne 1995. Mainz, Germany ISBN 3-7957-5140-3.

Instructiva introducción a la interpretación de Jazz al piano. Uno de los primeros capítulos contiene una panorámica de los acordes más comunes de la música de Jazz.

### *Gerou 1996*

Tom Gerou and Linda Lusk, *Essential Dictionary of Music Notation*. Alfred Publishing, Van Nuys CA ISBN 0-88284-768-6.

Una lista concisa y ordenada alfabéticamente de los problemas de la composición tipográfica y la notación musical, que abarca la mayor parte de los casos más comunes.

### *Read 1968*

Gardner Read, *Music Notation: A Manual of Modern Practice*. Taplinger Publishing, New York (2nd edition).

Una obra estándar sobre notación musical.

### *Ross 1987*

Ted Ross, *Teach yourself the art of music engraving and processing*. Hansen House, Miami, Florida 1987.

Este libro trata del grabado musical, es decir, composición tipográfica profesional. Contiene instrucciones sobre el estampado, la utilización de las plumillas y las convenciones notacionales. También son interesantes las secciones sobre los tecnicismos y la historia de la reproducción.

### *Schirmer 2001*

The G.Schirmer/AMP Manual of Style and Usage. G.Schirmer/AMP, NY, 2001. (Este libro se puede pedir al departamento de alquiler.)

Este manual se centra específicamente en la preparación de los manuscritos para la publicación por Schirmer. Discute muchos detalles que no se pueden encontrar en otros libros de notación más normales. También proporciona una buena idea sobre lo que se necesita para llevar la impresión hasta la calidad editorial.

### *Stone 1980*

Kurt Stone, *Music Notation in the Twentieth Century*. Norton, New York 1980.

Este libro describe la notación musical para la música seria moderna, pero empieza por una amplia panorámica de las prácticas existentes de la notación tradicional.

El archivo de código fuente incluye una bibliografía de BibT<sub>E</sub>X más elaborada con más de 100 elementos, en 'Documentation/bibliography/'.

## Apéndice B Tablas del manual sobre notación

### B.1 Carta de nombres de acordes

Las siguientes tablas muestran dos sistemas estándar para imprimir los nombres de los acordes, junto a las notas que representan.

Ignatzek (default)	C	Cm	C+	C <sup>o</sup>	
Alternative	C	C <sup>b3</sup>	C <sup>#5</sup>	C <sup>b3 b5</sup>	
Def	C <sup>7</sup>	Cm <sup>7</sup>	C <sup>Δ</sup>	C <sup>o7</sup>	Cm <sup>Δ/b5</sup>
Alt <sub>5</sub>	C <sup>7</sup>	C <sup>7 b3</sup>	C <sup>#7</sup>	C <sup>b3 b5 b7</sup>	C <sup>b3 b5 #7</sup>
Def	C <sup>7/#5</sup>	Cm <sup>Δ</sup>	C <sup>Δ/#5</sup>	C <sup>ø</sup>	
Alt <sub>6</sub>	C <sup>7 #5</sup>	C <sup>b3 #7</sup>	C <sup>#5 #7</sup>	C <sup>7 b3 b5</sup>	
Def	C <sup>6</sup>	Cm <sup>6</sup>	C <sup>9</sup>	Cm <sup>9</sup>	
Alt <sub>4</sub>	C <sup>6</sup>	C <sup>b3 6</sup>	C <sup>9</sup>	C <sup>9 b3</sup>	
Def	Cm <sup>13</sup>	Cm <sup>11</sup>	Cm <sup>7/b5/9</sup>	C <sup>7/b9</sup>	
Alt <sub>8</sub>	C <sup>13 b3</sup>	C <sup>11 b3</sup>	C <sup>9 b3 b5</sup>	C <sup>7 b9</sup>	
Def	C <sup>7/#9</sup>	C <sup>11</sup>	C <sup>7/#11</sup>	C <sup>13</sup>	
Alt <sub>22</sub>	C <sup>7 #9</sup>	C <sup>11</sup>	C <sup>9 #11</sup>	C <sup>13</sup>	

Def	$C^{7/\#11/b13}$	$C^{7/\#5/\#9}$	$C^{7/\#9/\#11}$	$C^{7/b13}$
Alt	$C^9 \#11 \flat13$	$C^7 \#5 \#9$	$C^7 \#9 \#11$	$C^{11 \flat13}$
26				
Def	$C^{7/b9/b13}$	$C^{7/\#11}$	$C^{\triangle/9}$	$C^{7/b13}$
Alt	$C^{11 \flat9 \flat13}$	$C^9 \#11$	$C^9 \#7$	$C^{11 \flat13}$
30				
Def	$C^{7/b9/b13}$	$C^{7/b9/13}$	$C^{\triangle/9}$	$C^{\triangle/13}$
Alt	$C^{11 \flat9 \flat13}$	$C^{13 \flat9}$	$C^9 \#7$	$C^{13 \#7}$
34				
Def	$C^{\triangle/\#11}$	$C^{7/b9/13}$	$C^{sus4}$	$C^{7/sus4}$
Alt	$C^9 \#7 \#11$	$C^{13 \flat9}$	$C^{add4 \ 5}$	$C^{add4 \ 5 \ 7}$
38				
Def	$C^{9/sus4}$	$C^{add9}$	$C^{m \ add11}$	
Alt	$C^{add4 \ 5 \ 7 \ 9}$	$C^{add9}$	$C^{\flat3 \ add11}$	
42				

## B.2 Instrumentos MIDI

A continuación figura una lista con los nombres que se pueden utilizar para la propiedad `midiInstrument` (instrumento 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 strings	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)	

## B.3 Lista de colores

### Colores normales

La sintaxis para el uso de colores se encuentra detallada en [\[Colorear objetos\]](#), página 111.

black	white	red	green
blue	cyan	magenta	yellow
grey	darkred	darkgreen	darkblue
darkcyan	darkmagenta	darkyellow	

### Nombres de los colores de X

Los nombres de los colores de X admiten algunas 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 y sin mayúsculas (p.ej. ‘light slate blue’).

La palabra ‘grey’ siempre se puede escribir como ‘gray’ (p.ej. ‘DarkSlateGray’).

Algunos nombres admiten un sufijo numérico (p.ej. ‘LightSalmon4’).

### Nombres de los colores sin sufijo numérico:

snow	GhostWhite	WhiteSmoke	gainsboro	FloralWhite
------	------------	------------	-----------	-------------

OldLace	linen	AntiqueWhite	PapayaWhip	BlanchedAlmond
bisque	PeachPuff	NavajoWhite	moccasin	cornsilk
ivory	LemonChiffon	seashell	honeydew	MintCream
azure	AliceBlue	lavender	LavenderBlush	MistyRose
white	black	DarkSlateGrey	DimGrey	SlateGrey
LightSlateGrey	grey	LightGrey	MidnightBlue	navy
NavyBlue	CornflowerBlue	DarkSlateBlue	SlateBlue	MediumSlateBlue
LightSlateBlue	MediumBlue	RoyalBlue	blue	DodgerBlue
DeepSkyBlue	SkyBlue	LightSkyBlue	SteelBlue	LightSteelBlue
LightBlue	PowderBlue	PaleTurquoise	DarkTurquoise	MediumTurquoise
turquoise	cyan	LightCyan	CadetBlue	MediumAquamarine
aquamarine	DarkGreen	DarkOliveGreen	DarkSeaGreen	SeaGreen
MediumSeaGreen	LightSeaGreen	PaleGreen	SpringGreen	LawnGreen
green	chartreuse	MediumSpringGreen	GreenYellow	LimeGreen
YellowGreen	ForestGreen	OliveDrab	DarkKhaki	khaki
PaleGoldenrod	LightGoldenrodYellow	LightYellow	yellow	gold
LightGoldenrod	goldenrod	DarkGoldenrod	RosyBrown	IndianRed
SaddleBrown	sienna	peru	burlywood	beige
wheat	SandyBrown	tan	chocolate	firebrick
brown	DarkSalmon	salmon	LightSalmon	orange
DarkOrange	coral	LightCoral	tomato	OrangeRed
red	HotPink	DeepPink	pink	LightPink
PaleVioletRed	maroon	MediumVioletRed	VioletRed	magenta
violet	plum	orchid	MediumOrchid	DarkOrchid
DarkViolet	BlueViolet	purple	MediumPurple	thistle
DarkGrey	DarkBlue	DarkCyan	DarkMagenta	DarkRed
LightGreen				

## Nombres de los colores con sufijo numérico

En los nombres siguientes, el sufijo N puede ser un número del 1 al 4:

snowN	seashellN	AntiqueWhiteN	bisqueN	PeachPuffN
NavajoWhiteN	LemonChiffonN	cornsilkN	ivoryN	honeydewN
LavenderBlushN	MistyRoseN	azureN	SlateBlueN	RoyalBlueN
blueN	DodgerBlueN	SteelBlueN	DeepSkyBlueN	SkyBlueN
LightSkyBlueN	LightSteelBlueN	LightBlueN	LightCyanN	PaleTurquoiseN
CadetBlueN	turquoiseN	cyanN	aquamarineN	DarkSeaGreenN
SeaGreenN	PaleGreenN	SpringGreenN	greenN	chartreuseN
OliveDrabN	DarkOliveGreenN	khakiN	LightGoldenrodN	LightYellowN
yellowN	goldN	goldenrodN	DarkGoldenrodN	RosyBrownN
IndianRedN	siennaN	burlywoodN	wheatN	tanN
chocolateN	firebrickN	brownN	salmonN	LightSalmonN
orangeN	DarkOrangeN	coralN	tomatoN	OrangeRedN
redN	DeepPinkN	HotPinkN	pinkN	LightPinkN
PaleVioletRedN	maroonN	VioletRedN	magentaN	orchidN
plumN	MediumOrchidN	DarkOrchidN	purpleN	MediumPurpleN
thistleN				

## Escala de grises

Se puede obtener una escala de grises utilizando:

greyN

donde N es un número entre 0 y 100.































































## B.4 La tipografía Feta

Los símbolos siguientes están disponibles en la tipografía Emmentaler y es posible acceder a ellos directamente utilizando elementos de marcado textual como `g^\markup { \musicglyph # "scripts.segno" }`, véase [Sección 1.8.2 \[Formatear el texto\]](#), página 125.

space		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>	f	<b><i>f</i></b>
m	<b><i>m</i></b>	p	<b><i>p</i></b>
r	<b><i>r</i></b>	s	<b><i>s</i></b>
z	<b><i>z</i></b>	rests.0	<b>—</b>
rests.1	<b>—</b>	rests.0o	<b>—</b>
rests.1o	<b>—</b>	rests.M3	<b>  </b>































rests.M2	█	rests.M1	■
rests.2	↯	rests.2classical	↯
rests.3	↯	rests.4	↯
rests.5	↯	rests.6	↯
rests.7	↯	accidentals.sharp	#
accidentals.sharp .slashslash.stem	♯	accidentals.sharp .slashslashslash.stemstem	♯
accidentals.sharp .slashslashslash.stem	♯	accidentals.sharp .slashslash.stemstemstem	♯
accidentals.natural	♮	accidentals.flat	♭
accidentals.flat.slash	♭	accidentals.flat .slashslash	♭
accidentals .mirroredflat.flat	♭	accidentals.mirroredflat	♭
accidentals .mirroredflat.backslash	♭	accidentals.flatflat	♭
accidentals .flatflat.slash	♭	accidentals.doublsharp	×
accidentals.rightparen	)	accidentals.leftparen	(
arrowheads.open.01	>	arrowheads.open.0M1	<
arrowheads.open.11	^	arrowheads.open.1M1	^

<code>arrowheads.close.01</code>		<code>arrowheads.close.0M1</code>	
<code>arrowheads.close.11</code>		<code>arrowheads.close.1M1</code>	
<code>dots.dot</code>		<code>noteheads.uM2</code>	
<code>noteheads.dM2</code>		<code>noteheads.sM1</code>	
<code>noteheads.s0</code>		<code>noteheads.s1</code>	
<code>noteheads.s2</code>		<code>noteheads.s0diamond</code>	
<code>noteheads.s1diamond</code>		<code>noteheads.s2diamond</code>	
<code>noteheads.s0triangle</code>		<code>noteheads.d1triangle</code>	
<code>noteheads.ultriangle</code>		<code>noteheads.u2triangle</code>	
<code>noteheads.d2triangle</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.s0do</code>		<code>noteheads.d1do</code>	
<code>noteheads.uldo</code>		<code>noteheads.d2do</code>	

<code>noteheads.u2do</code>		<code>noteheads.s0re</code>	
<code>noteheads.u1re</code>		<code>noteheads.d1re</code>	
<code>noteheads.u2re</code>		<code>noteheads.d2re</code>	
<code>noteheads.s0mi</code>		<code>noteheads.s1mi</code>	
<code>noteheads.s2mi</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.s0la</code>	
<code>noteheads.s1la</code>		<code>noteheads.s2la</code>	
<code>noteheads.s0ti</code>		<code>noteheads.ulti</code>	
<code>noteheads.d1ti</code>		<code>noteheads.u2ti</code>	
<code>noteheads.d2ti</code>		<code>scripts.ufermata</code>	
<code>scripts.dfermata</code>		<code>scripts.ushortfermata</code>	
<code>scripts.dshortfermata</code>		<code>scripts.ulongfermata</code>	
<code>scripts.dlongfermata</code>		<code>scripts.uverylongfermata</code>	

scripts.dverylongfermata		scripts.thumb	ø
scripts.sforzato	>	scripts.espr	<>
scripts.staccato	.	scripts.ustaccatissimo	!
scripts.dstaccatissimo	!	scripts.tenuto	—
scripts.upartato	÷	scripts.dpartato	÷
scripts.umarcato	^	scripts.dmarcato	v
scripts.open	o	scripts.stopped	+
scripts.upbow	V	scripts.downbow	⌊
scripts.reverseturn	∞	scripts.turn	∞
scripts.trill		scripts.upedalheel	u
scripts.dpedalheel	n	scripts.upedaltoe	v
scripts.dpedaltoe	^	scripts.flageolet	o
scripts.segno		scripts.coda	
scripts.varcoda		scripts.rcomma	,
scripts.lcomma	,	scripts.rvarcomma	/































<code>scripts.lvarcomma</code>	/	<code>scripts.arpeggio</code>	↗
<code>scripts.trill_element</code>	~	<code>scripts.arpeggio</code> <code>.arrow.M1</code>	↘
<code>scripts.arpeggio.arrow.1</code>	↗	<code>scripts.trilelement</code>	↘
<code>scripts.prall</code>	~	<code>scripts.mordent</code>	~
<code>scripts.prallprall</code>	~	<code>scripts.prallmordent</code>	~
<code>scripts.upprall</code>	~	<code>scripts.upmordent</code>	~
<code>scripts.pralldown</code>	~	<code>scripts.downprall</code>	~
<code>scripts.downmordent</code>	~	<code>scripts.prallup</code>	~
<code>scripts.lineprall</code>	~	<code>scripts.caesura.curved</code>	//
<code>scripts.caesura.straight</code>	//	<code>flags.u3</code>	)
<code>flags.u4</code>	~	<code>flags.u5</code>	~
<code>flags.u6</code>	~	<code>flags.d3</code>	)
<code>flags.ugrace</code>	/	<code>flags.dgrace</code>	/
<code>flags.d4</code>	~	<code>flags.d5</code>	~































flags.d6		clefs.C	
clefs.C_change		clefs.F	
clefs.F_change		clefs.G	
clefs.G_change		clefs.percussion	
clefs.percussion_change		clefs.tab	
clefs.tab_change		timesig.C44	
timesig.C22		pedal.*	
pedal.M		pedal..	
pedal.P		pedal.d	
pedal.e		pedal.Ped	
brackettips.up		brackettips.down	
accordion.accDiscant		accordion.accDot	
accordion.accFreebase		accordion.accStdbase	
accordion.accBayanbase		accordion.accOldEE	
rests.M3neomensural		rests.M2neomensural	

rests.M1neomensural	┆	rests.0neomensural	▪
rests.1neomensural	▪	rests.2neomensural	↗
rests.3neomensural	↖	rests.4neomensural	↗
rests.M3mensural	┆	rests.M2mensural	┆
rests.M1mensural	┆	rests.0mensural	‚
rests.1mensural	‚	rests.2mensural	↗
rests.3mensural	↖	rests.4mensural	↗
noteheads.s1neomensural	≡	noteheads.sM3neomensural	≡
noteheads.sM2neomensural	≡	noteheads.sM1neomensural	≡
noteheads.s0harmonic	◊	noteheads.s2harmonic	◆
noteheads.s0neomensural	◊	noteheads.s1neomensural	◊
noteheads.s2neomensural	◆	noteheads.s1mensural	≡
noteheads.sM3mensural	≡	noteheads.sM2mensural	≡
noteheads.sM1mensural	≡	noteheads.s0mensural	◊
noteheads.s1mensural	◊	noteheads.s2mensural	◆

































noteheads.s0petrucci	◊	noteheads.s1petrucci	◊
noteheads.s2petrucci	◆	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	▮	noteheads.ssolesmes .incl.parvum	▪
noteheads .ssolesmes.auct.asc	▮	noteheads .ssolesmes.auct.desc	▮
noteheads.ssolesmes .incl.auctum	▮	noteheads .ssolesmes.stropha	▮
noteheads.ssolesmes .stropha.aucta	▮	noteheads .ssolesmes.oriscus	▮
noteheads.smedicaea .inclinatum	◆	noteheads .smedicaea.punctum	▪

noteheads .smedicaea.rvirga		noteheads .smedicaea.virga	
noteheads .shufnagel.punctum		noteheads .shufnagel.virga	
noteheads.shufnagel.lpes		clefs.vaticana.do	
clefs.vaticana.do_change		clefs.vaticana.fa	
clefs.vaticana.fa_change		clefs.medicaea.do	
clefs.medicaea.do_change		clefs.medicaea.fa	
clefs.medicaea.fa_change		clefs.neomensural.c	
clefs.neomensural .c_change		clefs.petrucchi.c1	
clefs.petrucchi.c1_change		clefs.petrucchi.c2	
clefs.petrucchi.c2_change		clefs.petrucchi.c3	
clefs.petrucchi.c3_change		clefs.petrucchi.c4	
clefs.petrucchi.c4_change		clefs.petrucchi.c5	
clefs.petrucchi.c5_change		clefs.mensural.c	
clefs.mensural.c_change		clefs.petrucchi.f	
clefs.petrucchi.f_change		clefs.mensural.f	

clefs.mensural.f_change		clefs.petrucchi.g	
clefs.petrucchi.g_change		clefs.mensural.g	
clefs.mensural.g_change		clefs.hufnagel.do	
clefs.hufnagel.do_change		clefs.hufnagel.fa	
clefs.hufnagel.fa_change		clefs.hufnagel.do.fa	
clefs.hufnagel .do.fa_change		custodes.hufnagel.u0	
custodes.hufnagel.u1		custodes.hufnagel.u2	
custodes.hufnagel.d0		custodes.hufnagel.d1	
custodes.hufnagel.d2		custodes.medicaea.u0	
custodes.medicaea.u1		custodes.medicaea.u2	
custodes.medicaea.d0		custodes.medicaea.d1	
custodes.medicaea.d2		custodes.vaticana.u0	
custodes.vaticana.u1		custodes.vaticana.u2	
custodes.vaticana.d0		custodes.vaticana.d1	
custodes.vaticana.d2		custodes.mensural.u0	

custodes.mensural.u1	↗	custodes.mensural.u2	↗
custodes.mensural.d0	↘	custodes.mensural.d1	↘
custodes.mensural.d2	↘	accidentals.medicaeaM1	♭
accidentals.vaticanaM1	♭	accidentals.vaticana0	♯
accidentals.mensural1	✕	accidentals.mensuralM1	♭
accidentals.hufnagelM1	♭	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	}

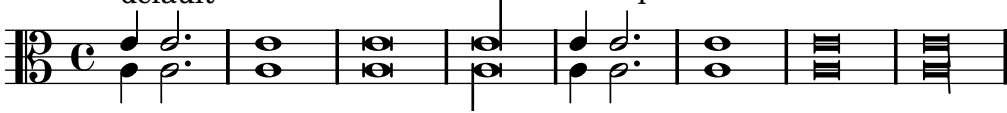
flags.mensuralu16		flags.mensuralu26	
flags.mensurald06		flags.mensurald16	
flags.mensurald26		timesig.mensural44	
timesig.mensural22		timesig.mensural32	
timesig.mensural64		timesig.mensural94	
timesig.mensural34		timesig.mensural68	
timesig.mensural98		timesig.mensural48	
timesig.mensural68alt		timesig.mensural24	
timesig.neomensural44		timesig.neomensural22	
timesig.neomensural32		timesig.neomensural64	
timesig.neomensural94		timesig.neomensural34	
timesig.neomensural68		timesig.neomensural98	
timesig.neomensural48		timesig.neomensural68alt	
timesig.neomensural24		scripts.ictus	
scripts.uaccentus		scripts.daccentus	

<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>	§
<code>dots.dotvaticana</code>	.		

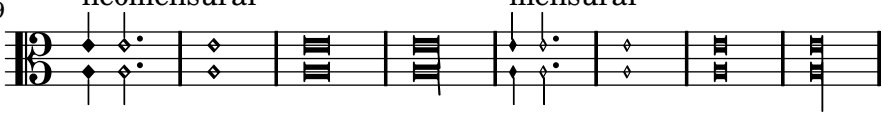
## B.5 Estilos de cabezas de nota

Se pueden usar los siguientes estilos para las cabezas de las notas.


default                      baroque




9                      neomensural                      mensural



17                      petrucci                      harmonic



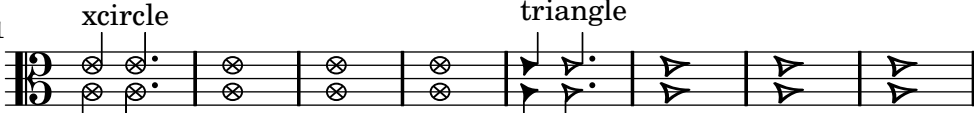
25                      harmonic-black                      harmonic-mixed



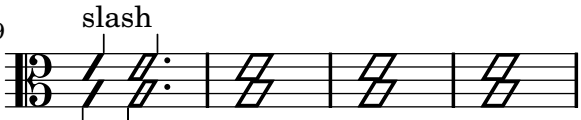
33                      diamond                      cross



41                      xcircle                      triangle



49                      slash



## B.6 Text markup commands

The following commands can all be used inside `\markup { }`.

### B.6.1 Font

`\abs-fontsize size (number) arg (markup)`

Use *size* as the absolute font size to display *arg*. Adjust 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**

`\bigger arg (markup)`

Increase the font size relative to current setting.

```
\markup {
  \huge {
    huge
    \hspace #2
    \bigger {
      bigger
    }
    \hspace #2
    huge
  }
}
```

**huge    bigger    huge**

`\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 the argument 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. Adjust 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)  
Copy of the `\bigger` command.

```
\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, in a normal font size.

```
\markup {
  default
  \normal-size-sub {
    subscript in standard size
  }
}
```

**default**  
subscript in standard size

Used properties:

- `baseline-skip`

`\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:

- `baseline-skip`

`\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 {
    Some text with font overrides
  }
  \hspace #2
  \normal-text {
    Default text, same font-size
  }
  \hspace #2
  More text as before
}
```

**SOME TEXT WITH FONT OVERRIDES**    Default text, same font-size    **MORE**

`\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 only contains numbers and some punctuation. It doesn't have any letters.

```
\markup {
  \number {
    0 1 2 3 4 5 6 7 8 9 . ,
  }
}
```

**0123456789.,**

`\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 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 text` (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 current setting.

```
\markup {
  \fontsize #3.5 {
    some large text
    \hspace #2
    \smaller {
      a bit smaller
    }
    \hspace #2
    more large text
  }
}
```

**some large text    a bit smaller    more large text**

`\sub arg` (markup)

Set *arg* in subscript.

```
\markup {
  \concat {
    H
    \sub {
      2
    }
  }
  0
```

```
}
}
```

**H<sub>2</sub>O**

Used properties:

- `baseline-skip`
- `font-size (0)`

`\super arg` (markup)

Raising and lowering texts can be done with `\super` and `\sub`:

```
\markup {
  E =
  \concat {
    mc
    \super
    2
  }
}
```

**E = mc<sup>2</sup>**

Used properties:

- `baseline-skip`
- `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**

`\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 and y offset.

```
\markup {
  default
  \hspace #2
  \override #'(thickness . 2)
  \underline {
    underline
  }
}
```

**default**    **underline**

Used properties:

- **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*

## B.6.2 Align

`\center-align` *arg* (markup)

Align *arg* to its X center.

```
\markup {
  \column {
    one
    \center-align
    two
    three
  }
}
```

one  
two  
three

`\center-column` *args* (list of markups)

Put *args* in a centered column.

```
\markup {
  \center-column {
    one
    two
    three
  }
}
```

one  
two  
three

Used properties:

- `baseline-skip`

`\column` *args* (list of markups)

Stack the markups in *args* vertically. The property `baseline-skip` determines the space between each markup in *args*.

```
\markup {
  \column {
    one
    two
    three
  }
}
```

one  
two  
three

Used properties:

- `baseline-skip`



`\combine` *m1* (markup) *m2* (markup)

Print two markups on top of each other.

Note: `\combine` cannot take a list of markups enclosed in curly braces as an argument; the follow example will not compile:

```
\combine { a list }
\markup {
  \fontsize #5
  \override #'(thickness . 2)
  \combine
    \draw-line #'(0 . 4)
    \arrow-head #Y #DOWN ##f
}
```



`\concat` *args* (list of markups)

Concatenate *args* in a horizontal line, without spaces inbetween. 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* (list of markups)

Make a column of *args*, going up or down, depending on the setting of the `direction` layout property.

```
\markup {
  \override #'(direction . 1) {
    \dir-column {
      going up
    }
  }
  \dir-column {
    going down
  }
}
```

**up**  
going going  
down

Used properties:

- `baseline-skip`
- `direction`

`\fill-line` *markups* (list of markups)

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 }
    }
  }
}
```

Words evenly spaced across the page

Text markups *evenly spaced* across the page

Used properties:

- `line-width` (#f)
- `word-space` (1)
- `text-direction` (1)

`\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

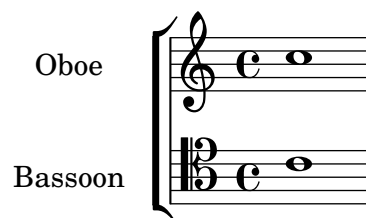
one  
two  
three

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)

This produces an invisible object taking horizontal space. For example,

```
\markup { A \hspace #2.0 B }
```

puts extra space between A and B, on top of the space that is normally inserted before elements on a line.

```
\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"
  descr = "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:descr
    }
  }
}
```

```
\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` *args* (list of markups)

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 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

`\left-align` *arg* (markup)

Align *arg* on its left edge.

```
\markup {
  \column {
    one
    \left-align
    two
```

```

        three
    }
}

```

```

    one
    two
    three

```

`\left-column` *args* (list of markups)  
Put *args* in a left-aligned column.

```

\markup {
  \left-column {
    one
    two
    three
  }
}

```

```

    one
    two
    three

```

Used properties:

- `baseline-skip`

`\line` *args* (list of markups)  
Put *args* in a horizontal line. The property `word-space` determines the space between each markup 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

```

`\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` *padding* (number) *arg* (markup)  
Add space around a markup object.

```
\markup {
  \box {
    default
  }
  \hspace #2
  \box {
    \pad-around #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* (list of markups)

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)

This translates an object. Its first argument is a cons of numbers.

A `\translate #(cons 2 -3) { B C } D`

This moves ‘B C’ 2 spaces to the right, and 3 down, relative to its surroundings.

This command cannot be used to move isolated scripts vertically, for the same reason that `\raise` cannot be used for that.

```
\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
  }
}

```

$\ast$  **translate**  $\ast$  **translate-scaled**

Used properties:

- font-size (0)

`\vcenter` *arg* (markup)  
Align *arg* to its Y center.

```

\markup {
  one
  \vcenter
  two
  three
}

```

one two three

`\wordwrap-field` *symbol* (symbol)  
Wordwrap the data which has been assigned to *symbol*.

```

\header {
  title = "My title"
  descr = "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:descr
    }
  }
}

```

```

\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* (list of markups)

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`

### B.6.3 Graphic

`\arrow-head` *axis* (integer) *direction* (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
    }
  }
}

```

▲Y ><

`\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
  }
}

```

}

[J.]

`\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) *fill* (boolean)

A circle of radius *radius*, thickness *thickness* and optionally filled.

```
\markup {
  \draw-circle #2 #0.5 ##f
  \hspace #2
  \draw-circle #2 #0 ##t
}
```



`\draw-line dest` (pair of numbers)

A simple line.

```
\markup {
  \draw-line #'(4 . 4)
  \override #'(thickness . 5)
  \draw-line #'(-3 . 0)
}
```



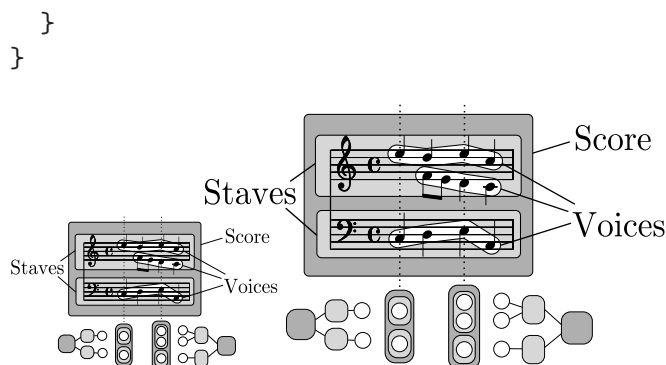
Used properties:

- `thickness` (1)

`\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
  \filled-box #'(1 . 8) #'(0 . 7) #0.2
  \with-color #white
  \filled-box #'(-4.5 . -2.5) #'(3.5 . 5.5) #0.7
}
```



`\hbracket` *arg* (markup)

Draw horizontal brackets around *arg*.

```
\markup {
  \hbracket {
    \line {
      one two three
    }
  }
}
```

one two three

`\postscript` *str* (string)

This inserts *str* directly into the output as a PostScript command string. Due to technicalities of the output backends, different scales should be used for the T<sub>E</sub>X and PostScript backend, selected with `-f`.

For the T<sub>E</sub>X backend, the following string prints a rotated text

```
0 0 moveto /ecrm10 findfont
1.75 scalefont setfont 90 rotate (hello) show
```

The magical constant 1.75 scales from LilyPond units (staff spaces) to T<sub>E</sub>X dimensions.

For the postscript backend, use the following

```

gsave /ecrm10 findfont
  10.0 output-scale div
  scalefont setfont 90 rotate (hello) show grestore

eyeglassesps = #"
  0.15 setlinewidth
  -0.9 0 translate
  1.1 1.1 scale
  1.2 0.7 moveto
  0.7 0.7 0.5 0 361 arc
  stroke
  2.20 0.70 0.50 0 361 arc
  stroke
  1.45 0.85 0.30 0 180 arc
  stroke
  0.20 0.70 moveto
  0.80 2.00 lineto
  0.92 2.26 1.30 2.40 1.15 1.70 curveto
  stroke
  2.70 0.70 moveto
  3.30 2.00 lineto
  3.42 2.26 3.80 2.40 3.65 1.70 curveto
  stroke"

eyeglasses = \markup {
  \with-dimensions #'(0 . 4.4) #'(0 . 2.5)
  \postscript #eyeglassesps
}

\relative c'' {
  c2^\eyeglasses
  a2_\eyeglasses
}

```



`\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)

`\triangle` *filled* (boolean)

A triangle, either filled or empty.

```
\markup {
  \triangle ##t
  \hspace #2
  \triangle ##f
}
```



Used properties:

- `baseline-skip` (2)
- `font-size` (0)
- `thickness` (0.1)

`\with-url` *url* (string) *arg* (markup)

Add a link to URL *url* around *arg*. This only works in the PDF backend.

```
\markup {
  \with-url #"http://lilypond.org/web/" {
    LilyPond ... \italic {
      music notation for everyone
    }
  }
}
```

LilyPond ... *music notation for everyone*

## B.6.4 Music

`\doubleflat`

Draw a double flat symbol.

```
\markup {
  \doubleflat
}
```



`\doublesharp`

Draw a double sharp symbol.

```
\markup {
  \doublesharp
}
```

✕

`\flat`

Draw a flat symbol.

```
\markup {
  \flat
}
```

♭

`\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 “La tipografía Feta”](#) in *Referencia de la Notación* for a complete listing of the possible glyphs.

```
\markup {
  \musicglyph #\"f\"
  \musicglyph #\"rests.2\"
  \musicglyph #\"clefs.G_change\"
}
```

*f* ♪

`\natural`

Draw a natural symbol.

```
\markup {
  \natural
}
```

♮

`\note-by-number` *log* (number) *dot-count* (number) *dir* (number)

Construct a note symbol, with stem. By using fractional values for *dir*, you can obtain longer or shorter stems.

```
\markup {
  \note-by-number #3 #0 #DOWN
  \hspace #2
  \note-by-number #1 #2 #0.8
}
```

♩ ♪..

Used properties:

- `style '()`
- `font-size (0)`

`\note` *duration* (string) *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 ('())`
- `font-size (0)`

`\score score` (unknown)

Inline an image of music.

```
\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
      }
    }
  }
}
```



`\semiflat`

Draw a semiflat symbol.

```
\markup {
  \semiflat
}
```



`\semisharp`

Draw a semi sharp 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 {
  \tied-lyric #"Lasciate~i monti"
}
```

Lasciate*~*i monti

## B.6.5 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 (in staff spaces). Default: 0.05.
  - **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* (list)

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*.

`(place-fret string-number fret-number finger-value)`

Place a fret playing indication on string *string-number* at fret *fret-number* with an optional fingering label *finger-value*. By default, the fret playing indicator is a solid dot. This can be changed by setting the value of the variable *dot-color*. 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` (1.0)

## B.6.6 Other

`\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
}
```



Used properties:

- `thickness` (1.6)

- `font-size` (0)

`\char` *num* (integer)

Produce a single character. For example, `\char #65` produces the letter ‘A’.

```
\markup {
  \char #65
}
```

A

`\fraction` *arg1* (markup) *arg2* (markup)

Make a fraction of two markups.

```
\markup {
  \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***

`\lookup` *glyph-name* (string)

Lookup a glyph by name.

```
\markup {
  \override #'(font-encoding . fetaBraces) {
    \lookup #"brace200"
    \hspace #2
    \rotate #180
    \lookup #"brace180"
  }
}
```



$$\left\{ \begin{array}{l} \left( \right. \\ \left. \right) \end{array} \right\}$$

`\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
}
```

**I    AA**

`\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
}
```

**J    AB**

`\null`

An empty markup with extents of a single point.

```
\markup {
  \null
}
```

`\on-the-fly` *procedure* (symbol) *arg* (markup)

Apply the *procedure* markup command to *arg*. *procedure* should take a single argument.

`\override` *new-prop* (pair) *arg* (markup)

Add the first argument in to the property list. Properties may be any sort of property supported by `font-interface` and `text-interface`, for example

```
\override #'(font-family . married) "bla"
\markup {
  \line {
    \column {
      default
      baseline-skip
    }
    \hspace #2
    \override #'(baseline-skip . 4) {
      \column {
        increased
        baseline-skip
      }
    }
  }
}
```

```
    }
  }
}
```

<b>default</b>	<b>increased</b>
<b>baseline-skip</b>	<b>baseline-skip</b>

`\page-ref` *label* (symbol) *gauge* (markup) *default* (markup)  
 Reference to a page number. *label* is the label set on the referenced page (using the `\label` command), *gauge* a markup used to estimate the maximum width of the page number, and *default* the value to display when *label* is not found.

`\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* (unknown)  
 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 the argument transparent.

```
\markup {
  \transparent {
    invisible text
  }
}
```

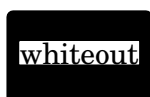
`\verbatim-file` *name* (string)  
 Read the contents of a file, and include it verbatim.

```
\markup {
  \verbatim-file #"simple.ly"
}

%% A simple piece in LilyPond, a scale.
\relative c' {
  c d e f g a b c
}
%% Optional helper for automatic updating by convert-ly. May be omitted.
\version "2.11.51"
```

`\whiteout` *arg* (markup)  
Provide a white background for *arg*.

```
\markup {
  \combine
    \filled-box #'(-1 . 10) #'(-3 . 4) #1
    \whiteout whiteout
}
```



`\with-color` *color* (list) *arg* (markup)  
Draw *arg* in color specified by *color*.

```
\markup {
  \with-color #red
  red
  \hspace #2
  \with-color #green
  green
  \hspace #2
  \with-color #blue
  blue
}
```

red green blue

`\with-dimensions` *x* (pair of numbers) *y* (pair of numbers) *arg* (markup)  
Set the dimensions of *arg* to *x* and *y*.

## B.7 Text markup list commands

The following commands can all be used with `\markuplines`.

`\column-lines` *args* (list of markups)

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* (list of markups)

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`

`\override-lines` *new-prop* (pair) *args* (list of markups)

Like `\override`, for markup lists.

`\wordwrap-internal` *justify* (boolean) *args* (list of markups)

Internal markup list command used to define `\justify` and `\wordwrap`.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)

`\wordwrap-lines` *args* (list of markups)

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)

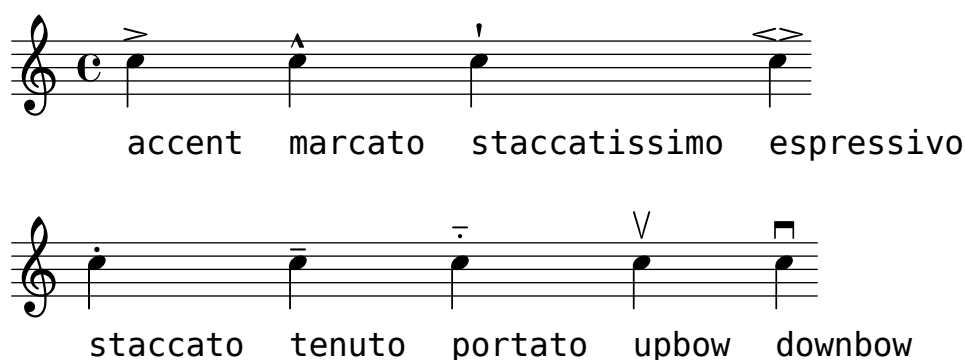
Internal markup list command used to define `\justify-string` and `\wordwrap-string`.


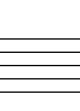

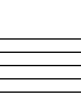
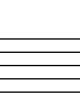
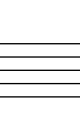


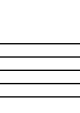

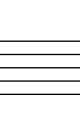

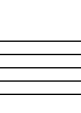

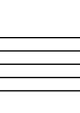

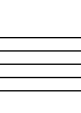
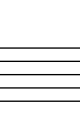
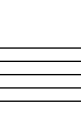

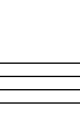

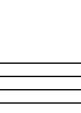

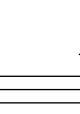
Used properties:

- `text-direction` (1)
- `word-space`
- `line-width`

## B.8 Lista de articulaciones

He aquí un cuadro que muestra todas las articulaciones disponibles:



					
flageolet	thumb	lheel	rheel	ltoe	rtoe
					
open	stopped	turn	reverseturn	trill	prall
					
mordent	prallprall	prallmordent	upprall		
					
downprall	upmordent	downmordent	pralldown		
					
prallup	lineprall	signumcongruentiae	shortfermata		
					
fermata	longfermata	verylongfermata	segno		
					
coda	varcoda				

## B.9 Todas las propiedades de contexto

`aDueText` (markup)

Text to print at a unisono passage.

`alignAboveContext` (string)

Where to insert newly created context in vertiical alignment.

`alignBassFigureAccidentals` (boolean)

If true, then the accidentals are aligned in bass figure context.

`alignBelowContext` (string)

Where to insert newly created context in vertiical alignment.

`associatedVoice` (string)

Name of the Voice 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 rule consists of

*context* In which context is the rule applied. For example, if *context* is `Score` then all staves share accidentals, and if *context* is `Staff` then all voices in the same staff share accidentals, but staves do not.

*octavation* Whether the accidental changes all octaves or only the current octave. Valid choices are

#### `same-octave`

This is the default algorithm. Accidentals are typeset if the note changes the accidental of that note in that octave. Accidentals lasts to the end of the measure and then as many measures as specified in the value. This is, `1` means to the end of next measure, `-1` means to the end of previous measure (that is: no duration at all), etc. `#t` means forever.

#### `any-octave`

Accidentals are typeset if the note is different from the previous note on the same pitch in any octave. The value has same meaning as in `same-octave`.

*laziness* Over how many bar lines the accidental lasts. If *laziness* is `-1` then the accidental is forgotten immediately, and if *laziness* is `#t` then the accidental lasts forever.

### `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.

### `autoBeamSettings` (list)

Specifies when automatically generated beams should begin and end. See [Sección “Establecer el comportamiento de las barras automáticas” in Referencia de la Notación](#) for more information.

### `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 true then bar lines will not be printed automatically; they must be explicitly created with a `\bar` command. Unlike the `\cadenza` keyword, measures are still counted. Bar 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.

**barNumberVisibility** (procedure)

A Procedure that takes an integer and returns whether the corresponding bar number should be printed.

**bassFigureFormatFunction** (procedure)

A procedure that is called to produce the formatting for a **BassFigure** grob. It takes a list of **BassFigureEvents**, a context, and the grob to format.

**bassStaffProperties** (list)

An alist of property settings to apply for the down staff of **PianoStaff**. Used by `\autochange`.

**beatGrouping** (list)

A list of beatgroups, e.g., in 5/8 time '(2 3).

**beatLength** (moment)

The length of one beat in this time signature.

**chordChanges** (boolean)

Only show changes in chords scheme?

**chordNameExceptions** (list)

An alist of chord exceptions. Contains (*chord . markup*) entries.

**chordNameExceptionsFull** (list)

An alist of full chord exceptions. Contains (*chord . markup*) entries.

**chordNameExceptionsPartial** (list)

An alist of partial chord exceptions. Contains (*chord . (prefix-markup suffix-markup)*) entries.

**chordNameFunction** (procedure)

The function that converts lists of pitches to chord names.

**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.

**clefOctavation** (integer)

Add this much extra octavation. Values of 7 and -7 are common.

**clefPosition** (number)

Where should the center of the clef symbol go, measured in half staff spaces from the center of the staff.

**completionBusy** (boolean)

Whether a completion-note head is playing.

**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 crescendi. Available values are ‘hairpin’, ‘line’, ‘dashed-line’, ‘dotted-line’. If unset, a hairpin crescendo is used.
- crescendoText** (markup)  
The text to print at start of non-hairpin crescendo, i.e., ‘**cresc.**’.
- currentBarNumber** (integer)  
Contains the current barnumber. This property is incremented at every bar line.
- decrescendoSpanner** (symbol)  
See **crescendoSpanner**.
- 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 **Timing\_translator** at **Score** level.
- 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’, ‘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.
- explicitClefVisibility** (vector)  
‘break-visibility’ function for 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 changing from a non-natural to another non-natural.
- 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.

**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.

**gridInterval** (moment)

Interval for which to generate **GridPoints**.

**harmonicAccidentals** (boolean)

If set, harmonic notes in chords get accidentals.

**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.

**ignoreFiguredBassRest** (boolean)

Don’t swallow rest events.

**ignoreMelismata** (boolean)

Ignore melismata for this **Lyrics** line.

**implicitBassFigures** (list)

A list of bass figures that are not printed as numbers, but only as extender lines.

**implicitTimeSignatureVisibility** (vector)

break visibility for the default 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 **instrument** property labels the staff in the first system, and the **instr** property labels following lines.

**instrumentTransposition** (pitch)

Define the transposition of the instrument. Its value is the pitch that sounds like 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)

An alist that defines in what order alterations should be printed. The format is `(step . alter)`, where *step* is a number from 0 to 6 and *alter* from -2 (sharp) to 2 (flat).

**keySignature** (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. `keySignature = #'((6 . ,FLAT))`.

**lyricMelismaAlignment** (direction)

Alignment to use for a melisma syllable.

**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.

**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 four arguments: text, duration, count 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`.

- 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**.
- midiInstrument** (string)  
Name of the MIDI instrument to use.
- midiMaximumVolume** (number)  
Analogous to **midiMinimumVolume**.
- midiMinimumVolume** (number)  
Set the minimum loudness for MIDI. Ranges from 0 to 1.
- 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.
- noteToFretFunction** (procedure)  
How to produce a fret diagram. Parameters: A list of note events and a list of tabstring events.
- ottavation** (markup)  
If set, the text for an ottava spanner. Changing this creates a new text spanner.
- output** (unknown)  
The output produced by a score-level translator during music interpretation.
- 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**.
- printKeyCancellation** (boolean)  
Print restoration alterations before a key signature change.

- printOctaveNames** (boolean)  
Print octave marks for 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.
- recordEventSequence** (procedure)  
When **Recording\_group\_engraver** is in this context, then upon termination of the context, this function is called with current context and a list of music objects. The list contains entries with start times, music objects and whether they are processed in this context.
- 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.
- restNumberThreshold** (number)  
If a multimeasure rest has more measures than this, a number is printed.
- shapeNoteStyles** (vector)  
Vector of symbols, listing style for each note head relative to the tonic (qv.) of the scale.
- shortInstrumentName** (markup)  
See **instrument**.
- 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.
- 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 **Pitch\_squash\_engraver**.

**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.

**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**.

**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 pitch (in semitones) of each string (starting with the lower one).

**strokeFingerOrientations** (list)

See **fingeringOrientations**.

**subdivideBeams** (boolean)

If set, multiple beams will be subdivided at beat positions by only drawing one beam over the beat.

**suggestAccidentals** (boolean)

If set, accidentals are typeset as cautionary suggestions over the note.

**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: string number, context and event. It returns the text as a string.

**tempoHideNote** (boolean)

Hide the note=count in tempo marks.

**tempoText** (markup)

Text for tempo marks.

**tempoUnitCount** (number)

Count for specifying tempo.

**tempoUnitDuration** (duration)

Unit for specifying tempo.

**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** (pair of numbers)

A pair of numbers, signifying the time signature. For example, `#'(4 . 4)` is a 4/4 time signature.

**timing** (boolean)

Keep administration of measure length, position, bar number, etc.? Switch off for cadenzas.

**tonic** (pitch)

The tonic of the current scale.

**trebleStaffProperties** (list)

An alist of property settings to apply for the up staff of `PianoStaff`. Used by `\autochange`.

**tremoloFlags** (integer)

The number of tremolo flags to add if no number is specified.

**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 }
}
```

**useBassFigureExtenders** (boolean)

Whether to use extender lines for repeated bass figures.

**verticallySpacedContexts** (list)

List of symbols, containing context names whose vertical axis groups should be taken into account for vertical spacing of systems.

**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 `bar-line-interface`.

## B.10 Propiedades de disposición

- X-extent** (pair of numbers)  
Hard coded extent in X direction.
- X-offset** (number)  
The horizontal amount that this object is moved relative to its X-parent.
- Y-extent** (pair of numbers)  
Hard coded extent in Y direction.
- Y-offset** (number)  
The vertical amount that this object is moved relative to its Y-parent.
- 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.
- annotation** (string)  
Annotate a grob for debug purposes.
- 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 that is larger than this number, make a kneed beam.
- 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-slur** (symbol)  
Method of handling slur collisions. Choices are **around**, **inside**, **outside**. If unset, scripts and slurs ignore each other. **around** only moves the script if there is a collision; **outside** always moves the script.

- axes** (list) List of axis numbers. In the case of alignment grobs, this should contain only one number.
- bar-size** (dimension, in staff space)  
The size of a 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.
- before-line-breaking** (boolean)  
Dummy property, used to trigger a callback function.
- 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 break-align grob 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)  
Defines the order in which prefatory matter (clefs, key signatures) appears. The format is a vector of length 3, where each element is one order for end-of-line, middle of line, and start-of-line, respectively. An order is a list of symbols.



For example, clefs are put after key signatures by setting

```
\override Score.BreakAlignment #'break-align-orders =
  #(make-vector 3 '(span-bar
                    breathing-sign
                    staff-bar
                    key
                    clef
                    time-signature))
```

**break-align-symbol** (symbol)

This key is used for aligning and spacing breakable items.

**break-align-symbols** (list)

A list of symbols that determine which break-aligned grobs 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).

**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.

**c0-position** (integer)

An integer indicating the position of middle C.

**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.

**color** (list)

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)

List of offsets (number pairs) that form control points for the tie, slur, or bracket shape. For Bézier, this should list the control points of a third-order Bézier curve.

**damping** (number)

Amount of beam slope damping.

- dash-fraction** (number)  
Size of the dashes, relative to **dash-period**. Should be between 0.0 (no line) and 1.0 (continuous line).
- dash-period** (number)  
The length of one dash together with whitespace. If negative, no line is drawn at all.
- default-direction** (direction)  
Direction determined by note head positions.
- digit-names** (unknown)  
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-placement-list** (list)  
List consisting of (*description string-number fret-number finger-number*) entries used to define fret diagrams.
- 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*).
- expand-limit** (integer)  
Maximum number of measures expanded in church rests.
- extra-X-extent** (pair of numbers)  
A grob is enlarged in X dimension by this much.
- extra-Y-extent** (pair of numbers)  
A grob is enlarged in Y dimension by this much.
- 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)`.

**flag-count** (number)

The number of tremolo beams.

**flag-style** (symbol)

A string determining what style of flag glyph is typeset on a **Stem**. Valid options include `()` and **mensural**. Additionally, **no-flag** switches off the flag.

**font-encoding** (symbol)

The font encoding is the broadest category for selecting a font. Options include: **fetaMusic**, **fetaNumber**, **TeX-text**, **TeX-math**, **fetaBraces**, **fetaDynamic**.

**font-family** (symbol)

The font family is the broadest category for selecting text fonts. Options include: **sans**, **roman**.

**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. Fractional values are allowed.

**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 **note-collision-interface**.

**forced** (boolean)

Manually forced accidental.

**fraction** (pair of numbers)

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**.
- **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-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-label-font-mag** – The magnification of the font used to label the lowest fret number. Default 0.5
- **fret-label-vertical-offset** – The vertical offset of the fret label from the fret. Default -0.2
- **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**, and **arabic**. Default **roman-lower**.
- **open-string** – Character string to be used to indicate open string. Default "o".
- **orientation** – Orientation of fret-diagram. Options include **normal** and **landscape**. Default **normal**.
- **string-count** – The number of strings. Default 6.
- **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.
- **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-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.

**glyph** (string)

A string determining what 'style' of glyph is typeset. Valid choices depend on the function that is reading this property.

**glyph-name-alist** (list)

An alist of key-string pairs.

**grow-direction** (direction)

Crescendo or decrescendo?

**hair-thickness** (number)

Thickness of the thin line in a bar line.

**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.

**horizontal-shift** (integer)

An integer that identifies ranking of **NoteColumns** for horizontal shifting. This is used by **note-collision-interface**.

**horizontal-skylines** (unknown)

Two skylines, one to the left and one to the right of this grob.

**ignore-collision** (boolean)

If set, don't do note collision resolution on this **NoteColumn**.

**implicit** (boolean)

Is this an implicit bass figure?

**inspect-index** (integer)

If debugging is set, set beam and slur configuration to this index, and print the respective scores.

**inspect-quants** (pair of numbers)

If debugging is set, set beam and slur quants to this position, and print the respective scores.

**keep-fixed-while-stretching** (boolean)

A grob with this property set to true is fixed relative to the staff above it when systems are stretched.

**keep-inside-line** (boolean)

If set, this column cannot have objects sticking into the margin.

**kern** (dimension, in staff space)

Amount of extra white space to add. For bar lines, this is the amount of space after a thick line.

**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.

**labels** (list)

List of labels (symbols) placed on a column

**layer** (integer)

The output layer (a value between 0 and 2: Layers define the order of printing objects. Objects in lower layers are overprinted by objects in higher layers.

**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.

**left-bound-info** (list)

An alist of properties for determining attachments of spanners to edges.

**left-padding** (dimension, in staff space)

The amount of space that is put left to an object (e.g., a group of accidentals).

**length** (dimension, in staff space)

User override for the stem length of unbeamed stems.

**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)

The thickness of the tie or slur contour.

**long-text** (markup)

Text markup. See [Sección “Formatear el texto”](#) in *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-stretch** (number)

The maximum amount that this `VerticalAxisGroup` can be vertically stretched (for example, in order to better fill a page).

**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 `note-collision-interface`.

**merge-differently-headed** only applies to opposing stem directions (i.e., voice 1 & 2).

**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.

**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-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).

**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** (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-default** (boolean)  
Set for manually specified clefs.
- non-musical** (boolean)  
True if the grob belongs to a `NonMusicalPaperColumn`.
- note-names** (vector)  
Vector of strings containing names for easy-notation note heads.
- outside-staff-horizontal-padding** (number)  
By default, an outside-staff-object can be placed so that is it 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 this grob and the staff when spacing according to **outside-staff-priority**.
- 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.
- 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-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**.
- parenthesized** (boolean)  
Parenthesize this grob.
- positions** (pair of numbers)  
Pair of staff coordinates (*left* . *right*), where both *left* and *right* are 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.
- 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 a orchestral score?
- 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-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.
- 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 sorting key that determines in what order a script is within a stack of scripts.
- 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.
- self-alignment-Y** (number)  
Like **self-alignment-X** but for the Y axis.
- shorten-pair** (pair of numbers)  
The lengths to shorten a text-spanner on both sides, for example a pedal bracket. Positive values shorten the text-spanner, while negative values lengthen it.
- shortest-duration-space** (dimension, in staff space)  
Start with this much space for the shortest duration. This is expressed in **spacing-increment** as unit. See also **spacing-spanner-interface**.
- 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.
- 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.

**size** (number)

Size of object, relative to standard size.

**slope** (number)

The slope of this object.

**slur-padding** (number)

Extra distance between slur and script.

**space-alist** (list)

A table that specifies distances between prefatory items, like clef and time-signature. The format is an alist of spacing tuples: (*break-align-symbol type . distance*), where *type* can be the symbols **minimum-space** or **extra-space**.

**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** (number)

Add this much space for a doubled duration. Typically, the width of a note head. See also **spacing-spanner-interface**.

**springs-and-rods** (boolean)

Dummy variable for triggering spacing routines.

**stacking-dir** (direction)

Stack objects in which direction?

**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**.

**stem-attachment** (pair of numbers)

An (*x . y*) pair where the stem attaches to the notehead.

**stem-end-position** (number)

Where does the stem end (the end is opposite to the support-head)?

**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 a stem over a rest be?

**stencil** (unknown)

The symbol to print.

**stencils** (list)

Multiple stencils, used as intermediate value.

**strict-grace-spacing** (boolean)

If set, grace notes are not spaced separately, but put before musical columns.

**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 “Formatear el texto” in 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)

Bar line thickness, measured in **line-thickness**.

**thickness** (number)

Line thickness, generally measured in **line-thickness**.

**thin-kern** (number)

The space after a hair-line in a bar line.

**threshold** (pair of numbers)

(*min* . *max*), where *min* and *max* are dimensions in staff space.

**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.

**transparent** (boolean)

This makes the grob invisible.

**uniform-stretching** (boolean)

If set, items stretch proportionally to their durations. This looks better in complex polyphonic patterns.

**used** (boolean)

If set, this spacing column is kept in the spacing problem.

**vertical-skylines** (unknown)

Two skylines, one above and one below this grob.

**when** (moment)

Global time step associated with this column happen?

**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.

**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 glissando line can be constructed from a whole number of squiggles.

## B.11 Identificadores

**acciaccatura** - *music* (music)

(undocumented; fixme)

**addChordShape** - *key-symbol* (symbol) *shape-string* (string)

(undocumented; fixme)

**addInstrumentDefinition** - *name* (string) *lst* (list)

(undocumented; fixme)

**addQuote** - *name* (string) *music* (music)

(undocumented; fixme)

**afterGrace** - *main* (music) *grace* (music)

(undocumented; fixme)

**allowPageTurn**

(undocumented; fixme)

**applyContext** - *proc* (procedure)

(undocumented; fixme)

**applyMusic** - *func* (procedure) *music* (music)

(undocumented; fixme)

**applyOutput** - *ctx* (symbol) *proc* (procedure)

(undocumented; fixme)

**appoggiatura** - *music* (music)

(undocumented; fixme)

**assertBeamQuant** - *l* (pair) *r* (pair)

(undocumented; fixme)

**assertBeamSlope** - *comp* (procedure)

(undocumented; fixme)

**autochange** - *music* (music)

(undocumented; fixme)

**balloonGrobText** - *grob-name* (symbol) *offset* (pair of numbers) *text* (markup)

(undocumented; fixme)

**balloonText** - *offset* (pair of numbers) *text* (markup)

(undocumented; fixme)

**bar** - *type* (string)

(undocumented; fixme)

**barNumberCheck** - *n* (integer)

(undocumented; fixme)

**bendAfter** - *delta* (unknown)

(undocumented; fixme)

**breathe** (undocumented; fixme)  
**clef - type** (string)  
 (undocumented; fixme)  
**cueDuring** - *what* (string) *dir* (direction) *main-music* (music)  
 (undocumented; fixme)  
**displayLilyMusic** - *music* (music)  
 (undocumented; fixme)  
**displayMusic** - *music* (music)  
 (undocumented; fixme)  
**endSpanners** - *music* (music)  
 (undocumented; fixme)  
**featherDurations** - *factor* (moment) *argument* (music)  
 (undocumented; fixme)  
**grace** - *music* (music)  
 (undocumented; fixme)  
**includePageLayoutFile**  
 (undocumented; fixme)  
**instrumentSwitch** - *name* (string)  
 (undocumented; fixme)  
**keepWithTag** - *tag* (symbol) *music* (music)  
 (undocumented; fixme)  
**killCues** - *music* (music)  
 (undocumented; fixme)  
**label** - *label* (symbol)  
 (undocumented; fixme)  
**makeClusters** - *arg* (music)  
 (undocumented; fixme)  
**musicMap** - *proc* (procedure) *mus* (music)  
 (undocumented; fixme)  
**noPageBreak**  
 (undocumented; fixme)  
**noPageTurn**  
 (undocumented; fixme)  
**octaveCheck** - *pitch-note* (music)  
 (undocumented; fixme)  
**oldaddyrics** - *music* (music) *lyrics* (music)  
 (undocumented; fixme)  
**ottava** - *octave* (number)  
 (undocumented; fixme)  
**overrideProperty** - *name* (string) *property* (symbol) *value* (any type)  
 (undocumented; fixme)  
**pageBreak**  
 (undocumented; fixme)

`pageTurn` - *(undocumented; fixme)*  
`parallelMusic` - *voice-ids* (list) *music* (music)  
                   *(undocumented; fixme)*  
`parenthesize` - *arg* (music)  
                   *(undocumented; fixme)*  
`partcombine` - *part1* (music) *part2* (music)  
                   *(undocumented; fixme)*  
`pitchedTrill` - *main-note* (music) *secondary-note* (music)  
                   *(undocumented; fixme)*  
`pointAndClickOff`  
                   *(undocumented; fixme)*  
`pointAndClickOn`  
                   *(undocumented; fixme)*  
`quoteDuring` - *what* (string) *main-music* (music)  
                   *(undocumented; fixme)*  
`removeWithTag` - *tag* (symbol) *music* (music)  
                   *(undocumented; fixme)*  
`resetRelativeOctave` - *reference-note* (music)  
                   *(undocumented; fixme)*  
`rightHandFinger` - *finger* (number or string)  
                   *(undocumented; fixme)*  
`scaleDurations` - *fraction* (pair of numbers) *music* (music)  
                   *(undocumented; fixme)*  
`scoreTweak` - *name* (string)  
                   *(undocumented; fixme)*  
`shiftDurations` - *dur* (integer) *dots* (integer) *arg* (music)  
                   *(undocumented; fixme)*  
`spacingTweaks` - *parameters* (list)  
                   *(undocumented; fixme)*  
`storePredefinedDiagram` - *chord* (music) *tuning* (list) *terse-definition* (string)  
                   *(undocumented; fixme)*  
`tag` - *tag* (symbol) *arg* (music)  
                   *(undocumented; fixme)*  
`tocItem` - *text* (markup)  
                   Add a line to the table of content, using the `tocItemMarkup` paper variable markup  
`transposedCueDuring` - *what* (string) *dir* (direction) *pitch-note* (music) *main-music* (music)  
                   *(undocumented; fixme)*  
`transposition` - *pitch-note* (music)  
                   *(undocumented; fixme)*  
`tweak` - *sym* (symbol) *val* (any type) *arg* (music)  
                   *(undocumented; fixme)*  
`unfoldRepeats` - *music* (music)  
                   *(undocumented; fixme)*  
`withMusicProperty` - *sym* (symbol) *val* (any type) *music* (music)  
                   *(undocumented; fixme)*

## B.12 Funciones de Scheme

|                                                                                                                                                                                                    |           |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>dispatcher</b> <i>x</i>                                                                                                                                                                         | [Función] |
| Is <i>x</i> a Dispatcher object?                                                                                                                                                                   |           |
| <b>listener</b> <i>x</i>                                                                                                                                                                           | [Función] |
| Is <i>x</i> a Listener object?                                                                                                                                                                     |           |
| <b>ly:add-file-name-alist</b> <i>alist</i>                                                                                                                                                         | [Función] |
| Add mappings for error messages from <i>alist</i> .                                                                                                                                                |           |
| <b>ly:add-interface</b> <i>a b c</i>                                                                                                                                                               | [Función] |
| Add an interface description.                                                                                                                                                                      |           |
| <b>ly:add-listener</b> <i>list disp cl</i>                                                                                                                                                         | [Función] |
| Add the listener <i>list</i> to the dispatcher <i>disp</i> . Whenever <i>disp</i> hears an event of class <i>cl</i> , it is forwarded to <i>list</i> .                                             |           |
| <b>ly:add-option</b> <i>sym val description</i>                                                                                                                                                    | [Función] |
| Add a program option <i>sym</i> with default <i>val</i> .                                                                                                                                          |           |
| <b>ly:all-grob-interfaces</b>                                                                                                                                                                      | [Función] |
| Get a hash table with all interface descriptions.                                                                                                                                                  |           |
| <b>ly:all-options</b>                                                                                                                                                                              | [Función] |
| Get all option settings in an alist.                                                                                                                                                               |           |
| <b>ly:all-stencil-expressions</b>                                                                                                                                                                  | [Función] |
| Return all symbols recognized as stencil expressions.                                                                                                                                              |           |
| <b>ly:assoc-get</b> <i>key alist default-value</i>                                                                                                                                                 | [Función] |
| Return value if <i>key</i> in <i>alist</i> , else <i>default-value</i> (or <i>#f</i> if not specified).                                                                                            |           |
| <b>ly:book-add-score!</b> <i>book-smob score</i>                                                                                                                                                   | [Función] |
| Add <i>score</i> to <i>book-smob</i> score list.                                                                                                                                                   |           |
| <b>ly:book-process</b> <i>book-smob default-paper default-layout output</i>                                                                                                                        | [Función] |
| Print book. <i>output</i> is passed to the backend unchanged. For example, it may be a string (for file based outputs) or a socket (for network based output).                                     |           |
| <b>ly:book-process-to-systems</b> <i>book-smob default-paper default-layout output</i>                                                                                                             | [Función] |
| Print book. <i>output</i> is passed to the backend unchanged. For example, it may be a string (for file based outputs) or a socket (for network based output).                                     |           |
| <b>ly:box?</b> <i>x</i>                                                                                                                                                                            | [Función] |
| Is <i>x</i> a Box object?                                                                                                                                                                          |           |
| <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>ly:broadcast</b> <i>disp ev</i>                                                                                                                                                                 | [Función] |
| Send the stream event <i>ev</i> to the dispatcher <i>disp</i> .                                                                                                                                    |           |

|                                                                                                                                                                                                                     |           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <code>ly:camel-case-&gt;lisp-identifier</code> <i>name-sym</i>                                                                                                                                                      | [Función] |
| Convert <code>FooBar_Bla</code> to <code>foo-bar-bla</code> style symbol.                                                                                                                                           |           |
| <code>ly:chain-assoc-get</code> <i>key achain dfault</i>                                                                                                                                                            | [Función] |
| Return value for <i>key</i> from a list of alists <i>achain</i> . If no entry is found, return <i>dfault</i> or <code>#f</code> if no <i>dfault</i> is specified.                                                   |           |
| <code>ly:clear-anonymous-modules</code>                                                                                                                                                                             | [Función] |
| Plug a GUILE 1.6 and 1.7 memory leak by breaking a weak reference pointer cycle explicitly.                                                                                                                         |           |
| <code>ly:cm</code> <i>num</i>                                                                                                                                                                                       | [Función] |
| <i>num</i> cm.                                                                                                                                                                                                      |           |
| <code>ly:command-line-code</code>                                                                                                                                                                                   | [Función] |
| The Scheme code specified on command-line with <code>‘-e’</code> .                                                                                                                                                  |           |
| <code>ly:command-line-options</code>                                                                                                                                                                                | [Función] |
| The Scheme options specified on command-line with <code>‘-d’</code> .                                                                                                                                               |           |
| <code>ly:command-line-verbose?</code>                                                                                                                                                                               | [Función] |
| Was <code>be_verbose_global</code> set?                                                                                                                                                                             |           |
| <code>ly:connect-dispatchers</code> <i>to from</i>                                                                                                                                                                  | [Función] |
| Make the dispatcher <i>to</i> listen to events from <i>from</i> .                                                                                                                                                   |           |
| <code>ly:context-event-source</code> <i>context</i>                                                                                                                                                                 | [Función] |
| Return <code>event-source</code> of context <i>context</i> .                                                                                                                                                        |           |
| <code>ly:context-events-below</code> <i>context</i>                                                                                                                                                                 | [Función] |
| Return a <code>stream-distributor</code> that distributes all events from <i>context</i> and all its subcontexts.                                                                                                   |           |
| <code>ly:context-find</code> <i>context name</i>                                                                                                                                                                    | [Función] |
| Find a parent of <i>context</i> that has name or alias <i>name</i> . Return <code>#f</code> if not found.                                                                                                           |           |
| <code>ly:context-grob-definition</code> <i>context name</i>                                                                                                                                                         | [Función] |
| Return the definition of <i>name</i> (a symbol) within <i>context</i> as an alist.                                                                                                                                  |           |
| <code>ly:context-id</code> <i>context</i>                                                                                                                                                                           | [Función] |
| Return the ID string of <i>context</i> , i.e., for <code>\context Voice = one ...</code> return the string <code>one</code> .                                                                                       |           |
| <code>ly:context-name</code> <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> .                                                                                          |           |
| <code>ly:context-now</code> <i>context</i>                                                                                                                                                                          | [Función] |
| Return <code>now-moment</code> of context <i>context</i> .                                                                                                                                                          |           |
| <code>ly:context-parent</code> <i>context</i>                                                                                                                                                                       | [Función] |
| Return the parent of <i>context</i> , <code>#f</code> if none.                                                                                                                                                      |           |
| <code>ly:context-property</code> <i>c name</i>                                                                                                                                                                      | [Función] |
| Return the value of <i>name</i> from context <i>c</i> .                                                                                                                                                             |           |
| <code>ly:context-property-where-defined</code> <i>context name</i>                                                                                                                                                  | [Función] |
| Return the context above <i>context</i> where <i>name</i> is defined.                                                                                                                                               |           |
| <code>ly:context-pushpop-property</code> <i>context grob eltprop val</i>                                                                                                                                            | [Función] |
| Do a single <code>\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>ly:context-unset-property</b> <i>context name</i>                                                                                                                             | [Función] |
| Unset value of property <i>name</i> in context <i>context</i> .                                                                                                                  |           |
| <b>ly:context?</b> <i>x</i>                                                                                                                                                      | [Función] |
| Is <i>x</i> a <b>Context</b> object?                                                                                                                                             |           |
| <b>ly:default-scale</b>                                                                                                                                                          | [Función] |
| Get the global default scale.                                                                                                                                                    |           |
| <b>ly:dimension?</b> <i>d</i>                                                                                                                                                    | [Función] |
| Return <i>d</i> as a number. Used to distinguish length variables from normal numbers.                                                                                           |           |
| <b>ly:dir?</b> <i>s</i>                                                                                                                                                          | [Función] |
| A type predicate. The direction <i>s</i> is -1, 0 or 1, where -1 represents left or down and 1 represents right or up.                                                           |           |
| <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>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>ly:duration-log</b> <i>dur</i>                                                                                                                                                | [Función] |
| Extract the duration log from <i>dur</i> .                                                                                                                                       |           |
| <b>ly:duration&lt;?</b> <i>p1 p2</i>                                                                                                                                             | [Función] |
| Is <i>p1</i> shorter than <i>p2</i> ?                                                                                                                                            |           |
| <b>ly:duration?</b> <i>x</i>                                                                                                                                                     | [Función] |
| Is <i>x</i> a <b>Duration</b> object?                                                                                                                                            |           |
| <b>ly:effective-prefix</b>                                                                                                                                                       | [Función] |
| Return effective prefix.                                                                                                                                                         |           |
| <b>ly:error</b> <i>str rest</i>                                                                                                                                                  | [Función] |
| A Scheme callable function to issue the error <i>str</i> . The error is formatted with <b>format</b> and <i>rest</i> .                                                           |           |
| <b>ly:eval-simple-closure</b> <i>delayed closure scm-start scm-end</i>                                                                                                           | [Función] |
| Evaluate a simple <i>closure</i> with the given <i>delayed</i> argument. If <i>scm-start</i> and <i>scm-end</i> are defined, evaluate it purely with those start and end points. |           |
| <b>ly:event-deep-copy</b> <i>m</i>                                                                                                                                               | [Función] |
| Copy <i>m</i> and all sub expressions of <i>m</i> .                                                                                                                              |           |
| <b>ly:event-property</b> <i>sev sym</i>                                                                                                                                          | [Función] |
| Get the property <i>sym</i> of stream event <i>mus</i> . If <i>sym</i> is undefined, return '().                                                                                 |           |

|                                                                                                                             |           |
|-----------------------------------------------------------------------------------------------------------------------------|-----------|
| <code>ly:event-set-property! ev sym val</code>                                                                              | [Función] |
| Set property <i>sym</i> in event <i>ev</i> to <i>val</i> .                                                                  |           |
| <code>ly:expand-environment str</code>                                                                                      | [Función] |
| Expand <i>\$VAR</i> and <i>\${VAR}</i> in <i>str</i> .                                                                      |           |
| <code>ly:export arg</code>                                                                                                  | [Función] |
| Export a Scheme object to the parser so it is treated as an identifier.                                                     |           |
| <code>ly:find-file name</code>                                                                                              | [Función] |
| Return the absolute file name of <i>name</i> , or <i>#f</i> if not found.                                                   |           |
| <code>ly:font-config-display-fonts</code>                                                                                   | [Función] |
| Dump a list of all fonts visible to FontConfig.                                                                             |           |
| <code>ly:font-config-get-font-file name</code>                                                                              | [Función] |
| Get the file for font <i>name</i> .                                                                                         |           |
| <code>ly:font-design-size font</code>                                                                                       | [Función] |
| Given the font metric <i>font</i> , return the design size, relative to the current output-scale.                           |           |
| <code>ly:font-file-name font</code>                                                                                         | [Función] |
| Given the font metric <i>font</i> , return the corresponding file name.                                                     |           |
| <code>ly:font-get-glyph font name</code>                                                                                    | [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. |           |
| <code>ly:font-glyph-name-to-charcode font name</code>                                                                       | [Función] |
| Return the character code for glyph <i>name</i> in <i>font</i> .                                                            |           |
| <code>ly:font-glyph-name-to-index font name</code>                                                                          | [Función] |
| Return the index for <i>name</i> in <i>font</i> .                                                                           |           |
| <code>ly:font-index-to-charcode font index</code>                                                                           | [Función] |
| Return the character code for <i>index</i> in <i>font</i> .                                                                 |           |
| <code>ly:font-load name</code>                                                                                              | [Función] |
| Load the font <i>name</i> .                                                                                                 |           |
| <code>ly:font-magnification font</code>                                                                                     | [Función] |
| Given the font metric <i>font</i> , return the magnification, relative to the current output-scale.                         |           |
| <code>ly:font-metric? x</code>                                                                                              | [Función] |
| Is <i>x</i> a Font_metric object?                                                                                           |           |
| <code>ly:font-name font</code>                                                                                              | [Función] |
| Given the font metric <i>font</i> , return the corresponding name.                                                          |           |
| <code>ly:font-sub-fonts font</code>                                                                                         | [Función] |
| Given the font metric <i>font</i> of an OpenType font, return the names of the subfonts within <i>font</i> .                |           |
| <code>ly:format str rest</code>                                                                                             | [Función] |
| LilyPond specific format, supporting <i>~a</i> and <i>~[0-9]f</i> .                                                         |           |
| <code>ly:format-output context</code>                                                                                       | [Función] |
| Given a global context in its final state, process it and return the Music_output object in its final state.                |           |

|                                                                                                                                                          |           |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>ly:get-all-function-documentation</b>                                                                                                                 | [Función] |
| Get a hash table with all LilyPond Scheme extension functions.                                                                                           |           |
| <b>ly:get-all-translators</b>                                                                                                                            | [Función] |
| Return a list of all translator objects that may be instantiated.                                                                                        |           |
| <b>ly:get-glyph <i>font index</i></b>                                                                                                                    | [Función] |
| Retrieve a stencil for the glyph numbered <i>index</i> in <i>font</i> .                                                                                  |           |
| <b>ly:get-listened-event-classes</b>                                                                                                                     | [Función] |
| Return a list of all event classes that some translator listens to.                                                                                      |           |
| <b>ly:get-option <i>var</i></b>                                                                                                                          | [Función] |
| Get a global option setting.                                                                                                                             |           |
| <b>ly:gettext <i>original</i></b>                                                                                                                        | [Función] |
| A Scheme wrapper function for <b>gettext</b> .                                                                                                           |           |
| <b>ly:grob-alist-chain <i>grob global</i></b>                                                                                                            | [Función] |
| Get an alist chain for grob <i>grob</i> , with <i>global</i> as the global default. If unspecified, <b>font-defaults</b> from the layout block is taken. |           |
| <b>ly:grob-array-length <i>grob-arr</i></b>                                                                                                              | [Función] |
| Return the length of <i>grob-arr</i> .                                                                                                                   |           |
| <b>ly:grob-array-ref <i>grob-arr index</i></b>                                                                                                           | [Función] |
| Retrieve the <i>index</i> th element of <i>grob-arr</i> .                                                                                                |           |
| <b>ly:grob-array? <i>x</i></b>                                                                                                                           | [Función] |
| Is <i>x</i> a <b>Grob_array</b> object?                                                                                                                  |           |
| <b>ly:grob-basic-properties <i>grob</i></b>                                                                                                              | [Función] |
| Get the immutable properties of <i>grob</i> .                                                                                                            |           |
| <b>ly:grob-common-refpoint <i>grob other axis</i></b>                                                                                                    | [Función] |
| Find the common refpoint of <i>grob</i> and <i>other</i> for <i>axis</i> .                                                                               |           |
| <b>ly:grob-common-refpoint-of-array <i>grob others axis</i></b>                                                                                          | [Función] |
| Find the common refpoint of <i>grob</i> and <i>others</i> (a grob-array) for <i>axis</i> .                                                               |           |
| <b>ly:grob-default-font <i>grob</i></b>                                                                                                                  | [Función] |
| Return the default font for grob <i>gr</i> .                                                                                                             |           |
| <b>ly:grob-extent <i>grob refp axis</i></b>                                                                                                              | [Función] |
| Get the extent in <i>axis</i> direction of <i>grob</i> relative to the grob <i>refp</i> .                                                                |           |
| <b>ly:grob-interfaces <i>grob</i></b>                                                                                                                    | [Función] |
| Return the interfaces list of grob <i>grob</i> .                                                                                                         |           |
| <b>ly:grob-layout <i>grob</i></b>                                                                                                                        | [Función] |
| Get \layout definition from grob <i>grob</i> .                                                                                                           |           |
| <b>ly:grob-object <i>grob sym</i></b>                                                                                                                    | [Función] |
| Return the value of a pointer in grob <i>g</i> of property <i>sym</i> . It returns '() (end-of-list) if <i>sym</i> is undefined in <i>g</i> .            |           |
| <b>ly:grob-original <i>grob</i></b>                                                                                                                      | [Función] |
| Return the unbroken original grob of <i>grob</i> .                                                                                                       |           |

|                                                                                                                                                                             |           |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>ly:grob-parent</b> <i>grob axis</i>                                                                                                                                      | [Función] |
| Get the parent of <i>grob</i> . <i>axis</i> is 0 for the X-axis, 1 for the Y-axis.                                                                                          |           |
| <b>ly:grob-pq&lt;?</b> <i>a b</i>                                                                                                                                           | [Función] |
| Compare two grob priority queue entries. This is an internal function.                                                                                                      |           |
| <b>ly:grob-properties</b> <i>grob</i>                                                                                                                                       | [Función] |
| Get the mutable properties of <i>grob</i> .                                                                                                                                 |           |
| <b>ly:grob-property</b> <i>grob sym deflt</i>                                                                                                                               | [Función] |
| Return the value of a value in grob <i>g</i> of property <i>sym</i> . It returns '()' (end-of-list) or <i>deflt</i> (if specified) if <i>sym</i> is undefined in <i>g</i> . |           |
| <b>ly:grob-property-data</b> <i>grob sym</i>                                                                                                                                | [Función] |
| Retrieve <i>sym</i> for <i>grob</i> but don't process callbacks.                                                                                                            |           |
| <b>ly:grob-relative-coordinate</b> <i>grob refp axis</i>                                                                                                                    | [Función] |
| Get the coordinate in <i>axis</i> direction of <i>grob</i> relative to the grob <i>refp</i> .                                                                               |           |
| <b>ly:grob-robust-relative-extent</b> <i>grob refp axis</i>                                                                                                                 | [Función] |
| Get the extent in <i>axis</i> direction of <i>grob</i> relative to the grob <i>refp</i> , or (0,0) if empty.                                                                |           |
| <b>ly:grob-script-priority-less</b> <i>a b</i>                                                                                                                              | [Función] |
| Compare two grobs by script priority. For internal use.                                                                                                                     |           |
| <b>ly:grob-set-property!</b> <i>grob sym val</i>                                                                                                                            | [Función] |
| Set <i>sym</i> in grob <i>grob</i> to value <i>val</i> .                                                                                                                    |           |
| <b>ly:grob-staff-position</b> <i>sg</i>                                                                                                                                     | [Función] |
| Return the Y-position of <i>sg</i> relative to the staff.                                                                                                                   |           |
| <b>ly:grob-suicide!</b> <i>grob</i>                                                                                                                                         | [Función] |
| Kill <i>grob</i> .                                                                                                                                                          |           |
| <b>ly:grob-system</b> <i>grob</i>                                                                                                                                           | [Función] |
| Return the system grob of <i>grob</i> .                                                                                                                                     |           |
| <b>ly:grob-translate-axis!</b> <i>grob d a</i>                                                                                                                              | [Función] |
| Translate <i>g</i> on axis <i>a</i> over distance <i>d</i> .                                                                                                                |           |
| <b>ly:grob?</b> <i>x</i>                                                                                                                                                    | [Función] |
| Is <i>x</i> a Grob object?                                                                                                                                                  |           |
| <b>ly:gulp-file</b> <i>name size</i>                                                                                                                                        | [Función] |
| Read the file <i>name</i> , and return its contents in a string. The file is looked up using the search path.                                                               |           |
| <b>ly:hash-table-keys</b> <i>tab</i>                                                                                                                                        | [Función] |
| Return a list of keys in <i>tab</i> .                                                                                                                                       |           |
| <b>ly:inch</b> <i>num</i>                                                                                                                                                   | [Función] |
| <i>num</i> 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> an input-location?                                                                                                                                                                                           |           |
| <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: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>ly:interpret-stencil-expression</b> <i>expr func arg1 offset</i>                                                                                                                                                      | [Función] |
| Parse <i>expr</i> , feed bits to <i>func</i> with first arg <i>arg1</i> having offset <i>offset</i> .                                                                                                                    |           |
| <b>ly:intlog2</b> <i>d</i>                                                                                                                                                                                               | [Función] |
| The 2-logarithm of 1/ <i>d</i> .                                                                                                                                                                                         |           |
| <b>ly:is-listened-event-class</b> <i>sym</i>                                                                                                                                                                             | [Función] |
| Is <i>sym</i> a listened event class?                                                                                                                                                                                    |           |
| <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?</b> <i>g</i>                                                                                                                                                                                                 | [Función] |
| Is <i>g</i> an Item object?                                                                                                                                                                                              |           |
| <b>ly:iterator?</b> <i>x</i>                                                                                                                                                                                             | [Función] |
| Is <i>x</i> a Music_iterator object?                                                                                                                                                                                     |           |
| <b>ly:lexer-keywords</b> <i>lexer</i>                                                                                                                                                                                    | [Función] |
| Return a list of (KEY . CODE) pairs, signifying the LilyPond reserved words list.                                                                                                                                        |           |
| <b>ly:lily-lexer?</b> <i>x</i>                                                                                                                                                                                           | [Función] |
| Is <i>x</i> a Lily_lexer object?                                                                                                                                                                                         |           |
| <b>ly:lily-parser?</b> <i>x</i>                                                                                                                                                                                          | [Función] |
| Is <i>x</i> a Lily_parser object?                                                                                                                                                                                        |           |
| <b>ly:load-text-dimensions</b> <i>dimension-alist</i>                                                                                                                                                                    | [Función] |
| Load dimensions from T <sub>E</sub> X in a (KEY . (W H D)) format alist.                                                                                                                                                 |           |
| <b>ly:make-book</b> <i>paper header scores</i>                                                                                                                                                                           | [Función] |
| Make a \book of <i>paper</i> and <i>header</i> (which may be #f as well) containing \scores.                                                                                                                             |           |
| <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 <i>num</i> and <i>den</i> .                                                                                                                                                   |           |
| 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.                                                                     |           |

- ly:make-global-context** *output-def* [Función]  
 Set up a global interpretation context, using the output block *output-def*. 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.
- ly:make-listener** *callback* [Función]  
 Create a listener. Any time the listener hears an object, it will call *callback* with that object. *callback* should take exactly one argument.
- ly:make-moment** *n d gn gd* [Función]  
 Create the rational number with main timing *n/d*, and optional grace timing *gn/gd*.  
 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.
- 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.
- 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** is a list containing either **ly:music?** predicates or other type predicates.
- ly:make-output-def** [Función]  
 Make an output definition.
- ly:make-page-label-marker** *label* [Función]  
 Return page marker with 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 format* [Función]  
 Create an outputter that evaluates within *output-format*, writing to *port*.
- ly:make-pitch** *octave note alter* [Función]  
*octave* is specified by an integer, zero for the octave containing middle C. *note* is a number from 0 to 6, with 0 corresponding to pitch C and 6 corresponding to pitch B. *alter* is a rational number of whole tones for alteration.
- ly:make-prob** *type init rest* [Función]  
 Create a Prob object.
- ly:make-scale** *steps* [Función]  
 Create a scale. Takes a vector of integers as argument.
- ly:make-score** *music* [Función]  
 Return score with *music* encapsulated in *score*.

- ly:make-simple-closure** *expr* [Función]  
 Make a simple closure. *expr* should be form of *(func a1 A2 ...)*, and will be invoked as *(func delayed-arg a1 a2 ...)*.
- 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 *(1000 . -1000)* as its value), it is taken to be empty.
- ly:make-stream-event** *cl proplist* [Función]  
 Create a stream event of class *cl* with the given mutable property list.
- ly:message** *str rest* [Función]  
 A Scheme callable function to issue the message *str*. The message is formatted with **format** and *rest*.
- 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.
- 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-add** *a b* [Función]  
 Add two moments.
- ly:moment-div** *a b* [Función]  
 Divide two moments.
- ly:moment-grace-denominator** *mom* [Función]  
 Extract denominator from grace timing.
- ly:moment-grace-numerator** *mom* [Función]  
 Extract numerator from grace timing.
- ly:moment-main-denominator** *mom* [Función]  
 Extract denominator from main timing.
- ly:moment-main-numerator** *mom* [Función]  
 Extract numerator from main timing.
- ly:moment-mod** *a b* [Función]  
 Modulo of two moments.

|                                                                                                                                                                                                |           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <code>ly:moment-mul</code> <i>a b</i>                                                                                                                                                          | [Función] |
| Multiply two moments.                                                                                                                                                                          |           |
| <code>ly:moment-sub</code> <i>a b</i>                                                                                                                                                          | [Función] |
| Subtract two moments.                                                                                                                                                                          |           |
| <code>ly:moment&lt;?</code> <i>a b</i>                                                                                                                                                         | [Función] |
| Compare two moments.                                                                                                                                                                           |           |
| <code>ly:moment?</code> <i>x</i>                                                                                                                                                               | [Función] |
| Is <i>x</i> a <code>Moment</code> object?                                                                                                                                                      |           |
| <code>ly:music-compress</code> <i>m factor</i>                                                                                                                                                 | [Función] |
| Compress music object <i>m</i> by moment <i>factor</i> .                                                                                                                                       |           |
| <code>ly:music-deep-copy</code> <i>m</i>                                                                                                                                                       | [Función] |
| Copy <i>m</i> and all sub expressions of <i>m</i> .                                                                                                                                            |           |
| <code>ly:music-duration-compress</code> <i>mus fact</i>                                                                                                                                        | [Función] |
| Compress <i>mus</i> by factor <i>fact</i> , which is a <code>Moment</code> .                                                                                                                   |           |
| <code>ly:music-duration-length</code> <i>mus</i>                                                                                                                                               | [Función] |
| Extract the duration field from <i>mus</i> and return the length.                                                                                                                              |           |
| <code>ly:music-function-extract</code> <i>x</i>                                                                                                                                                | [Función] |
| Return the Scheme function inside <i>x</i> .                                                                                                                                                   |           |
| <code>ly:music-function?</code> <i>x</i>                                                                                                                                                       | [Función] |
| Is <i>x</i> a <code>music-function</code> ?                                                                                                                                                    |           |
| <code>ly:music-length</code> <i>mus</i>                                                                                                                                                        | [Función] |
| Get the length of music expression <i>mus</i> and return it as a <code>Moment</code> object.                                                                                                   |           |
| <code>ly:music-list?</code> <i>lst</i>                                                                                                                                                         | [Función] |
| Type predicate: Return true if <i>lst</i> is a list of music objects.                                                                                                                          |           |
| <code>ly:music-mutable-properties</code> <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 <code>make-music</code> function. |           |
| <code>ly:music-output?</code> <i>x</i>                                                                                                                                                         | [Función] |
| Is <i>x</i> a <code>Music_output</code> object?                                                                                                                                                |           |
| <code>ly:music-property</code> <i>mus sym dfault</i>                                                                                                                                           | [Función] |
| Get the property <i>sym</i> of music expression <i>mus</i> . If <i>sym</i> is undefined, return '().                                                                                           |           |
| <code>ly:music-set-property!</code> <i>mus sym val</i>                                                                                                                                         | [Función] |
| Set property <i>sym</i> in music expression <i>mus</i> to <i>val</i> .                                                                                                                         |           |
| <code>ly:music-transpose</code> <i>m p</i>                                                                                                                                                     | [Función] |
| Transpose <i>m</i> such that central C is mapped to <i>p</i> . Return <i>m</i> .                                                                                                               |           |
| <code>ly:music?</code> <i>obj</i>                                                                                                                                                              | [Función] |
| Type predicate.                                                                                                                                                                                |           |
| <code>ly:note-head::stem-attachment</code> <i>font-metric glyph-name</i>                                                                                                                       | [Función] |
| Get attachment in <i>font-metric</i> for attaching a stem to notehead <i>glyph-name</i> .                                                                                                      |           |



|                                                                                                                                                                        |           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <code>ly:number-&gt;string s</code>                                                                                                                                    | [Función] |
| Convert <i>num</i> to a string without generating many decimals.                                                                                                       |           |
| <code>ly:optimal-breaking pb</code>                                                                                                                                    | [Función] |
| Optimally break (pages and lines) the <code>Paper_book</code> object <i>pb</i> to minimize badness in both vertical and horizontal spacing.                            |           |
| <code>ly:option-usage</code>                                                                                                                                           | [Función] |
| Print <code>ly:set-option</code> usage.                                                                                                                                |           |
| <code>ly:otf-&gt;cff otf-file-name</code>                                                                                                                              | [Función] |
| Convert the contents of an OTF file to a CFF file, returning it as a string.                                                                                           |           |
| <code>ly:otf-font-glyph-info font glyph</code>                                                                                                                         | [Función] |
| Given the font metric <i>font</i> of an OpenType font, return the information about named glyph <i>glyph</i> (a string).                                               |           |
| <code>ly:otf-font-table-data font tag</code>                                                                                                                           | [Función] |
| Extract a table <i>tag</i> from <i>font</i> . Return empty string for non-existent <i>tag</i> .                                                                        |           |
| <code>ly:otf-font? font</code>                                                                                                                                         | [Función] |
| Is <i>font</i> an OpenType font?                                                                                                                                       |           |
| <code>ly:otf-glyph-list font</code>                                                                                                                                    | [Función] |
| Return a list of glyph names for <i>font</i> .                                                                                                                         |           |
| <code>ly:output-def-clone def</code>                                                                                                                                   | [Función] |
| Clone output definition <i>def</i> .                                                                                                                                   |           |
| <code>ly:output-def-lookup pap sym def</code>                                                                                                                          | [Función] |
| Look up <i>sym</i> in the <i>pap</i> output definition (e.g., <code>\paper</code> ). Return the value or <i>def</i> (which defaults to <code>()</code> ) if undefined. |           |
| <code>ly:output-def-parent def</code>                                                                                                                                  | [Función] |
| Get the parent output definition of <i>def</i> .                                                                                                                       |           |
| <code>ly:output-def-scope def</code>                                                                                                                                   | [Función] |
| Get the variable scope inside <i>def</i> .                                                                                                                             |           |
| <code>ly:output-def-set-variable! def sym val</code>                                                                                                                   | [Función] |
| Set an output definition <i>def</i> variable <i>sym</i> to <i>val</i> .                                                                                                |           |
| <code>ly:output-def? def</code>                                                                                                                                        | [Función] |
| Is <i>def</i> a layout definition?                                                                                                                                     |           |
| <code>ly:output-description output-def</code>                                                                                                                          | [Función] |
| Return the description of translators in <i>output-def</i> .                                                                                                           |           |
| <code>ly:output-formats</code>                                                                                                                                         | [Función] |
| Formats passed to <code>--format</code> as a list of strings, used for the output.                                                                                     |           |
| <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</code> <i>outputter str</i>                                                                                                | [Función] |
| Dump <i>str</i> onto <i>outputter</i> .                                                                                                                   |           |
| <code>ly:outputter-output-scheme</code> <i>outputter expr</i>                                                                                             | [Función] |
| Eval <i>expr</i> in module of <i>outputter</i> .                                                                                                          |           |
| <code>ly:outputter-port</code> <i>outputter</i>                                                                                                           | [Función] |
| Return output port for <i>outputter</i> .                                                                                                                 |           |
| <code>ly:page-marker?</code> <i>x</i>                                                                                                                     | [Función] |
| Is <i>x</i> a <code>Page_marker</code> object?                                                                                                            |           |
| <code>ly:page-turn-breaking</code> <i>pb</i>                                                                                                              | [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-physical-fonts</code> <i>f</i>                                                                                                        | [Función] |
| Return alist of (PSNAME . FILENAME) tuples.                                                                                                               |           |
| <code>ly:pango-font?</code> <i>f</i>                                                                                                                      | [Función] |
| Is <i>f</i> a pango font?                                                                                                                                 |           |
| <code>ly:paper-book-pages</code> <i>pb</i>                                                                                                                | [Función] |
| Return pages in book <i>pb</i> .                                                                                                                          |           |
| <code>ly:paper-book-paper</code> <i>pb</i>                                                                                                                | [Función] |
| Return pages in book <i>pb</i> .                                                                                                                          |           |
| <code>ly:paper-book-performances</code> <i>paper-book</i>                                                                                                 | [Función] |
| Return performances in book <i>paper-book</i> .                                                                                                           |           |
| <code>ly:paper-book-scopes</code> <i>book</i>                                                                                                             | [Función] |
| Return pages in layout book <i>book</i> .                                                                                                                 |           |
| <code>ly:paper-book-systems</code> <i>pb</i>                                                                                                              | [Función] |
| Return systems in book <i>pb</i> .                                                                                                                        |           |
| <code>ly:paper-book?</code> <i>x</i>                                                                                                                      | [Función] |
| Is <i>x</i> a <code>Paper_book</code> object?                                                                                                             |           |
| <code>ly:paper-fonts</code> <i>bp</i>                                                                                                                     | [Función] |
| Return fonts from the <code>\paper</code> block <i>bp</i> .                                                                                               |           |
| <code>ly:paper-get-font</code> <i>paper-smob chain</i>                                                                                                    | [Función] |
| Return a font metric satisfying the font-qualifiers in the alist chain <i>chain</i> . (An alist chain is a list of alists, containing grob properties.)   |           |
| <code>ly:paper-get-number</code> <i>layout-smob name</i>                                                                                                  | [Función] |
| Return the layout variable <i>name</i> .                                                                                                                  |           |
| <code>ly:paper-outputscales</code> <i>bp</i>                                                                                                              | [Función] |
| Get output-scale for <i>bp</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-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>ly:paper-system? obj</code>                                                                                                     | [Función] |
| Type predicate.                                                                                                                       |           |
| <code>ly:parse-file name</code>                                                                                                       | [Función] |
| Parse a single .ly file. Upon failure, throw <code>ly-file-failed</code> key.                                                         |           |
| <code>ly:parser-clear-error parser</code>                                                                                             | [Función] |
| Clear the error flag for the parser.                                                                                                  |           |
| <code>ly:parser-clone parser-smob</code>                                                                                              | [Función] |
| Return a clone of <i>parser-smob</i> .                                                                                                |           |
| <code>ly:parser-define! parser-smob symbol val</code>                                                                                 | [Función] |
| Bind <i>symbol</i> to <i>val</i> in <i>parser-smob</i> 's module.                                                                     |           |
| <code>ly:parser-error parser msg input</code>                                                                                         | [Función] |
| Display an error message and make the parser fail.                                                                                    |           |
| <code>ly:parser-has-error? parser</code>                                                                                              | [Función] |
| Does <i>parser</i> have an error flag?                                                                                                |           |
| <code>ly:parser-lexer parser-smob</code>                                                                                              | [Función] |
| Return the lexer for <i>parser-smob</i> .                                                                                             |           |
| <code>ly:parser-lookup parser-smob symbol</code>                                                                                      | [Función] |
| Look up <i>symbol</i> in <i>parser-smob</i> 's module. Return '()' if not defined.                                                    |           |
| <code>ly:parser-output-name parser</code>                                                                                             | [Función] |
| Return the base name of the output file.                                                                                              |           |
| <code>ly:parser-parse-string parser-smob ly-code</code>                                                                               | [Función] |
| Parse the string <i>ly-code</i> with <i>parser-smob</i> . Upon failure, throw <code>ly-file-failed</code> key.                        |           |
| <code>ly:parser-set-note-names parser names</code>                                                                                    | [Función] |
| Replace current note names in <i>parser</i> . <i>names</i> is an alist of symbols. This only has effect if the current mode is notes. |           |
| <code>ly:performance-write performance filename</code>                                                                                | [Función] |
| Write <i>performance</i> to <i>filename</i> .                                                                                         |           |
| <code>ly:pfb-&gt;pfa pfb-file-name</code>                                                                                             | [Función] |
| Convert the contents of a PFB file to PFA.                                                                                            |           |
| <code>ly:pitch-alteration pp</code>                                                                                                   | [Función] |
| Extract the alteration from pitch <i>pp</i> .                                                                                         |           |
| <code>ly:pitch-diff pitch root</code>                                                                                                 | [Función] |
| Return pitch <i>delta</i> such that <i>pitch</i> transposed by <i>delta</i> equals <i>root</i> .                                      |           |
| <code>ly:pitch-negate p</code>                                                                                                        | [Función] |
| Negate <i>p</i> .                                                                                                                     |           |
| <code>ly:pitch-notename pp</code>                                                                                                     | [Función] |
| Extract the note name from pitch <i>pp</i> .                                                                                          |           |
| <code>ly:pitch-octave pp</code>                                                                                                       | [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-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:pitch&lt;?</b> <i>p1 p2</i>                                                                                                   | [Función] |
| Is <i>p1</i> lexicographically smaller than <i>p2</i> ?                                                                             |           |
| <b>ly:pitch?</b> <i>x</i>                                                                                                           | [Función] |
| Is <i>x</i> a Pitch object?                                                                                                         |           |
| <b>ly:prob-immutable-properties</b> <i>prob</i>                                                                                     | [Función] |
| Retrieve an alist of mutable properties.                                                                                            |           |
| <b>ly:prob-mutable-properties</b> <i>prob</i>                                                                                       | [Función] |
| Retrieve an alist of mutable properties.                                                                                            |           |
| <b>ly:prob-property</b> <i>obj sym dfault</i>                                                                                       | [Función] |
| Return the value for <i>sym</i> .                                                                                                   |           |
| <b>ly:prob-property?</b> <i>obj sym</i>                                                                                             | [Función] |
| Is boolean prop <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:prob?</b> <i>x</i>                                                                                                            | [Función] |
| Is <i>x</i> a Prob object?                                                                                                          |           |
| <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 <b>format</b> 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 <b>format</b> 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 <b>prob</b> , <b>grob</b> , and <b>context</b> . |           |
| <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:register-stencil-expression</b> <i>symbol</i>                                                                                                                                                                                                                                                                                                                                                                                        | [Función] |
| Add <i>symbol</i> as head of a stencil expression.                                                                                                                                                                                                                                                                                                                                                                                         |           |
| <b>ly:relative-group-extent</b> <i>elements common axis</i>                                                                                                                                                                                                                                                                                                                                                                                | [Función] |
| Determine the extent of <i>elements</i> relative to <i>common</i> in the <i>axis</i> direction.                                                                                                                                                                                                                                                                                                                                            |           |
| <b>ly:reset-all-fonts</b>                                                                                                                                                                                                                                                                                                                                                                                                                  | [Función] |
| Forget all about previously loaded fonts.                                                                                                                                                                                                                                                                                                                                                                                                  |           |
| <b>ly:round-filled-box</b> <i>xext yext blot</i>                                                                                                                                                                                                                                                                                                                                                                                           | [Función] |
| Make a <b>Stencil</b> object that prints a black box of dimensions <i>xext</i> , <i>yext</i> and roundness <i>blot</i> .                                                                                                                                                                                                                                                                                                                   |           |
| <b>ly:round-filled-polygon</b> <i>points blot</i>                                                                                                                                                                                                                                                                                                                                                                                          | [Función] |
| Make a <b>Stencil</b> object that prints a black polygon with corners at the points defined by <i>points</i> (list of coordinate pairs) and roundness <i>blot</i> .                                                                                                                                                                                                                                                                        |           |
| <b>ly:run-translator</b> <i>mus output-def</i>                                                                                                                                                                                                                                                                                                                                                                                             | [Función] |
| Process <i>mus</i> according to <i>output-def</i> . An interpretation context is set up, and <i>mus</i> 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.                                                                                                                                                                                                                                                                                                                                           |           |
| <b>ly:score-add-output-def!</b> <i>score def</i>                                                                                                                                                                                                                                                                                                                                                                                           | [Función] |
| Add an output definition <i>def</i> to <i>score</i> .                                                                                                                                                                                                                                                                                                                                                                                      |           |
| <b>ly:score-embedded-format</b> <i>score layout</i>                                                                                                                                                                                                                                                                                                                                                                                        | [Función] |
| Run <i>score</i> through <i>layout</i> (an output definition) scaled to correct output-scale already, returning a list of layout-lines. This function takes an optional <b>Object_key</b> argument.                                                                                                                                                                                                                                        |           |
| <b>ly:score-error?</b> <i>score</i>                                                                                                                                                                                                                                                                                                                                                                                                        | [Función] |
| Was there an error in the score?                                                                                                                                                                                                                                                                                                                                                                                                           |           |
| <b>ly:score-header</b> <i>score</i>                                                                                                                                                                                                                                                                                                                                                                                                        | [Función] |
| Return score header.                                                                                                                                                                                                                                                                                                                                                                                                                       |           |
| <b>ly:score-music</b> <i>score</i>                                                                                                                                                                                                                                                                                                                                                                                                         | [Función] |
| Return score music.                                                                                                                                                                                                                                                                                                                                                                                                                        |           |
| <b>ly:score-output-defs</b> <i>score</i>                                                                                                                                                                                                                                                                                                                                                                                                   | [Función] |
| All output definitions in a score.                                                                                                                                                                                                                                                                                                                                                                                                         |           |
| <b>ly:score-set-header!</b> <i>score module</i>                                                                                                                                                                                                                                                                                                                                                                                            | [Función] |
| Set the score header.                                                                                                                                                                                                                                                                                                                                                                                                                      |           |
| <b>ly:score?</b> <i>x</i>                                                                                                                                                                                                                                                                                                                                                                                                                  | [Función] |
| Is <i>x</i> a <b>Score</b> object?                                                                                                                                                                                                                                                                                                                                                                                                         |           |
| <b>ly:set-default-scale</b> <i>scale</i>                                                                                                                                                                                                                                                                                                                                                                                                   | [Función] |
| Set the global default scale.                                                                                                                                                                                                                                                                                                                                                                                                              |           |
| <b>ly:set-grob-modification-callback</b> <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. |           |







|                                                                                                                                                                                                                                                                                                                                                                                                 |           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>ly:set-middle-C!</b> <i>context</i>                                                                                                                                                                                                                                                                                                                                                          | [Función] |
| Set the <code>middleCPosition</code> variable in <i>context</i> based on the variables <code>middleCClefPosition</code> and <code>middleCOffset</code> .                                                                                                                                                                                                                                        |           |
| <b>ly:set-option</b> <i>var val</i>                                                                                                                                                                                                                                                                                                                                                             | [Función] |
| Set a program option.                                                                                                                                                                                                                                                                                                                                                                           |           |
| <b>ly:set-point-and-click</b> <i>what</i>                                                                                                                                                                                                                                                                                                                                                       | [Función] |
| Deprecated.                                                                                                                                                                                                                                                                                                                                                                                     |           |
| <b>ly:set-property-cache-callback</b> <i>cb</i>                                                                                                                                                                                                                                                                                                                                                 | [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.                                                                              |           |
| <b>ly:simple-closure?</b> <i>clos</i>                                                                                                                                                                                                                                                                                                                                                           | [Función] |
| Type predicate.                                                                                                                                                                                                                                                                                                                                                                                 |           |
| <b>ly:skyline-pair?</b> <i>x</i>                                                                                                                                                                                                                                                                                                                                                                | [Función] |
| Is <i>x</i> a <code>Skyline_pair</code> object?                                                                                                                                                                                                                                                                                                                                                 |           |
| <b>ly:skyline?</b> <i>x</i>                                                                                                                                                                                                                                                                                                                                                                     | [Función] |
| Is <i>x</i> a <code>Skyline</code> object?                                                                                                                                                                                                                                                                                                                                                      |           |
| <b>ly:smob-protects</b>                                                                                                                                                                                                                                                                                                                                                                         | [Función] |
| Return LilyPond's internal smob protection list.                                                                                                                                                                                                                                                                                                                                                |           |
| <b>ly:solve-spring-rod-problem</b> <i>springs rods length ragged</i>                                                                                                                                                                                                                                                                                                                            | [Función] |
| Solve a spring and rod problem for <i>count</i> objects, that are connected by <i>count</i> -1 <i>springs</i> , and an arbitrary number of <i>rods</i> . <i>count</i> is implicitly given by <i>springs</i> and <i>rods</i> . The <i>springs</i> argument has the format ( <i>ideal</i> , <i>inverse_hook</i> ) and <i>rods</i> is of the form ( <i>idx1</i> , <i>idx2</i> , <i>distance</i> ). |           |
| <i>length</i> is a number, <i>ragged</i> a boolean.                                                                                                                                                                                                                                                                                                                                             |           |
| The function returns a list containing the force (positive for stretching, negative for compressing and <i>#f</i> for non-satisfied constraints) followed by <i>spring-count</i> +1 positions of the objects.                                                                                                                                                                                   |           |
| <b>ly:source-file?</b> <i>x</i>                                                                                                                                                                                                                                                                                                                                                                 | [Función] |
| Is <i>x</i> a <code>Source_file</code> object?                                                                                                                                                                                                                                                                                                                                                  |           |
| <b>ly:spanner-bound</b> <i>slur dir</i>                                                                                                                                                                                                                                                                                                                                                         | [Función] |
| Get one of the bounds of <i>slur</i> . <i>dir</i> is -1 for left, and 1 for right.                                                                                                                                                                                                                                                                                                              |           |
| <b>ly:spanner-broken-into</b> <i>spanner</i>                                                                                                                                                                                                                                                                                                                                                    | [Función] |
| Return broken-into list for <i>spanner</i> .                                                                                                                                                                                                                                                                                                                                                    |           |
| <b>ly:spanner?</b> <i>g</i>                                                                                                                                                                                                                                                                                                                                                                     | [Función] |
| Is <i>g</i> a spanner object?                                                                                                                                                                                                                                                                                                                                                                   |           |
| <b>ly:start-environment</b>                                                                                                                                                                                                                                                                                                                                                                     | [Función] |
| Return the environment (a list of strings) that was in effect at program start.                                                                                                                                                                                                                                                                                                                 |           |
| <b>ly:stderr-redirect</b> <i>file-name mode</i>                                                                                                                                                                                                                                                                                                                                                 | [Función] |
| Redirect stderr to <i>file-name</i> , opened with <i>mode</i> .                                                                                                                                                                                                                                                                                                                                 |           |

|                                                                                                                                                                                                                                                                                                                                                                                                           |           |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <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 minimum</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. If this puts the reference points closer than <i>minimum</i> , they are moved by the latter amount. <i>first</i> and <i>second</i> may also be '()' or #f. |           |
| <b>ly:stencil-empty?</b> <i>stil</i>                                                                                                                                                                                                                                                                                                                                                                      | [Función] |
| Return whether <i>stil</i> is empty.                                                                                                                                                                                                                                                                                                                                                                      |           |
| <b>ly:stencil-expr</b> <i>stil</i>                                                                                                                                                                                                                                                                                                                                                                        | [Función] |
| Return the expression of <i>stil</i> .                                                                                                                                                                                                                                                                                                                                                                    |           |
| <b>ly:stencil-extent</b> <i>stil axis</i>                                                                                                                                                                                                                                                                                                                                                                 | [Función] |
| Return a pair of numbers signifying the extent of <i>stil</i> in <i>axis</i> direction (0 or 1 for x and y axis, respectively).                                                                                                                                                                                                                                                                           |           |
| <b>ly:stencil-fonts</b> <i>s</i>                                                                                                                                                                                                                                                                                                                                                                          | [Función] |
| Analyze <i>s</i> , and return a list of fonts used in <i>s</i> .                                                                                                                                                                                                                                                                                                                                          |           |
| <b>ly:stencil-in-color</b> <i>stc r g b</i>                                                                                                                                                                                                                                                                                                                                                               | [Función] |
| Put <i>stc</i> in a different color.                                                                                                                                                                                                                                                                                                                                                                      |           |
| <b>ly:stencil-rotate</b> <i>stil angle x y</i>                                                                                                                                                                                                                                                                                                                                                            | [Función] |
| Return a stencil <i>stil</i> rotated <i>angle</i> degrees around point (x, y).                                                                                                                                                                                                                                                                                                                            |           |
| <b>ly:stencil-translate</b> <i>stil offset</i>                                                                                                                                                                                                                                                                                                                                                            | [Función] |
| Return a <i>stil</i> , but translated by <i>offset</i> (a pair of numbers).                                                                                                                                                                                                                                                                                                                               |           |
| <b>ly:stencil-translate-axis</b> <i>stil amount axis</i>                                                                                                                                                                                                                                                                                                                                                  | [Función] |
| Return a copy of <i>stil</i> but translated by <i>amount</i> in <i>axis</i> direction.                                                                                                                                                                                                                                                                                                                    |           |
| <b>ly:stencil?</b> <i>x</i>                                                                                                                                                                                                                                                                                                                                                                               | [Función] |
| Is <i>x</i> a Stencil object?                                                                                                                                                                                                                                                                                                                                                                             |           |
| <b>ly:stream-event?</b> <i>x</i>                                                                                                                                                                                                                                                                                                                                                                          | [Función] |
| Is <i>x</i> a Stream_event object?                                                                                                                                                                                                                                                                                                                                                                        |           |
| <b>ly:string-substitute</b> <i>a b s</i>                                                                                                                                                                                                                                                                                                                                                                  | [Función] |
| Replace string <i>a</i> by string <i>b</i> in string <i>s</i> .                                                                                                                                                                                                                                                                                                                                           |           |
| <b>ly:system-print</b> <i>system</i>                                                                                                                                                                                                                                                                                                                                                                      | [Función] |
| Draw the system and return the prob containing its stencil.                                                                                                                                                                                                                                                                                                                                               |           |
| <b>ly:system-stretch</b> <i>system amount-scm</i>                                                                                                                                                                                                                                                                                                                                                         | [Función] |
| Stretch the system vertically by the given amount. This must be called before the system is drawn (for example with <b>ly:system-print</b> ).                                                                                                                                                                                                                                                             |           |
| <b>ly:text-dimension</b> <i>font text</i>                                                                                                                                                                                                                                                                                                                                                                 | [Función] |
| Given the font metric in <i>font</i> and the string <i>text</i> , compute the extents of that text in that font. The return value is a pair of number-pairs.                                                                                                                                                                                                                                              |           |

- 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 a alist chain, ie. a list of alists. This is typically obtained with (ly:grob-alist-chain (ly:layout-lookup layout 'text-font-defaults)). *markup* is the markup text to be processed.
- ly:translator-description** *me* [Función]  
 Return an alist of properties of translator *me*.
- ly:translator-group?** *x* [Función]  
 Is *x* a Translator\_group object?
- ly:translator-name** *trans* [Función]  
 Return the type name of the translator object *trans*. The name is a symbol.
- ly:translator?** *x* [Función]  
 Is *x* a Translator object?
- ly:transpose-key-alist** *l* *pit* [Función]  
 Make a new key alist of *l* transposed by pitch *pit*.
- ly:truncate-list!** *lst* *i* [Función]  
 Take at most the first *i* of list *lst*.
- ly:ttf->pfa** *ttf-file-name* [Función]  
 Convert the contents of a TTF file to Type42 PFA, returning it as a string.
- ly:ttf-ps-name** *ttf-file-name* [Función]  
 Extract the PostScript name from a TrueType font.
- ly:unit** [Función]  
 Return the unit used for lengths as a string.
- ly:usage** [Función]  
 Print usage message.
- ly:version** [Función]  
 Return the current lilypond version as a list, e.g., (1 3 127 uu1).
- ly:warning** *str* *rest* [Función]  
 A Scheme callable function to issue the warning *str*. 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.



## Apéndice C 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 |  |
| <code>\key es \major</code>          | 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`

dinámica



`a\< a a\!`

crescendo



`a\> a a\!`

decrescendo



`< >`

acorde



`\partial 8`

anacrusa



`\times 2/3 {f g a}`

tresillos



`\grace`

mordentes



`\lyricmode { twinkle }`

escribir la letra

twinkle

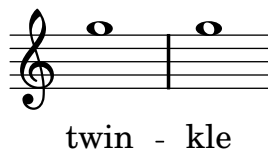
`\new Lyrics`

imprimir la letra

twinkle

`twin -- kle`

separador de sílabas



`\chordmode { c:dim f:maj7 }`

acordes



`\context ChordNames`

imprimir los nombres de los acordes

$C^{\circ} F^{\Delta}$

`<<{e f} \ \ {c d}>>`

polifonía



s4 s8 s16

silencios de separación

# Apéndice D Licencia de documentación libre de GNU

Version 1.1, March 2000

Copyright © 2000 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

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 written 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 that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The ‘Document’, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as ‘you’.

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. (For example, 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.

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 ‘Transparent’ copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and 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 has been designed to thwart or discourage subsequent modification by readers is not Transparent. 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 designed for human modification. Opaque formats include PostScript, PDF, 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 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.

## 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 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 publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. 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 less than five).
- 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', and 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. In any section entitled 'Acknowledgments' or 'Dedications', preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgments 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 as 'Endorsements' or to conflict in title with any Invariant Section.

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.

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 'Acknowledgments', 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, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an 'aggregate', and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they 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 quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around 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 provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 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 <http://www.gnu.org/copyleft/>.

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.

## ADDENDUM: cómo utilizar esta licencia para sus documentos

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 1.1
or any later version published by the Free Software Foundation;
with the Invariant Sections being list their titles, with the
Front-Cover Texts being list, and with the Back-Cover Texts being list.
A copy of the license is included in the section entitled 'GNU
Free Documentation License'
```

If you have no Invariant Sections, write 'with no Invariant Sections' instead of saying which ones are invariant. If you have no Front-Cover Texts, write 'no Front-Cover Texts' instead of 'Front-Cover Texts being *list*'; likewise for Back-Cover Texts.

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 E 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. Cada uno de los enlaces consta de dos partes. La primera parte apunta a la situación exacta del manual en que aparece la instrucción o palabra clave; la segunda parte apunta al comienzo de la sección correspondiente del manual en la que aparece la instrucción o palabra clave.

|                                                 |                             |
|-------------------------------------------------|-----------------------------|
| !                                               | ^                           |
| !..... 5                                        | ^..... 167                  |
| #                                               | -                           |
| #{set-accidental-style 'piano-cautionary)... 22 | -..... 131, 135             |
| ,                                               | \                           |
| '..... 1                                        | \!..... 66                  |
| (                                               | \(..... 71                  |
| (begin * * * *)..... 49                         | \)..... 71                  |
| (end * * * *)..... 49                           | \<..... 66                  |
| ,                                               | \>..... 66                  |
| ,                                               | \abs-fontsize..... 258      |
| ,                                               | \accepts..... 219           |
| ,                                               | \addlyrics..... 134         |
| ,                                               | \aeolian..... 14            |
| ,                                               | \afterGrace..... 60         |
| ,                                               | \aikenHeads..... 27         |
| ,                                               | \arpeggio..... 74, 75       |
| ,                                               | \arpeggioArrowDown..... 75  |
| ,                                               | \arpeggioArrowUp..... 75    |
| ,                                               | \arpeggioBracket..... 75    |
| ,                                               | \arpeggioNormal..... 75     |
| ,                                               | \arrow-head..... 280        |
| ,                                               | \ascendens..... 196         |
| ,                                               | \auctum..... 196            |
| ,                                               | \augmentum..... 196         |
| ,                                               | \autoBeamOff..... 52        |
| ,                                               | \autoBeamOn..... 52         |
| ,                                               | \backslashed-digit..... 290 |
| ,                                               | \bar..... 54                |
| ,                                               | \beam..... 280              |
| ,                                               | \bigger..... 258            |
| ,                                               | \bold..... 258              |
| ,                                               | \box..... 258               |
| ,                                               | \bracket..... 280           |
| ,                                               | \breathe..... 72            |
| ,                                               | \breve..... 30              |
| ,                                               | \cadenzaOff..... 46         |
| ,                                               | \cadenzaOn..... 46          |
| ,                                               | \caesura..... 188           |
| ,                                               | \caps..... 259              |
| ,                                               | \cavum..... 196             |
| ,                                               | \center-align..... 267      |
| ,                                               | \center-column..... 267     |
| ,                                               | \char..... 291              |
| ,                                               | \chordmode..... 4, 11       |
| ,                                               | \chords..... 169            |
| ,                                               | \circle..... 281            |
| ,                                               | \clef..... 11               |
| ,                                               | \column..... 267            |
| ,                                               | \column-lines..... 294      |

|                                    |          |                                         |          |
|------------------------------------|----------|-----------------------------------------|----------|
| <code>\combine</code>              | 268      | <code>\improvisationOff</code>          | 29       |
| <code>\concat</code>               | 268      | <code>\improvisationOn</code>           | 29       |
| <code>\context</code>              | 215      | <code>\inclinatum</code>                | 196      |
| <code>\cr</code>                   | 66       | <code>\ionian</code>                    | 14       |
| <code>\crescHairpin</code>         | 67       | <code>\italianChords</code>             | 172      |
| <code>\crescTextCresc</code>       | 67       | <code>\italic</code>                    | 260      |
| <code>\decr</code>                 | 66       | <code>\justified-lines</code>           | 295      |
| <code>\deminutum</code>            | 196      | <code>\justify</code>                   | 272      |
| <code>\denies</code>               | 219      | <code>\justify-field</code>             | 272      |
| <code>\descendens</code>           | 196      | <code>\justify-string</code>            | 273      |
| <code>\dimHairpin</code>           | 67       | <code>\key</code>                       | 14, 27   |
| <code>\dimTextDecr</code>          | 67       | <code>\label</code>                     | 204      |
| <code>\dimTextDecresc</code>       | 67       | <code>\laissezVibrer</code>             | 36       |
| <code>\dimTextDim</code>           | 67       | <code>\large</code>                     | 109, 260 |
| <code>\dir-column</code>           | 268      | <code>\larger</code>                    | 260      |
| <code>\divisioMaior</code>         | 188      | <code>\left-align</code>                | 273      |
| <code>\divisioMaxima</code>        | 188      | <code>\left-column</code>               | 274      |
| <code>\divisioMinima</code>        | 188      | <code>\line</code>                      | 274      |
| <code>\dorian</code>               | 14       | <code>\linea</code>                     | 196      |
| <code>\dotsDown</code>             | 31       | <code>\locrian</code>                   | 14       |
| <code>\dotsNeutral</code>          | 31       | <code>\longa</code>                     | 30       |
| <code>\dotsUp</code>               | 31       | <code>\lookup</code>                    | 291      |
| <code>\doubleflat</code>           | 284      | <code>\lower</code>                     | 274      |
| <code>\doublesharp</code>          | 284      | <code>\lydian</code>                    | 14       |
| <code>\draw-circle</code>          | 281      | <code>\lyricmode</code>                 | 131, 134 |
| <code>\draw-line</code>            | 281      | <code>\lyricsto</code>                  | 134      |
| <code>\dynamic</code>              | 259      | <code>\magnify</code>                   | 261      |
| <code>\dynamicDown</code>          | 67       | <code>\major</code>                     | 14       |
| <code>\dynamicNeutral</code>       | 67       | <code>\makeClusters</code>              | 85       |
| <code>\dynamicUp</code>            | 67       | <code>\mark</code>                      | 57, 120  |
| <code>\easyHeadsOff</code>         | 27       | <code>\markalphabet</code>              | 292      |
| <code>\easyHeadsOn</code>          | 27       | <code>\markletter</code>                | 292      |
| <code>\epsfile</code>              | 281      | <code>\markup</code>                    | 124      |
| <code>\f</code>                    | 66       | <code>\markuplines</code>               | 124, 127 |
| <code>\featherDurations</code>     | 53       | <code>\maxima</code>                    | 30       |
| <code>\ff</code>                   | 66       | <code>\medium</code>                    | 261      |
| <code>\fff</code>                  | 66       | <code>\melisma</code>                   | 137      |
| <code>\ffff</code>                 | 66       | <code>\melismaEnd</code>                | 137      |
| <code>\fill-line</code>            | 269      | <code>\mergeDifferentlyDottedOff</code> | 90       |
| <code>\filled-box</code>           | 282      | <code>\mergeDifferentlyDottedOn</code>  | 90       |
| <code>\finalis</code>              | 188      | <code>\mergeDifferentlyHeadedOff</code> | 90       |
| <code>\finger</code>               | 259      | <code>\mergeDifferentlyHeadedOn</code>  | 90       |
| <code>\flat</code>                 | 285      | <code>\mf</code>                        | 66       |
| <code>\flexa</code>                | 196      | <code>\minor</code>                     | 14       |
| <code>\fontCaps</code>             | 259      | <code>\mixolydian</code>                | 14       |
| <code>\fontsize</code>             | 259      | <code>\mp</code>                        | 66       |
| <code>\fp</code>                   | 66       | <code>\musicglyph</code>                | 285      |
| <code>\fraction</code>             | 291      | <code>\natural</code>                   | 285      |
| <code>\frenchChords</code>         | 172      | <code>\new</code>                       | 214      |
| <code>\fret-diagram</code>         | 288      | <code>\normal-size-sub</code>           | 261      |
| <code>\fret-diagram-terse</code>   | 288      | <code>\normal-size-super</code>         | 262      |
| <code>\fret-diagram-verbose</code> | 289      | <code>\normal-text</code>               | 262      |
| <code>\fromproperty</code>         | 291      | <code>\normalsize</code>                | 109, 262 |
| <code>\general-align</code>        | 269      | <code>\note</code>                      | 285      |
| <code>\germanChords</code>         | 172      | <code>\note-by-number</code>            | 285      |
| <code>\glissando</code>            | 73       | <code>\null</code>                      | 292      |
| <code>\grace</code>                | 59       | <code>\number</code>                    | 263      |
| <code>\halign</code>               | 270      | <code>\octaveCheck</code>               | 7        |
| <code>\harp-pedal</code>           | 290      | <code>\on-the-fly</code>                | 292      |
| <code>\hbracket</code>             | 282      | <code>\oneVoice</code>                  | 87       |
| <code>\hcenter-in</code>           | 271      | <code>\oriscus</code>                   | 196      |
| <code>\hideNotes</code>            | 111      | <code>\override</code>                  | 227, 292 |
| <code>\hideStaffSwitch</code>      | 148      | <code>\override-lines</code>            | 295      |
| <code>\hspace</code>               | 271      | <code>\p</code>                         | 66       |
| <code>\huge</code>                 | 109, 260 | <code>\pad-around</code>                | 274      |

|                                                   |          |                                       |          |
|---------------------------------------------------|----------|---------------------------------------|----------|
| <code>\pad-markup</code> .....                    | 275      | <code>\stemUp</code> .....            | 113      |
| <code>\pad-to-box</code> .....                    | 275      | <code>\stencil</code> .....           | 293      |
| <code>\pad-x</code> .....                         | 275      | <code>\stopTrillSpan</code> .....     | 77       |
| <code>\page-ref</code> .....                      | 204, 293 | <code>\strophas</code> .....          | 196      |
| <code>\partcombine</code> .....                   | 91       | <code>\strut</code> .....             | 293      |
| <code>\partial</code> .....                       | 45       | <code>\sub</code> .....               | 264      |
| <code>\pes</code> .....                           | 196      | <code>\super</code> .....             | 265      |
| <code>\phrasingSlurDown</code> .....              | 72       | <code>\table-of-contents</code> ..... | 206      |
| <code>\phrasingSlurNeutral</code> .....           | 72       | <code>\teeny</code> .....             | 109, 265 |
| <code>\phrasingSlurUp</code> .....                | 72       | <code>\tempo</code> .....             | 101      |
| <code>\phrygian</code> .....                      | 14       | <code>\text</code> .....              | 265      |
| <code>\postscript</code> .....                    | 282      | <code>\textLengthOff</code> .....     | 119      |
| <code>\pp</code> .....                            | 66       | <code>\textLengthOn</code> .....      | 119      |
| <code>\ppp</code> .....                           | 66       | <code>\tied-lyric</code> .....        | 288      |
| <code>\pppp</code> .....                          | 66       | <code>\tieDashed</code> .....         | 36       |
| <code>\property dentro de \lyricmode</code> ..... | 131      | <code>\tieDotted</code> .....         | 36       |
| <code>\put-adjacent</code> .....                  | 276      | <code>\tieDown</code> .....           | 36       |
| <code>\quilisma</code> .....                      | 196      | <code>\tieNeutral</code> .....        | 36       |
| <code>\raise</code> .....                         | 276      | <code>\tieSolid</code> .....          | 36       |
| <code>\relative</code> .....                      | 2, 4, 11 | <code>\tieUp</code> .....             | 36       |
| <code>\repeat</code> .....                        | 78       | <code>\time</code> .....              | 43       |
| <code>\repeatTie</code> .....                     | 36, 79   | <code>\times</code> .....             | 31       |
| <code>\rest</code> .....                          | 37       | <code>\tiny</code> .....              | 109, 265 |
| <code>\rfz</code> .....                           | 66       | <code>\tocItem</code> .....           | 206      |
| <code>\right-align</code> .....                   | 276      | <code>\translate</code> .....         | 277      |
| <code>\right-column</code> .....                  | 276      | <code>\translate-scaled</code> .....  | 277      |
| <code>\roman</code> .....                         | 263      | <code>\transparent</code> .....       | 293      |
| <code>\rotate</code> .....                        | 277      | <code>\transpose</code> .....         | 4, 8, 11 |
| <code>\rounded-box</code> .....                   | 283      | <code>\transposition</code> .....     | 17       |
| <code>\sacredHarpHeads</code> .....               | 27       | <code>\triangle</code> .....          | 284      |
| <code>\sans</code> .....                          | 263      | <code>\trill</code> .....             | 77       |
| <code>\scaleDurations</code> .....                | 34       | <code>\tupletDown</code> .....        | 31       |
| <code>\score</code> .....                         | 286      | <code>\tupletNeutral</code> .....     | 31       |
| <code>\semiflat</code> .....                      | 287      | <code>\tupletUp</code> .....          | 31       |
| <code>\semiGermanChords</code> .....              | 172      | <code>\tweak</code> .....             | 227      |
| <code>\semisharp</code> .....                     | 287      | <code>\typewriter</code> .....        | 266      |
| <code>\sesquiflat</code> .....                    | 287      | <code>\underline</code> .....         | 266      |
| <code>\sesquisharp</code> .....                   | 287      | <code>\unfoldRepeats</code> .....     | 209      |
| <code>\set</code> .....                           | 225      | <code>\unHideNotes</code> .....       | 111      |
| <code>\sf</code> .....                            | 66       | <code>\unset</code> .....             | 226      |
| <code>\sff</code> .....                           | 66       | <code>\upright</code> .....           | 266      |
| <code>\sfz</code> .....                           | 66       | <code>\vcenter</code> .....           | 278      |
| <code>\sharp</code> .....                         | 287      | <code>\verbatim-file</code> .....     | 293      |
| <code>\shiftOff</code> .....                      | 88, 90   | <code>\virga</code> .....             | 195      |
| <code>\shiftOn</code> .....                       | 88, 90   | <code>\virgula</code> .....           | 188      |
| <code>\shiftOnn</code> .....                      | 88, 90   | <code>\voiceFour</code> .....         | 87       |
| <code>\shiftOnnn</code> .....                     | 88, 90   | <code>\voiceFourStyle</code> .....    | 87       |
| <code>\showStaffSwitch</code> .....               | 148      | <code>\voiceNeutralStyle</code> ..... | 87       |
| <code>\simple</code> .....                        | 263      | <code>\voiceOne</code> .....          | 87       |
| <code>\skip</code> .....                          | 39       | <code>\voiceOneStyle</code> .....     | 87       |
| <code>\slashed-digit</code> .....                 | 293      | <code>\voiceThree</code> .....        | 87       |
| <code>\slurDashed</code> .....                    | 71       | <code>\voiceThreeStyle</code> .....   | 87       |
| <code>\slurDotted</code> .....                    | 71       | <code>\voiceTwo</code> .....          | 87       |
| <code>\slurDown</code> .....                      | 71       | <code>\voiceTwoStyle</code> .....     | 87       |
| <code>\slurNeutral</code> .....                   | 71       | <code>\whiteout</code> .....          | 294      |
| <code>\slurSolid</code> .....                     | 71       | <code>\with</code> .....              | 216      |
| <code>\slurUp</code> .....                        | 71       | <code>\with-color</code> .....        | 294      |
| <code>\small</code> .....                         | 109, 264 | <code>\with-dimensions</code> .....   | 294      |
| <code>\smallCaps</code> .....                     | 264      | <code>\with-url</code> .....          | 284      |
| <code>\smaller</code> .....                       | 264      | <code>\wordwrap</code> .....          | 279      |
| <code>\sp</code> .....                            | 66       | <code>\wordwrap-field</code> .....    | 278      |
| <code>\spp</code> .....                           | 66       | <code>\wordwrap-internal</code> ..... | 295      |
| <code>\startTrillSpan</code> .....                | 77       | <code>\wordwrap-lines</code> .....    | 295      |
| <code>\stemDown</code> .....                      | 113      | <code>\wordwrap-string</code> .....   | 279      |
| <code>\stemNeutral</code> .....                   | 113      |                                       |          |

\wordwrap-string-internal..... 295

|

|..... 56, 57

~

~..... 35

## A

arranger..... 199

aug..... 165

autoBeaming..... 51

autoBeamSettings..... 49

## B

barCheckSynchronize..... 56

barNumberVisibility..... 55

bookTitleMarkup..... 202

breakbefore..... 199

## C

chordNameExceptions..... 172

chordNameSeparator..... 171

chordNoteNamer..... 171

chordPrefixSpacer..... 172

chordRootNamer..... 171

composer..... 199

copyright..... 199

currentBarNumber..... 55

currentBarNumber..... 63

## D

dedicatoria..... 199

defaultBarType..... 55

dim..... 165

divisible..... 48

## E

evenFooterMarkup..... 203

evenHeaderMarkup..... 202

## F

followVoice..... 147

font-interface..... 128

forget accidental style..... 23

## I

instrument..... 199

## M

m..... 165

maj..... 165

major seven symbols..... 172

majorSevenSymbol..... 171

measureLength..... 63

measurePosition..... 63

meter..... 199

minimumFret..... 151

modern..... 21

modern-cautionary..... 21

modern-voice..... 21

modern-voice-cautionary..... 22

## N

no-reset accidental style..... 22

## O

oddFooterMarkup..... 203

oddHeaderMarkup..... 202

opus..... 199

## P

parallelMusic..... 95

piece..... 199

pipeSymbol..... 57

poet..... 199

printallheaders..... 202

## R

r..... 37

R..... 40

repeatCommands..... 55, 81

## S

s..... 39

scoreTitleMarkup..... 202

set-accidental-style..... 18

shapeNoteStyles..... 27

Staff.midiInstrument..... 208

stemLeftBeamCount..... 52

stemRightBeamCount..... 52

subdivideBeams..... 48

subsubtitle..... 199

subtitle..... 199

suggestAccidentals..... 197

sus..... 167

## T

tagline..... 199

textSpannerDown..... 120

textSpannerNeutral..... 120

textSpannerUp..... 120

title..... 199

tremoloFlags..... 84

TupletNumber..... 32

tupletNumberFormatFunction..... 32

tupletSpannerDuration..... 32

## W

whichBar..... 55

## Apéndice F 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. Cada uno de los enlaces consta de dos partes. La primera parte apunta a la situación exacta del manual en que aparece el término; la segunda parte apunta al comienzo de la sección correspondiente del manual en la que se discute dicho término.

|                                                 |                              |
|-------------------------------------------------|------------------------------|
| !                                               | ]                            |
| !..... 5                                        | ] ..... 52                   |
| #                                               | ^                            |
| #{set-accidental-style 'piano-cautionary)... 22 | ^ ..... 167                  |
| ,                                               | -                            |
| ' ..... 1                                       | - ..... 131, 135             |
| (                                               | \                            |
| (begin * * * *) ..... 49                        | \! ..... 66                  |
| (end * * * *) ..... 49                          | \( ..... 71                  |
| ,                                               | \) ..... 71                  |
| ,                                               | \< ..... 66                  |
| .                                               | \> ..... 66                  |
| .                                               | \abs-fontsize ..... 258      |
| .                                               | \accepts ..... 219           |
| .                                               | \addlyrics ..... 130         |
| .                                               | \addlyrics ..... 134         |
| .                                               | \aeolian ..... 14            |
| .                                               | \afterGrace ..... 60         |
| .                                               | \aikenHeads ..... 27         |
| .                                               | \arpeggio ..... 74, 75       |
| .                                               | \arpeggioArrowDown ..... 75  |
| .                                               | \arpeggioArrowUp ..... 75    |
| .                                               | \arpeggioBracket ..... 75    |
| .                                               | \arpeggioNormal ..... 75     |
| .                                               | \arrow-head ..... 280        |
| .                                               | \ascendens ..... 196         |
| .                                               | \auctum ..... 196            |
| .                                               | \augmentum ..... 196         |
| .                                               | \autoBeamOff ..... 52        |
| .                                               | \autoBeamOn ..... 52         |
| .                                               | \backslashed-digit ..... 290 |
| .                                               | \bar ..... 54                |
| .                                               | \beam ..... 280              |
| .                                               | \bigger ..... 258            |
| .                                               | \bold ..... 258              |
| .                                               | \box ..... 258               |
| .                                               | \bracket ..... 280           |
| .                                               | \breathe ..... 72            |
| .                                               | \breve ..... 30              |
| .                                               | \cadenzaOff ..... 46         |
| .                                               | \cadenzaOn ..... 46          |
| .                                               | \caesura ..... 188           |
| .                                               | \caps ..... 259              |
| .                                               | \cavum ..... 196             |
| .                                               | \center-align ..... 267      |
| .                                               | \center-column ..... 267     |

|                                          |       |                                               |          |
|------------------------------------------|-------|-----------------------------------------------|----------|
| <code>\char</code> .....                 | 291   | <code>\harp-pedal</code> .....                | 290      |
| <code>\chordmode</code> .....            | 4, 11 | <code>\hbracket</code> .....                  | 282      |
| <code>\chords</code> .....               | 169   | <code>\hcenter-in</code> .....                | 271      |
| <code>\circle</code> .....               | 281   | <code>\hideNotes</code> .....                 | 111      |
| <code>\clef</code> .....                 | 11    | <code>\hideStaffSwitch</code> .....           | 148      |
| <code>\column</code> .....               | 267   | <code>\hspace</code> .....                    | 271      |
| <code>\column-lines</code> .....         | 294   | <code>\huge</code> .....                      | 109, 260 |
| <code>\combine</code> .....              | 268   | <code>\improvisationOff</code> .....          | 29       |
| <code>\concat</code> .....               | 268   | <code>\improvisationOn</code> .....           | 29       |
| <code>\context</code> .....              | 215   | <code>\inclinatum</code> .....                | 196      |
| <code>\cr</code> .....                   | 66    | <code>\ionian</code> .....                    | 14       |
| <code>\crescHairpin</code> .....         | 67    | <code>\italianChords</code> .....             | 172      |
| <code>\crescTextCresc</code> .....       | 67    | <code>\italic</code> .....                    | 260      |
| <code>\decr</code> .....                 | 66    | <code>\justified-lines</code> .....           | 295      |
| <code>\deminutum</code> .....            | 196   | <code>\justify</code> .....                   | 272      |
| <code>\denies</code> .....               | 219   | <code>\justify-field</code> .....             | 272      |
| <code>\descendens</code> .....           | 196   | <code>\justify-string</code> .....            | 273      |
| <code>\dimHairpin</code> .....           | 67    | <code>\key</code> .....                       | 14, 27   |
| <code>\dimTextDecr</code> .....          | 67    | <code>\label</code> .....                     | 204      |
| <code>\dimTextDecresc</code> .....       | 67    | <code>\laissezVibrer</code> .....             | 36       |
| <code>\dimTextDim</code> .....           | 67    | <code>\large</code> .....                     | 109, 260 |
| <code>\dir-column</code> .....           | 268   | <code>\larger</code> .....                    | 260      |
| <code>\divisioMaior</code> .....         | 188   | <code>\left-align</code> .....                | 273      |
| <code>\divisioMaxima</code> .....        | 188   | <code>\left-column</code> .....               | 274      |
| <code>\divisioMinima</code> .....        | 188   | <code>\line</code> .....                      | 274      |
| <code>\dorian</code> .....               | 14    | <code>\linea</code> .....                     | 196      |
| <code>\dotsDown</code> .....             | 31    | <code>\locrian</code> .....                   | 14       |
| <code>\dotsNeutral</code> .....          | 31    | <code>\longa</code> .....                     | 30       |
| <code>\dotsUp</code> .....               | 31    | <code>\lookup</code> .....                    | 291      |
| <code>\doubleflat</code> .....           | 284   | <code>\lower</code> .....                     | 274      |
| <code>\doublesharp</code> .....          | 284   | <code>\lydian</code> .....                    | 14       |
| <code>\draw-circle</code> .....          | 281   | <code>\lyricmode</code> .....                 | 131, 134 |
| <code>\draw-line</code> .....            | 281   | <code>\lyricsto</code> .....                  | 134      |
| <code>\dynamic</code> .....              | 259   | <code>\magnify</code> .....                   | 261      |
| <code>\dynamicDown</code> .....          | 67    | <code>\major</code> .....                     | 14       |
| <code>\dynamicNeutral</code> .....       | 67    | <code>\makeClusters</code> .....              | 85       |
| <code>\dynamicUp</code> .....            | 67    | <code>\mark</code> .....                      | 57, 120  |
| <code>\easyHeadsOff</code> .....         | 27    | <code>\markalphabet</code> .....              | 292      |
| <code>\easyHeadsOn</code> .....          | 27    | <code>\markletter</code> .....                | 292      |
| <code>\epsfile</code> .....              | 281   | <code>\markup</code> .....                    | 124      |
| <code>\f</code> .....                    | 66    | <code>\markuplines</code> .....               | 124, 127 |
| <code>\featherDurations</code> .....     | 53    | <code>\maxima</code> .....                    | 30       |
| <code>\ff</code> .....                   | 66    | <code>\medium</code> .....                    | 261      |
| <code>\fff</code> .....                  | 66    | <code>\melisma</code> .....                   | 137      |
| <code>\ffff</code> .....                 | 66    | <code>\melismaEnd</code> .....                | 137      |
| <code>\fill-line</code> .....            | 269   | <code>\mergeDifferentlyDottedOff</code> ..... | 90       |
| <code>\filled-box</code> .....           | 282   | <code>\mergeDifferentlyDottedOn</code> .....  | 90       |
| <code>\finalis</code> .....              | 188   | <code>\mergeDifferentlyHeadedOff</code> ..... | 90       |
| <code>\finger</code> .....               | 259   | <code>\mergeDifferentlyHeadedOn</code> .....  | 90       |
| <code>\flat</code> .....                 | 285   | <code>\mf</code> .....                        | 66       |
| <code>\flexa</code> .....                | 196   | <code>\minor</code> .....                     | 14       |
| <code>\fontCaps</code> .....             | 259   | <code>\mixolydian</code> .....                | 14       |
| <code>\fontsize</code> .....             | 259   | <code>\mp</code> .....                        | 66       |
| <code>\fp</code> .....                   | 66    | <code>\musicglyph</code> .....                | 285      |
| <code>\fraction</code> .....             | 291   | <code>\natural</code> .....                   | 285      |
| <code>\frenchChords</code> .....         | 172   | <code>\new</code> .....                       | 214      |
| <code>\fret-diagram</code> .....         | 288   | <code>\normal-size-sub</code> .....           | 261      |
| <code>\fret-diagram-terse</code> .....   | 288   | <code>\normal-size-super</code> .....         | 262      |
| <code>\fret-diagram-verbose</code> ..... | 289   | <code>\normal-text</code> .....               | 262      |
| <code>\fromproperty</code> .....         | 291   | <code>\normalsize</code> .....                | 109, 262 |
| <code>\general-align</code> .....        | 269   | <code>\note</code> .....                      | 285      |
| <code>\germanChords</code> .....         | 172   | <code>\note-by-number</code> .....            | 285      |
| <code>\glissando</code> .....            | 73    | <code>\null</code> .....                      | 292      |
| <code>\grace</code> .....                | 59    | <code>\number</code> .....                    | 263      |
| <code>\halign</code> .....               | 270   | <code>\octaveCheck</code> .....               | 7        |



|                                                   |          |                                       |          |
|---------------------------------------------------|----------|---------------------------------------|----------|
| <code>\on-the-fly</code> .....                    | 292      | <code>\slurUp</code> .....            | 71       |
| <code>\once</code> .....                          | 226      | <code>\small</code> .....             | 109, 264 |
| <code>\oneVoice</code> .....                      | 87       | <code>\smallCaps</code> .....         | 264      |
| <code>\oriscus</code> .....                       | 196      | <code>\smaller</code> .....           | 264      |
| <code>\override</code> .....                      | 227, 292 | <code>\sp</code> .....                | 66       |
| <code>\override-lines</code> .....                | 295      | <code>\spp</code> .....               | 66       |
| <code>\p</code> .....                             | 66       | <code>\startTrillSpan</code> .....    | 77       |
| <code>\pad-around</code> .....                    | 274      | <code>\stemDown</code> .....          | 113      |
| <code>\pad-markup</code> .....                    | 275      | <code>\stemNeutral</code> .....       | 113      |
| <code>\pad-to-box</code> .....                    | 275      | <code>\stemUp</code> .....            | 113      |
| <code>\pad-x</code> .....                         | 275      | <code>\stencil</code> .....           | 293      |
| <code>\page-ref</code> .....                      | 204, 293 | <code>\stopTrillSpan</code> .....     | 77       |
| <code>\paper</code> .....                         | 203      | <code>\strophæ</code> .....           | 196      |
| <code>\partcombine</code> .....                   | 91       | <code>\strut</code> .....             | 293      |
| <code>\partial</code> .....                       | 45       | <code>\sub</code> .....               | 264      |
| <code>\pes</code> .....                           | 196      | <code>\super</code> .....             | 265      |
| <code>\phrasingSlurDown</code> .....              | 72       | <code>\table-of-contents</code> ..... | 206      |
| <code>\phrasingSlurNeutral</code> .....           | 72       | <code>\teeny</code> .....             | 109, 265 |
| <code>\phrasingSlurUp</code> .....                | 72       | <code>\tempo</code> .....             | 101      |
| <code>\phrygian</code> .....                      | 14       | <code>\text</code> .....              | 265      |
| <code>\postscript</code> .....                    | 282      | <code>\textLengthOff</code> .....     | 119      |
| <code>\pp</code> .....                            | 66       | <code>\textLengthOn</code> .....      | 119      |
| <code>\ppp</code> .....                           | 66       | <code>\tied-lyric</code> .....        | 288      |
| <code>\pppp</code> .....                          | 66       | <code>\tieDashed</code> .....         | 36       |
| <code>\property dentro de \lyricmode</code> ..... | 131      | <code>\tieDotted</code> .....         | 36       |
| <code>\put-adjacent</code> .....                  | 276      | <code>\tieDown</code> .....           | 36       |
| <code>\quilisma</code> .....                      | 196      | <code>\tieNeutral</code> .....        | 36       |
| <code>\raise</code> .....                         | 276      | <code>\tieSolid</code> .....          | 36       |
| <code>\relative</code> .....                      | 2, 4, 11 | <code>\tieUp</code> .....             | 36       |
| <code>\repeat</code> .....                        | 78       | <code>\time</code> .....              | 43       |
| <code>\repeatTie</code> .....                     | 36, 79   | <code>\times</code> .....             | 31       |
| <code>\rest</code> .....                          | 37       | <code>\tiny</code> .....              | 109, 265 |
| <code>\rfz</code> .....                           | 66       | <code>\tocItem</code> .....           | 206      |
| <code>\right-align</code> .....                   | 276      | <code>\translate</code> .....         | 277      |
| <code>\right-column</code> .....                  | 276      | <code>\translate-scaled</code> .....  | 277      |
| <code>\roman</code> .....                         | 263      | <code>\transparent</code> .....       | 293      |
| <code>\rotate</code> .....                        | 277      | <code>\transpose</code> .....         | 4, 8, 11 |
| <code>\rounded-box</code> .....                   | 283      | <code>\transposition</code> .....     | 17       |
| <code>\sacredHarpHeads</code> .....               | 27       | <code>\triangle</code> .....          | 284      |
| <code>\sans</code> .....                          | 263      | <code>\trill</code> .....             | 77       |
| <code>\scaleDurations</code> .....                | 34       | <code>\tupletDown</code> .....        | 31       |
| <code>\score</code> .....                         | 286      | <code>\tupletNeutral</code> .....     | 31       |
| <code>\semiflat</code> .....                      | 287      | <code>\tupletUp</code> .....          | 31       |
| <code>\semiGermanChords</code> .....              | 172      | <code>\tweak</code> .....             | 227      |
| <code>\semisharp</code> .....                     | 287      | <code>\typewriter</code> .....        | 266      |
| <code>\sesquiflat</code> .....                    | 287      | <code>\underline</code> .....         | 266      |
| <code>\sesquisharp</code> .....                   | 287      | <code>\unfoldRepeats</code> .....     | 209      |
| <code>\set</code> .....                           | 225      | <code>\unHideNotes</code> .....       | 111      |
| <code>\sf</code> .....                            | 66       | <code>\unset</code> .....             | 226      |
| <code>\sff</code> .....                           | 66       | <code>\upright</code> .....           | 266      |
| <code>\sfz</code> .....                           | 66       | <code>\vcenter</code> .....           | 278      |
| <code>\sharp</code> .....                         | 287      | <code>\verbatim-file</code> .....     | 293      |
| <code>\shiftOff</code> .....                      | 88, 90   | <code>\virga</code> .....             | 195      |
| <code>\shiftOn</code> .....                       | 88, 90   | <code>\virgula</code> .....           | 188      |
| <code>\shiftOnn</code> .....                      | 88, 90   | <code>\voiceFour</code> .....         | 87       |
| <code>\shiftOnnn</code> .....                     | 88, 90   | <code>\voiceFourStyle</code> .....    | 87       |
| <code>\showStaffSwitch</code> .....               | 148      | <code>\voiceNeutralStyle</code> ..... | 87       |
| <code>\simple</code> .....                        | 263      | <code>\voiceOne</code> .....          | 87       |
| <code>\skip</code> .....                          | 39       | <code>\voiceOneStyle</code> .....     | 87       |
| <code>\slashed-digit</code> .....                 | 293      | <code>\voiceThree</code> .....        | 87       |
| <code>\slurDashed</code> .....                    | 71       | <code>\voiceThreeStyle</code> .....   | 87       |
| <code>\slurDotted</code> .....                    | 71       | <code>\voiceTwo</code> .....          | 87       |
| <code>\slurDown</code> .....                      | 71       | <code>\voiceTwoStyle</code> .....     | 87       |
| <code>\slurNeutral</code> .....                   | 71       | <code>\whiteout</code> .....          | 294      |
| <code>\slurSolid</code> .....                     | 71       | <code>\with</code> .....              | 216      |

|                                              |     |
|----------------------------------------------|-----|
| <code>\with-color</code> .....               | 294 |
| <code>\with-dimensions</code> .....          | 294 |
| <code>\with-url</code> .....                 | 284 |
| <code>\wordwrap</code> .....                 | 279 |
| <code>\wordwrap-field</code> .....           | 278 |
| <code>\wordwrap-internal</code> .....        | 295 |
| <code>\wordwrap-lines</code> .....           | 295 |
| <code>\wordwrap-string</code> .....          | 279 |
| <code>\wordwrap-string-internal</code> ..... | 295 |

|       |        |
|-------|--------|
| ..... | 56, 57 |
|-------|--------|

|         |    |
|---------|----|
| ~ ..... | 35 |
|---------|----|

|         |    |
|---------|----|
| 1 ..... | 16 |
|---------|----|

|         |    |
|---------|----|
| 8 ..... | 16 |
|---------|----|

|                                              |              |
|----------------------------------------------|--------------|
| <b>A</b>                                     |              |
| a due .....                                  | 94           |
| absoluta .....                               | 1            |
| absoluta, especificación, de la octava ..... | 1            |
| absolutos, matices .....                     | 66           |
| accent .....                                 | 295          |
| acciaccatura .....                           | 59           |
| acciaccatura .....                           | 235, 319     |
| accidental .....                             | 4            |
| Accidental .....                             | 6, 24, 181   |
| Accidental, musica ficta .....               | 197          |
| accidental-interface .....                   | 6            |
| accidental-suggestion-interface .....        | 24           |
| Accidental_engraver .....                    | 6, 24, 198   |
| AccidentalCautionary .....                   | 6            |
| AccidentalPlacement .....                    | 24           |
| Accidentals and key signatures .....         | 4, 6, 14, 16 |
| AccidentalSuggestion .....                   | 24, 198      |
| acento .....                                 | 64           |
| acento .....                                 | 66           |
| acento .....                                 | 295          |
| acorde .....                                 | 85, 164, 170 |
| acorde quebrado .....                        | 74           |
| acordes .....                                | 85, 168      |
| acordes alterados .....                      | 165          |
| acordes de jazz .....                        | 163, 171     |
| acordes extendidos .....                     | 165          |
| acordes, alteración de notas .....           | 166          |
| acordes, alteraciones dentro de .....        | 24           |
| acordes, bajo de .....                       | 167          |
| acordes, diagramas de .....                  | 153          |
| acordes, inversiones .....                   | 167          |
| acordes, modo de .....                       | 163          |
| acordes, nombres de .....                    | 163, 168     |
| acordes, supresiones en .....                | 167          |
| acordes, variantes de .....                  | 165          |
| acortar compases .....                       | 45           |

|                                                      |          |
|------------------------------------------------------|----------|
| addChordShape .....                                  | 235, 319 |
| adding a white background to text .....              | 294      |
| addInstrumentDefinition .....                        | 235, 319 |
| addQuote .....                                       | 235, 319 |
| adiciones en acordes .....                           | 166      |
| adorno, notas de, dentro de corchetes de grupo       |          |
| especial .....                                       | 34       |
| afterGrace .....                                     | 235, 319 |
| Aiken, cabezas de nota con forma de .....            | 27       |
| ajuste del símbolo del pentagrama .....              | 100      |
| ajuste fino de las barras automáticas .....          | 49       |
| al niente .....                                      | 68       |
| alignAboveContext .....                              | 220      |
| alignBelowContext .....                              | 220      |
| All layout objects .....                             | 225      |
| allowPageTurn .....                                  | 235, 319 |
| alteración con paréntesis .....                      | 5        |
| alteración de precaución .....                       | 5        |
| alteración de voz de precaución moderno, estilo de   |          |
| .....                                                | 22       |
| alteración recordatoria .....                        | 5        |
| alteración, estilo de voz de precaución moderno ...  | 22       |
| alteración, estilo de, olvidar .....                 | 23       |
| alteración, estilo moderno de .....                  | 21       |
| alteración, moderno de precaución, voz, estilo de .. | 22       |
| alteraciones accidentales .....                      | 18, 181  |
| alteraciones automáticas .....                       | 18       |
| alteraciones de piano .....                          | 22       |
| alteraciones en acordes .....                        | 24       |
| alteraciones modernas .....                          | 21       |
| alteraciones no restablecer, estilo de .....         | 22       |
| alteraciones y notas simultáneas .....               | 24       |
| alteraciones, en varias voces .....                  | 21       |
| alteraciones, estilo de las .....                    | 18       |
| alteraciones, estilo moderno .....                   | 21       |
| alteraciones, estilo moderno de precaución .....     | 21       |
| alteraciones, estilo predeterminado de las .....     | 18       |
| alteraciones, piano de precaución .....              | 22       |
| alteraciones, piano de precaución, estilo de .....   | 22       |
| alto, clave de .....                                 | 11       |
| altura de concierto .....                            | 18       |
| alturas .....                                        | 1        |
| ambitus .....                                        | 24       |
| Ambitus .....                                        | 25       |
| ambitus-interface .....                              | 25       |
| Ambitus_engraver .....                               | 25       |
| AmbitusAccidental .....                              | 25       |
| AmbitusLine .....                                    | 25       |
| AmbitusNoteHead .....                                | 25       |
| anacrusa .....                                       | 45       |
| antiguas, claves .....                               | 11       |
| apagada .....                                        | 64       |
| apagado .....                                        | 295      |
| applyContext .....                                   | 235, 319 |
| applyMusic .....                                     | 235, 319 |
| applyOutput .....                                    | 235, 319 |
| appoggiatura .....                                   | 59       |
| appoggiatura .....                                   | 235, 319 |
| arco abajo .....                                     | 64, 295  |
| arco arriba .....                                    | 64, 295  |
| Armadura de la tonalidad .....                       | 14       |
| armónicos artificiales .....                         | 150      |
| armónicos, cabezas de nota .....                     | 26       |
| arpa sacra, cabezas de nota de .....                 | 27       |
| arpeggio .....                                       | 74       |

|                                   |          |
|-----------------------------------|----------|
| Arpeggio                          | 77       |
| arpeggio                          | 77       |
| arranger                          | 199      |
| articulaciones                    | 64, 186  |
| Articulation and dynamics         | 69       |
| aspas, cabezas de nota            | 26       |
| assertBeamQuant                   | 235, 319 |
| assertBeamSlope                   | 235, 319 |
| aug                               | 165      |
| auto-barrado                      | 51       |
| auto-knee-gap                     | 49       |
| autoBeaming                       | 51       |
| autoBeamOff                       | 47       |
| autoBeamOn                        | 47       |
| autoBeamSettings                  | 49       |
| autochange                        | 235, 319 |
| AutoChangeMusic                   | 147      |
| automática, combinación de partes | 91       |
| automática, generación de barras  | 51       |
| automáticas, alteraciones         | 18       |

## B

|                                   |          |
|-----------------------------------|----------|
| Backend                           | 221, 225 |
| backslashed digits                | 290      |
| bajo cifrado, alineación          | 179      |
| bajo cifrado, líneas de extensión | 177      |
| bajo de los acordes               | 167      |
| bajo, clave de                    | 11       |
| balloon-interface                 | 115      |
| Balloon_engraver                  | 115      |
| balloonGrobText                   | 235, 319 |
| balloonText                       | 236, 319 |
| BalloonTextItem                   | 115      |
| Banjo, tablaturas de              | 156      |
| Banter                            | 171      |
| bar                               | 236, 319 |
| bar numbers, regular spacing      | 55       |
| bar-line-interface                | 305      |
| Bar_engraver                      | 170      |
| barCheckSynchronize               | 56       |
| barítono, clave de                | 11       |
| BarLine                           | 55       |
| BarNumber                         | 56       |
| barNumberCheck                    | 236, 319 |
| barNumberVisibility               | 55       |
| barradas, cabezas de nota         | 28       |
| barras automáticas, ajuste fino   | 49       |
| Barras de compás                  | 54       |
| barras de trémolo                 | 83       |
| barras en ángulo                  | 49       |
| barras manuales                   | 47, 52   |
| barras progresivas                | 53       |
| barras y saltos de línea          | 49       |
| Bass, figured                     | 174      |
| Bass, thorough                    | 174      |
| BassFigure                        | 177, 179 |
| BassFigureAlignment               | 177, 179 |
| BassFigureBracket                 | 177, 179 |
| BassFigureContinuation            | 177, 179 |
| BassFigureLine                    | 177, 179 |
| Basso continuo                    | 174      |
| batería                           | 157, 158 |
| Beam                              | 49, 84   |

|                         |          |
|-------------------------|----------|
| becuadro                | 4        |
| bemol                   | 4        |
| bemol                   | 6        |
| bemol, doble            | 4        |
| bendAfter               | 236, 319 |
| bloque MIDI             | 208      |
| bookTitleMarkup         | 202      |
| breakbefore             | 199      |
| breathe                 | 236, 320 |
| BreathingSign           | 72, 188  |
| breve                   | 31       |
| buscar objetos gráficos | 227      |

## C

|                                          |          |
|------------------------------------------|----------|
| cabezas de nota antiguas                 | 181      |
| cadencia                                 | 46       |
| calderón                                 | 64, 295  |
| calderón sobre la línea divisoria        | 120      |
| calderón sobre silencios multicompas     | 41       |
| cambiar propiedades                      | 225      |
| Cambio automático de pentagrama          | 146      |
| cambio de dedo                           | 109      |
| cantante, nombre del                     | 141      |
| caracteres especiales en modo de marcado | 125      |
| centering a column of text               | 267      |
| changing direction of text columns       | 268      |
| ChoirStaff                               | 98       |
| choral, clave de tenor                   | 12       |
| chordNameExceptions                      | 172      |
| ChordNames                               | 100, 170 |
| chordNameSeparator                       | 171      |
| chordNoteNamer                           | 171      |
| chordPrefixSpacer                        | 172      |
| chordRootNamer                           | 171      |
| chords and ties                          | 35       |
| chords, fingering                        | 110      |
| cifrado americano                        | 163      |
| circling text                            | 281      |
| clave                                    | 11       |
| clave antigua                            | 11       |
| clave de alto                            | 11       |
| clave de bajo                            | 11       |
| clave de barítono                        | 11       |
| clave de contrabajo                      | 11       |
| clave de Do                              | 11       |
| clave de Fa                              | 11       |
| clave de mezzosoprano                    | 11       |
| clave de Sol                             | 11       |
| clave de soprano                         | 11       |
| clave de tenor                           | 11       |
| clave de violín                          | 11       |
| clave francesa                           | 11       |
| clave transpositora                      | 12       |
| claves                                   | 182      |
| clef                                     | 4        |
| clef                                     | 236, 320 |
| Clef                                     | 14       |
| clef-interface                           | 14       |
| Clef_engraver                            | 14       |
| cluster                                  | 85       |
| Cluster_spanner_engraver                 | 85       |
| ClusterSpanner                           | 85       |
| ClusterSpannerBeacon                     | 85       |

|                                              |             |
|----------------------------------------------|-------------|
| coda.....                                    | 58, 64, 295 |
| coda sobre la línea divisoria.....           | 120         |
| colisiones.....                              | 88          |
| <b>Collisions of objects</b> .....           | 91          |
| colocación de los silencios multicompás..... | 42          |
| coloreadas, notas.....                       | 111         |
| coloreados, objetos.....                     | 111         |
| colorear notas.....                          | 111         |
| colorear objects.....                        | 111         |
| colores.....                                 | 111         |
| colores, lista de.....                       | 242         |
| coloring text.....                           | 294         |
| coloring voices.....                         | 87          |
| <b>Combining notes into chords</b> .....     | 85          |
| comienzo del sistema.....                    | 98          |
| comillas, en la letra.....                   | 131         |
| compás completo, silencios de.....           | 40          |
| compás compuesto, indicaciones de.....       | 44          |
| compás parcial.....                          | 45          |
| compás, comprobación de.....                 | 56          |
| compás, indicación de.....                   | 185         |
| compás, número de.....                       | 63          |
| compás, números de.....                      | 55          |
| compás, repeticiones de.....                 | 82          |
| compases distintos al mismo tiempo.....      | 217         |
| <b>Completion_heads_engraver</b> .....       | 46, 47      |
| <b>composer</b> .....                        | 199         |
| compresión de música.....                    | 34          |
| compuesto, indicaciones de compás.....       | 44          |
| concatenating text.....                      | 268         |
| Contexto, creación de.....                   | 214         |
| contexto, definición de.....                 | 208         |
| contrabajo, clave de.....                    | 11          |
| controlling general text alignment.....      | 269         |
| <b>copyright</b> .....                       | 199         |
| copyright.....                               | 202         |
| coral, partitura.....                        | 137         |
| corchete vertical.....                       | 98          |
| corchetes.....                               | 117, 184    |
| crear contextos.....                         | 215         |
| creating empty text objects.....             | 292         |
| creating horizontal spaces in text.....      | 271         |
| creating text fractions.....                 | 291         |
| creating vertical spaces in text.....        | 293         |
| crescendo.....                               | 67          |
| <b>crescendo</b> .....                       | 69          |
| cruce de pentagramas.....                    | 147         |
| cuartos de tono.....                         | 5           |
| <b>cueDuring</b> .....                       | 236, 320    |
| cuerda, número de la.....                    | 150         |
| <b>currentBarNumber</b> .....                | 55          |
| <b>currentBarNumber</b> .....                | 63          |
| custodias.....                               | 187         |
| custos.....                                  | 187         |
| <b>Custos</b> .....                          | 188         |
| <b>Custos_engraver</b> .....                 | 187         |

## D

|                             |     |
|-----------------------------|-----|
| D.S al Fine.....            | 58  |
| decrescendo.....            | 67  |
| <b>decrescendo</b> .....    | 69  |
| dedicatoria.....            | 199 |
| <b>defaultBarType</b> ..... | 55  |

|                                                     |               |
|-----------------------------------------------------|---------------|
| Desplazamiento.....                                 | 39            |
| desplazamiento de silencios, automático.....        | 88            |
| desplazar nota.....                                 | 89            |
| digitación.....                                     | 109           |
| digitaciones, mano derecha, de guitarra.....        | 154           |
| <b>dim</b> .....                                    | 165           |
| diminuendo.....                                     | 67            |
| dinámica.....                                       | 66            |
| dinámicas, indicaciones, nuevas.....                | 70            |
| <b>dispatcher</b> .....                             | 322           |
| <b>displayLilyMusic</b> .....                       | 236, 320      |
| <b>displayMusic</b> .....                           | 236, 320      |
| disposición de la página.....                       | 203           |
| distancia entre pentagramas en música de piano..... | 148           |
| <b>divisible</b> .....                              | 48            |
| divisio.....                                        | 188           |
| divisiones.....                                     | 188           |
| Do, clave de.....                                   | 11            |
| dobladas, barras.....                               | 49            |
| doble bemol.....                                    | 4             |
| <b>doble bemol</b> .....                            | 6             |
| doble sostenido.....                                | 4             |
| <b>doble sostenido</b> .....                        | 6             |
| documentación interna.....                          | 227           |
| <b>DotColumn</b> .....                              | 31            |
| <b>Dots</b> .....                                   | 31            |
| <b>DoublePercentRepeat</b> .....                    | 83            |
| <b>DoublePercentRepeatCounter</b> .....             | 83            |
| downbow.....                                        | 295           |
| drawing beams within text.....                      | 280           |
| drawing boxes with rounded corners.....             | 282           |
| drawing boxes with rounded corners around text..... | 283           |
| drawing circles within text.....                    | 281           |
| drawing lines within text.....                      | 281           |
| drawing solid boxes within text.....                | 282           |
| drawing triangles within text.....                  | 284           |
| <b>DrumStaff</b> .....                              | 158, 160      |
| <b>DrumVoice</b> .....                              | 158, 159, 160 |
| duración de las notas.....                          | 30            |
| duraciones automáticas de las sílabas.....          | 134           |
| durations, scaling.....                             | 34            |
| <b>DynamicLineSpanner</b> .....                     | 67, 69        |
| <b>DynamicText</b> .....                            | 69            |

## E

|                                                   |          |
|---------------------------------------------------|----------|
| eclesiásticos, silencios.....                     | 41       |
| encabezamiento.....                               | 203      |
| enclosing text in a box with rounded corners..... | 283      |
| enclosing text within a box.....                  | 258      |
| <b>endSpanners</b> .....                          | 236, 320 |
| ensayo, letra de, estilo.....                     | 57       |
| ensayo, letra de, formato.....                    | 57       |
| ensayo, letras de.....                            | 57       |
| escribir música en paralelo.....                  | 95       |
| Espaciado de la letra.....                        | 139      |
| espacios, en la letra.....                        | 131      |
| especiales, cabezas de figura.....                | 26       |
| espressivo.....                                   | 64, 295  |
| espressivo, articulación.....                     | 67       |
| Establecer el comportamiento de las barras.....   | 297      |
| automáticas.....                                  | 297      |

|                                                    |     |
|----------------------------------------------------|-----|
| estándar, tamaño de la tipografía (de notación) .. | 109 |
| estilo de la letra de ensayo .....                 | 57  |
| evenFooterMarkup .....                             | 203 |
| evenHeaderMarkup .....                             | 202 |
| Event .....                                        | 228 |
| excepciones, nombres de acorde .....               | 172 |
| Explicitly instantiating voices .....              | 87  |
| extender .....                                     | 137 |

## F

|                                                             |                    |
|-------------------------------------------------------------|--------------------|
| Fa, clave de .....                                          | 11                 |
| fácil, notación, cabezas de nota de .....                   | 26                 |
| fantasma, notas .....                                       | 113                |
| FDL, GNU Free Documentation License .....                   | 344                |
| featherDurations .....                                      | 236, 320           |
| fermata .....                                               | 295                |
| Feta, tipografía .....                                      | 244                |
| figura .....                                                | 31                 |
| figura, cabeza de la .....                                  | 26                 |
| Figured bass .....                                          | 174                |
| FiguredBass .....                                           | 100, 177, 179      |
| finalis .....                                               | 188                |
| finger-interface .....                                      | 222                |
| Fingering .....                                             | 111, 220, 221, 222 |
| fingering chords .....                                      | 110                |
| fingering-event .....                                       | 111, 221           |
| Fingering_engraver .....                                    | 111, 221, 223      |
| FingeringEvent .....                                        | 111, 221           |
| flageolet .....                                             | 64, 295            |
| followVoice .....                                           | 147                |
| font-interface .....                                        | 109, 128, 222, 292 |
| foot marks .....                                            | 295                |
| Forbid_line_break_engraver .....                            | 46                 |
| forget accidental style .....                               | 23                 |
| formas, notas con .....                                     | 27                 |
| Formatear el texto .....                                    | 313, 318           |
| formato de la letra de ensayo .....                         | 57                 |
| francesa, clave .....                                       | 11                 |
| francesa, partituras a la .....                             | 100                |
| fraseo, corchetes de .....                                  | 117                |
| fraseo, en la letra .....                                   | 136                |
| fret-diagram-interface .....                                | 154                |
| fuerza tipográfica (de notación), escalado del tamaño ..... | 108                |
| fuerza tipográfica (de notación), tamaño .....              | 108                |
| fuerza tipográfica (de notación), tamaño estándar de .....  | 109                |
| fuerza, ampliación .....                                    | 128                |
| fuerza, ampliación de la .....                              | 129                |
| fuerza, establecer la familia de .....                      | 127                |
| fuerza, selección .....                                     | 128                |
| fuerza, tamaño de .....                                     | 129                |
| fundamental de un acorde .....                              | 165                |

## G

|                                            |          |
|--------------------------------------------|----------|
| glissando .....                            | 73       |
| glissando .....                            | 74       |
| Glissando .....                            | 74, 233  |
| globo .....                                | 114      |
| grace .....                                | 236, 320 |
| GraceMusic .....                           | 61       |
| gráficos, descripción de los objetos ..... | 227      |
| GrandStaff .....                           | 24, 98   |

|                                                     |          |
|-----------------------------------------------------|----------|
| grid-line-interface .....                           | 117      |
| grid-point-interface .....                          | 117      |
| Grid_line_span_engraver .....                       | 117      |
| Grid_point_engraver .....                           | 117      |
| GridLine .....                                      | 117      |
| GridPoint .....                                     | 117      |
| grob .....                                          | 221      |
| grob-interface .....                                | 221, 222 |
| grosor de las líneas del pentagrama, establecer ... | 100      |
| grupeto .....                                       | 64       |
| grupeto circular .....                              | 295      |
| grupeto circular invertido .....                    | 295      |
| grupo especial .....                                | 33       |
| grupos especiales .....                             | 31       |
| guía, formato de las notas .....                    | 106      |
| guías .....                                         | 105      |
| guiones .....                                       | 64, 137  |
| guitarra, cabezas de nota .....                     | 26       |
| guitarra, tablatura de .....                        | 150      |

## H

|                                    |     |
|------------------------------------|-----|
| Hairpin .....                      | 69  |
| Hal Leonard .....                  | 26  |
| header .....                       | 203 |
| horizontal-bracket-interface ..... | 117 |
| Horizontal_bracket_engraver .....  | 117 |
| HorizontalBracket .....            | 117 |
| horizontally centering text .....  | 267 |
| hufnagel .....                     | 180 |

## I

|                                                 |          |
|-------------------------------------------------|----------|
| I'm hearing Voices .....                        | 88, 158  |
| importing stencils into text .....              | 293      |
| impresión de caracteres especiales .....        | 125      |
| impresión de los nombres de acorde .....        | 168      |
| improvisación .....                             | 28       |
| includePageLayoutFile .....                     | 236, 320 |
| Indicación de compás .....                      | 43       |
| indicación de pulgar .....                      | 64       |
| indicaciones de pedal .....                     | 64       |
| indicaciones de pedalero de órgano .....        | 64       |
| indicaciones dinámicas nuevas .....             | 70       |
| inlining an Encapsulated PostScript image ..... | 281      |
| inserting music into text .....                 | 286      |
| inserting PostScript directly into text .....   | 282      |
| inserting URL links into text .....             | 284      |
| instrument .....                                | 199      |
| InstrumentName .....                            | 104      |
| instrumento transpositor .....                  | 18       |
| instrumento, nombres de .....                   | 208      |
| instrumentSwitch .....                          | 236, 320 |
| interfaz de la presentación .....               | 221      |
| interna, documentación .....                    | 227      |
| intervalo .....                                 | 4        |
| invisibles, notas .....                         | 111      |
| item-interface .....                            | 222      |

## J

|                                |          |
|--------------------------------|----------|
| jazz, acordes de .....         | 163, 171 |
| justifying lines of text ..... | 295      |
| justifying text .....          | 272      |

## K

|                                  |              |
|----------------------------------|--------------|
| keepWithTag .....                | 236, 320     |
| key signature .....              | 4            |
| key-cancellation-interface ..... | 16           |
| key-signature-interface .....    | 16           |
| Key_engraver .....               | 16           |
| Key_performer .....              | 16           |
| KeyCancellation .....            | 16           |
| KeyChangeEvent .....             | 16           |
| KeySignature .....               | 16, 181, 182 |
| killCues .....                   | 236, 320     |
| kirchenpausen .....              | 41           |

## L

|                                                               |          |
|---------------------------------------------------------------|----------|
| La tipografía Feta .....                                      | 285      |
| label .....                                                   | 236, 320 |
| laissez vibrer .....                                          | 36       |
| laissez vibrer .....                                          | 37       |
| LaissezVibrerTie .....                                        | 37       |
| LaissezVibrerTieColumn .....                                  | 37       |
| layout, bloque .....                                          | 207      |
| ledger-line-spanner-interface .....                           | 26       |
| Ledger_line_engraver .....                                    | 26       |
| LedgerLineSpanner .....                                       | 26       |
| left aligning text .....                                      | 273      |
| letra .....                                                   | 51, 131  |
| letra de ensayo, estilo .....                                 | 57       |
| letra de ensayo, formato .....                                | 57       |
| letra y melodías .....                                        | 134      |
| Letra, aumentar el espaciado .....                            | 139      |
| letra, identificadores de .....                               | 133      |
| letras de ensayo .....                                        | 57       |
| ligadura de expresión .....                                   | 71       |
| ligadura de unión .....                                       | 35       |
| ligadura de unión .....                                       | 37       |
| Ligaduras .....                                               | 188      |
| ligaduras de expresión .....                                  | 70       |
| ligaduras de fraseo .....                                     | 71       |
| ligaduras de neumas cuadrados .....                           | 190      |
| ligaduras de unión y corchetes de primera y segunda vez ..... | 36       |
| ligaduras de unión, repeticiones y .....                      | 36       |
| ligaduras gregorianas de neumas cuadrados .....               | 190      |
| ligaduras laissez vibrer .....                                | 36       |
| ligaduras mensurales .....                                    | 189      |
| ligaduras mensurales blancas .....                            | 189      |
| ligaduras, apariencia .....                                   | 36       |
| ligaduras, colocación .....                                   | 36       |
| ligaduras, en el texto .....                                  | 135      |
| ligaduras, en la letra .....                                  | 131      |
| Ligature_bracket_engraver .....                               | 189, 190 |
| LigatureBracket .....                                         | 188      |
| ligatures in text .....                                       | 268      |
| line-spanner-interface .....                                  | 233      |
| línea, saltos de .....                                        | 48       |
| líneas divisorias .....                                       | 54       |
| líneas divisorias, símbolos sobre las .....                   | 120      |
| lista de colores .....                                        | 242      |
| listener .....                                                | 322      |
| llave vertical .....                                          | 98       |
| longa .....                                                   | 31       |
| lowering text .....                                           | 274      |
| ly:add-file-name-alist .....                                  | 322      |

|                                         |     |
|-----------------------------------------|-----|
| ly:add-interface .....                  | 322 |
| ly:add-listener .....                   | 322 |
| ly:add-option .....                     | 322 |
| ly:all-grob-interfaces .....            | 322 |
| ly:all-options .....                    | 322 |
| ly:all-stencil-expressions .....        | 322 |
| ly:assoc-get .....                      | 322 |
| ly:book-add-score! .....                | 322 |
| ly:book-process .....                   | 322 |
| ly:book-process-to-systems .....        | 322 |
| ly:box? .....                           | 322 |
| ly:bp .....                             | 322 |
| ly:bracket .....                        | 322 |
| ly:broadcast .....                      | 322 |
| ly:camel-case->lisp-identifier .....    | 323 |
| ly:chain-assoc-get .....                | 323 |
| ly:clear-anonymous-modules .....        | 323 |
| ly:cm .....                             | 323 |
| ly:command-line-code .....              | 323 |
| ly:command-line-options .....           | 323 |
| ly:command-line-verbose? .....          | 323 |
| ly:connect-dispatchers .....            | 323 |
| ly:context-event-source .....           | 323 |
| ly:context-events-below .....           | 323 |
| ly:context-find .....                   | 323 |
| ly:context-grob-definition .....        | 323 |
| ly:context-id .....                     | 323 |
| ly:context-name .....                   | 323 |
| ly:context-now .....                    | 323 |
| ly:context-parent .....                 | 323 |
| ly:context-property .....               | 323 |
| ly:context-property-where-defined ..... | 323 |
| ly:context-pushpop-property .....       | 323 |
| ly:context-set-property! .....          | 324 |
| ly:context-unset-property .....         | 324 |
| ly:context? .....                       | 324 |
| ly:default-scale .....                  | 324 |
| ly:dimension? .....                     | 324 |
| ly:dir? .....                           | 324 |
| ly:duration->string .....               | 324 |
| ly:duration-dot-count .....             | 324 |
| ly:duration-factor .....                | 324 |
| ly:duration-length .....                | 324 |
| ly:duration-log .....                   | 324 |
| ly:duration<? .....                     | 324 |
| ly:duration? .....                      | 324 |
| ly:effective-prefix .....               | 324 |
| ly:error .....                          | 324 |
| ly:eval-simple-closure .....            | 324 |
| ly:event-deep-copy .....                | 324 |
| ly:event-property .....                 | 324 |
| ly:event-set-property! .....            | 325 |
| ly:expand-environment .....             | 325 |
| ly:export .....                         | 325 |
| ly:find-file .....                      | 325 |
| ly:font-config-display-fonts .....      | 325 |
| ly:font-config-get-font-file .....      | 325 |
| ly:font-design-size .....               | 325 |
| ly:font-file-name .....                 | 325 |
| ly:font-get-glyph .....                 | 325 |
| ly:font-glyph-name-to-charcode .....    | 325 |
| ly:font-glyph-name-to-index .....       | 325 |
| ly:font-index-to-charcode .....         | 325 |
| ly:font-load .....                      | 325 |
| ly:font-magnification .....             | 325 |



|                                   |     |                                  |     |
|-----------------------------------|-----|----------------------------------|-----|
| ly:font-metric?                   | 325 | ly:make-output-def               | 329 |
| ly:font-name                      | 325 | ly:make-page-label-marker        | 329 |
| ly:font-sub-fonts                 | 325 | ly:make-page-permission-marker   | 329 |
| ly:format                         | 325 | ly:make-pango-description-string | 329 |
| ly:format-output                  | 325 | ly:make-paper-outputter          | 329 |
| ly:get-all-function-documentation | 326 | ly:make-pitch                    | 329 |
| ly:get-all-translators            | 326 | ly:make-prob                     | 329 |
| ly:get-glyph                      | 326 | ly:make-scale                    | 329 |
| ly:get-listened-event-classes     | 326 | ly:make-score                    | 329 |
| ly:get-option                     | 326 | ly:make-simple-closure           | 330 |
| ly:gettext                        | 326 | ly:make-stencil                  | 330 |
| ly:grob-alist-chain               | 326 | ly:make-stream-event             | 330 |
| ly:grob-array-length              | 326 | ly:message                       | 330 |
| ly:grob-array-ref                 | 326 | ly:minimal-breaking              | 330 |
| ly:grob-array?                    | 326 | ly:mm                            | 330 |
| ly:grob-basic-properties          | 326 | ly:module->alist                 | 330 |
| ly:grob-common-repoint            | 326 | ly:module-copy                   | 330 |
| ly:grob-common-repoint-of-array   | 326 | ly:modules-lookup                | 330 |
| ly:grob-default-font              | 326 | ly:moment-add                    | 330 |
| ly:grob-extent                    | 326 | ly:moment-div                    | 330 |
| ly:grob-interfaces                | 326 | ly:moment-grace-denominator      | 330 |
| ly:grob-layout                    | 326 | ly:moment-grace-numerator        | 330 |
| ly:grob-object                    | 326 | ly:moment-main-denominator       | 330 |
| ly:grob-original                  | 326 | ly:moment-main-numerator         | 330 |
| ly:grob-parent                    | 327 | ly:moment-mod                    | 330 |
| ly:grob-pq<?                      | 327 | ly:moment-mul                    | 331 |
| ly:grob-properties                | 327 | ly:moment-sub                    | 331 |
| ly:grob-property                  | 327 | ly:moment<?                      | 331 |
| ly:grob-property-data             | 327 | ly:moment?                       | 331 |
| ly:grob-relative-coordinate       | 327 | ly:music-compress                | 331 |
| ly:grob-robust-relative-extent    | 327 | ly:music-deep-copy               | 331 |
| ly:grob-script-priority-less      | 327 | ly:music-duration-compress       | 331 |
| ly:grob-set-property!             | 327 | ly:music-duration-length         | 331 |
| ly:grob-staff-position            | 327 | ly:music-function-extract        | 331 |
| ly:grob-suicide!                  | 327 | ly:music-function?               | 331 |
| ly:grob-system                    | 327 | ly:music-length                  | 331 |
| ly:grob-translate-axis!           | 327 | ly:music-list?                   | 331 |
| ly:grob?                          | 327 | ly:music-mutable-properties      | 331 |
| ly:gulp-file                      | 327 | ly:music-output?                 | 331 |
| ly:hash-table-keys                | 327 | ly:music-property                | 331 |
| ly:inch                           | 327 | ly:music-set-property!           | 331 |
| ly:input-both-locations           | 327 | ly:music-transpose               | 331 |
| ly:input-file-line-char-column    | 328 | ly:music?                        | 331 |
| ly:input-location?                | 328 | ly:note-head::stem-attachment    | 331 |
| ly:input-message                  | 328 | ly:number->string                | 332 |
| ly:interpret-music-expression     | 328 | ly:optimal-breaking              | 332 |
| ly:interpret-stencil-expression   | 328 | ly:option-usage                  | 332 |
| ly:intlog2                        | 328 | ly:otf->cff                      | 332 |
| ly:is-listened-event-class        | 328 | ly:otf-font-glyph-info           | 332 |
| ly:item-break-dir                 | 328 | ly:otf-font-table-data           | 332 |
| ly:item?                          | 328 | ly:otf-font?                     | 332 |
| ly:iterator?                      | 328 | ly:otf-glyph-list                | 332 |
| ly:lexer-keywords                 | 328 | ly:output-def-clone              | 332 |
| ly:lily-lexer?                    | 328 | ly:output-def-lookup             | 332 |
| ly:lily-parser?                   | 328 | ly:output-def-parent             | 332 |
| ly:load-text-dimensions           | 328 | ly:output-def-scope              | 332 |
| ly:make-book                      | 328 | ly:output-def-set-variable!      | 332 |
| ly:make-dispatcher                | 328 | ly:output-def?                   | 332 |
| ly:make-duration                  | 328 | ly:output-description            | 332 |
| ly:make-global-context            | 329 | ly:output-formats                | 332 |
| ly:make-global-translator         | 329 | ly:outputter-close               | 332 |
| ly:make-listener                  | 329 | ly:outputter-dump-stencil        | 332 |
| ly:make-moment                    | 329 | ly:outputter-dump-string         | 333 |
| ly:make-music                     | 329 | ly:outputter-output-scheme       | 333 |
| ly:make-music-function            | 329 | ly:outputter-port                | 333 |

|                                        |     |
|----------------------------------------|-----|
| ly:page-marker? .....                  | 333 |
| ly:page-turn-breaking .....            | 333 |
| ly:pango-font-physical-fonts .....     | 333 |
| ly:pango-font? .....                   | 333 |
| ly:paper-book-pages .....              | 333 |
| ly:paper-book-paper .....              | 333 |
| ly:paper-book-performances .....       | 333 |
| ly:paper-book-scopes .....             | 333 |
| ly:paper-book-systems .....            | 333 |
| ly:paper-book? .....                   | 333 |
| ly:paper-fonts .....                   | 333 |
| ly:paper-get-font .....                | 333 |
| ly:paper-get-number .....              | 333 |
| ly:paper-outputsacle .....             | 333 |
| ly:paper-score-paper-systems .....     | 333 |
| ly:paper-system-minimum-distance ..... | 333 |
| ly:paper-system? .....                 | 334 |
| ly:parse-file .....                    | 334 |
| ly:parser-clear-error .....            | 334 |
| ly:parser-clone .....                  | 334 |
| ly:parser-define! .....                | 334 |
| ly:parser-error .....                  | 334 |
| ly:parser-has-error? .....             | 334 |
| ly:parser-lexer .....                  | 334 |
| ly:parser-lookup .....                 | 334 |
| ly:parser-output-name .....            | 334 |
| ly:parser-parse-string .....           | 334 |
| ly:parser-set-note-names .....         | 334 |
| ly:performance-write .....             | 334 |
| ly:pfb->pfa .....                      | 334 |
| ly:pitch-alteration .....              | 334 |
| ly:pitch-diff .....                    | 334 |
| ly:pitch-negate .....                  | 334 |
| ly:pitch-notename .....                | 334 |
| ly:pitch-octave .....                  | 334 |
| ly:pitch-quartertines .....            | 335 |
| ly:pitch-semitones .....               | 335 |
| ly:pitch-steps .....                   | 335 |
| ly:pitch-transpose .....               | 335 |
| ly:pitch<? .....                       | 335 |
| ly:pitch? .....                        | 335 |
| ly:prob-immutable-properties .....     | 335 |
| ly:prob-mutable-properties .....       | 335 |
| ly:prob-property .....                 | 335 |
| ly:prob-property? .....                | 335 |
| ly:prob-set-property! .....            | 335 |
| ly:prob-type? .....                    | 335 |
| ly:prob? .....                         | 335 |
| ly:programming-error .....             | 335 |
| ly:progress .....                      | 335 |
| ly:property-lookup-stats .....         | 335 |
| ly:protects .....                      | 335 |
| ly:pt .....                            | 335 |
| ly:register-stencil-expression .....   | 336 |
| ly:relative-group-extent .....         | 336 |
| ly:reset-all-fonts .....               | 336 |
| ly:round-filled-box .....              | 336 |
| ly:round-filled-polygon .....          | 336 |
| ly:run-translator .....                | 336 |
| ly:score-add-output-def! .....         | 336 |
| ly:score-embedded-format .....         | 336 |
| ly:score-error? .....                  | 336 |
| ly:score-header .....                  | 336 |
| ly:score-music .....                   | 336 |
| ly:score-output-defs .....             | 336 |

|                                           |                         |
|-------------------------------------------|-------------------------|
| ly:score-set-header! .....                | 336                     |
| ly:score? .....                           | 336                     |
| ly:set-default-scale .....                | 336                     |
| ly:set-grob-modification-callback .....   | 336                     |
| ly:set-middle-C! .....                    | 337                     |
| ly:set-option .....                       | 337                     |
| ly:set-point-and-click .....              | 337                     |
| ly:set-property-cache-callback .....      | 337                     |
| ly:simple-closure? .....                  | 337                     |
| ly:skyline-pair? .....                    | 337                     |
| ly:skyline? .....                         | 337                     |
| ly:smob-protects .....                    | 337                     |
| ly:solve-spring-rod-problem .....         | 337                     |
| ly:source-file? .....                     | 337                     |
| ly:spanner-bound .....                    | 337                     |
| ly:spanner-broken-into .....              | 337                     |
| ly:spanner? .....                         | 337                     |
| ly:start-environment .....                | 337                     |
| ly:stderr-redirect .....                  | 337                     |
| ly:stencil-add .....                      | 338                     |
| ly:stencil-aligned-to .....               | 338                     |
| ly:stencil-combine-at-edge .....          | 338                     |
| ly:stencil-empty? .....                   | 338                     |
| ly:stencil-expr .....                     | 338                     |
| ly:stencil-extent .....                   | 338                     |
| ly:stencil-fonts .....                    | 338                     |
| ly:stencil-in-color .....                 | 338                     |
| ly:stencil-rotate .....                   | 338                     |
| ly:stencil-translate .....                | 338                     |
| ly:stencil-translate-axis .....           | 338                     |
| ly:stencil? .....                         | 338                     |
| ly:stream-event? .....                    | 338                     |
| ly:string-substitute .....                | 338                     |
| ly:system-print .....                     | 338                     |
| ly:system-stretch .....                   | 338                     |
| ly:text-dimension .....                   | 338                     |
| ly:text-interface::interpret-markup ..... | 339                     |
| ly:translator-description .....           | 339                     |
| ly:translator-group? .....                | 339                     |
| ly:translator-name .....                  | 339                     |
| ly:translator? .....                      | 339                     |
| ly:transpose-key-alist .....              | 339                     |
| ly:truncate-list! .....                   | 339                     |
| ly:ttf->pfa .....                         | 339                     |
| ly:ttf-ps-name .....                      | 339                     |
| ly:unit .....                             | 339                     |
| ly:usage .....                            | 339                     |
| ly:version .....                          | 339                     |
| ly:warning .....                          | 339                     |
| ly:wide-char->utf-8 .....                 | 339                     |
| LyricCombineMusic .....                   | 133, 136                |
| LyricExtender .....                       | 137                     |
| LyricHyphen .....                         | 137                     |
| Lyrics .....                              | 100, 133, 134, 135, 300 |
| LyricSpace .....                          | 133                     |
| LyricText .....                           | 133, 145                |

## M

|                           |     |
|---------------------------|-----|
| m .....                   | 165 |
| magnifying text .....     | 261 |
| maj .....                 | 165 |
| major seven symbols ..... | 172 |
| majorSevenSymbol .....    | 171 |



|                                                     |               |
|-----------------------------------------------------|---------------|
| make-dynamic-script .....                           | 70            |
| makeClusters .....                                  | 236, 320      |
| mano derecha, digitaciones para guitarra .....      | 154           |
| manual, cambio de pentagrama .....                  | 146           |
| manuales, barras .....                              | 47            |
| marca de ensayo, estilo .....                       | 57            |
| marca de ensayo, formato .....                      | 57            |
| marcado .....                                       | 125           |
| marcado de texto .....                              | 125           |
| marcado, caracteres especiales en modo de .....     | 125           |
| marcado, expresiones de .....                       | 125           |
| marcado, sintaxis .....                             | 125           |
| marcas de fraseo .....                              | 71            |
| marcato .....                                       | 64, 295       |
| matices .....                                       | 66            |
| matices absolutos .....                             | 66            |
| matices editoriales .....                           | 70            |
| matices entre paréntesis .....                      | 70            |
| Measure_grouping_engraver .....                     | 44            |
| MeasureGrouping .....                               | 44            |
| measureLength .....                                 | 63            |
| measurePosition .....                               | 63            |
| Medicaea, Editio .....                              | 180           |
| medida, música sin .....                            | 63            |
| melisma .....                                       | 136, 137      |
| melismata .....                                     | 136           |
| mensural .....                                      | 180           |
| Mensural_ligature_engraver .....                    | 181, 189, 190 |
| MensuralStaffContext .....                          | 197           |
| MensuralVoiceContext .....                          | 197           |
| merging text .....                                  | 268           |
| meter .....                                         | 199           |
| metro, medida .....                                 | 43            |
| MetronomeMark .....                                 | 102           |
| metrónomo, indicación de .....                      | 101           |
| mezclar notas .....                                 | 88            |
| mezzosoprano, clave de .....                        | 11            |
| MIDI .....                                          | 17, 206       |
| minimumFret .....                                   | 151           |
| modern .....                                        | 21            |
| modern-cautionary .....                             | 21            |
| modern-voice .....                                  | 21            |
| modern-voice-cautionary .....                       | 22            |
| moderno de precaución, alteraciones de estilo ..... | 21            |
| moderno, alteraciones de estilo .....               | 21            |
| modificadores de acorde .....                       | 165           |
| modo de acordes .....                               | 163           |
| modo eclesiástico .....                             | 16            |
| modos .....                                         | 14            |
| modos eclesiásticos .....                           | 14            |
| mordent .....                                       | 295           |
| mordente .....                                      | 64            |
| mordente circular .....                             | 64            |
| mordente circular, inverso .....                    | 64            |
| mordentes .....                                     | 59            |
| multicompás, posicionamiento de los silencios ..... | 42            |
| multicompás, silencios .....                        | 40            |
| MultiMeasureRest .....                              | 42            |
| MultiMeasureRestNumber .....                        | 42            |
| MultiMeasureRestText .....                          | 42            |
| Multiple notes at once .....                        | 91            |
| multiple voices .....                               | 90            |
| Musica ficta .....                                  | 197           |
| música intercalada .....                            | 95            |
| música para principiantes .....                     | 26            |

|                              |          |
|------------------------------|----------|
| música sin medida .....      | 63       |
| musicMap .....               | 236, 320 |
| musicológico, análisis ..... | 117      |

## N

|                                                     |                     |
|-----------------------------------------------------|---------------------|
| New_fingering_engraver .....                        | 111, 221            |
| niente, al .....                                    | 68                  |
| no restablecer, estilo de alteraciones .....        | 22                  |
| no vacíos, textos .....                             | 118                 |
| no-reset accidental style .....                     | 22                  |
| Nombre de las duraciones de notas y silencios ..... | 31                  |
| nombre del cantante .....                           | 141                 |
| nombres de las notas .....                          | 1                   |
| Nombres de las notas .....                          | 2, 4, 6, 7          |
| nombres de las notas en otros idiomas .....         | 6                   |
| noPageBreak .....                                   | 236, 320            |
| noPageTurn .....                                    | 236, 320            |
| nota abierta .....                                  | 64                  |
| Nota de espaciado .....                             | 39                  |
| nota tapada .....                                   | 64                  |
| nota, cabeza de, armónico .....                     | 26                  |
| nota, cabeza de, en aspás .....                     | 26                  |
| nota, cabeza de, en parlato .....                   | 26                  |
| nota, cabeza de, en rombo .....                     | 26                  |
| nota, cabeza de, guitarra .....                     | 26                  |
| nota, cabeza especial de .....                      | 26                  |
| nota, estilos de cabeza .....                       | 257                 |
| notación, explicar la .....                         | 114                 |
| notas en otros idiomas, nombres de .....            | 6                   |
| notas entre paréntesis .....                        | 113                 |
| notas fantasma .....                                | 113                 |
| notas, cabezas de, Aiken .....                      | 27                  |
| notas, cabezas de, arpa sacra .....                 | 27                  |
| notas, cabezas de, barradas .....                   | 28                  |
| notas, cabezas de, con formas .....                 | 27                  |
| notas, cabezas de, estudio .....                    | 26                  |
| notas, cabezas de, improvisación .....              | 28                  |
| notas, cabezas de, notación simplificada .....      | 26                  |
| notas, colisiones de .....                          | 88                  |
| notas, coloreadas .....                             | 111                 |
| notas, con puntillo .....                           | 30                  |
| notas, corchetes de agrupación de .....             | 117                 |
| notas, duración de .....                            | 30                  |
| notas, nombres holandeses de .....                  | 4                   |
| notas, nombres predeterminados .....                | 4                   |
| note-collision-interface .....                      | 310, 312, 314       |
| note-event .....                                    | 26, 27, 28, 158     |
| note-head-interface .....                           | 26, 27, 28          |
| Note_heads_engraver .....                           | 26, 27, 28, 46, 219 |
| Note_spacing_engraver .....                         | 111                 |
| NoteCollision .....                                 | 91                  |
| NoteColumn .....                                    | 91                  |
| NoteHead .....                                      | 26, 27, 28, 181     |
| notes within text by log and dot-count .....        | 285                 |
| notes within text by string .....                   | 285                 |
| NoteSpacing .....                                   | 111                 |
| nots, cabezas de, para practicar .....              | 26                  |
| nuevos, contextos .....                             | 214                 |
| número de líneas del pentagrama, establecer .....   | 100                 |
| números de compás .....                             | 55                  |

## O

|                                                |          |
|------------------------------------------------|----------|
| octava, comprobación de .....                  | 7        |
| octava, corrección de .....                    | 7        |
| octava, transposición de .....                 | 12       |
| octavación .....                               | 16       |
| octavación .....                               | 17       |
| OctavateEight .....                            | 14       |
| octaveCheck .....                              | 236, 320 |
| ocultación de pentagramas .....                | 100      |
| ocultas, notas .....                           | 111      |
| oddFooterMarkup .....                          | 203      |
| oddHeaderMarkup .....                          | 202      |
| oldaddyrics .....                              | 237, 320 |
| olvidar, estilo de alteraciones .....          | 23       |
| open .....                                     | 295      |
| opus .....                                     | 199      |
| organ pedal marks .....                        | 295      |
| Organizing pieces with variables .....         | 97       |
| órgano, indicaciones de pedal de .....         | 295      |
| ornamentos .....                               | 59, 64   |
| ossia .....                                    | 100, 220 |
| Other sources of information .....             | 88       |
| ottava .....                                   | 16       |
| ottava .....                                   | 237, 320 |
| ottava-bracket-interface .....                 | 17       |
| Ottava_spanner_engraver .....                  | 17       |
| OttavaBracket .....                            | 17       |
| overrideProperty .....                         | 237, 320 |
| OverrideProperty .....                         | 225      |
| overriding properties within text markup ..... | 292      |

## P

|                                                          |          |
|----------------------------------------------------------|----------|
| padding text .....                                       | 275      |
| padding text horizontally .....                          | 275      |
| pageBreak .....                                          | 237, 320 |
| pageTurn .....                                           | 237, 321 |
| página, saltos, forzar .....                             | 199      |
| Pango .....                                              | 127      |
| parallelMusic .....                                      | 95       |
| parallelMusic .....                                      | 237, 321 |
| parcial, compás .....                                    | 45       |
| paréntesis .....                                         | 113      |
| paréntesis en ángulo .....                               | 85       |
| paréntesis, alteración con .....                         | 5        |
| parentheses-interface .....                              | 113      |
| ParenthesesItem .....                                    | 113      |
| Parenthesis_engraver .....                               | 113      |
| parenthesize .....                                       | 237, 321 |
| parlato, cabezas de nota .....                           | 26       |
| partcombine .....                                        | 237, 321 |
| PartCombineMusic .....                                   | 94       |
| parte .....                                              | 94       |
| parte al alzar .....                                     | 45       |
| partes, combinación automática .....                     | 91       |
| partes, combinador de .....                              | 91       |
| pauta de coro .....                                      | 98       |
| pedal, indicaciones de .....                             | 295      |
| Pedales .....                                            | 148      |
| pentagrama cruzado, plica de .....                       | 148      |
| pentagrama, cambio .....                                 | 147      |
| pentagrama, cambios manuales de .....                    | 146      |
| pentagrama, establecer el grosor de las líneas del ..... | 100      |

|                                                     |              |
|-----------------------------------------------------|--------------|
| pentagrama, establecer el número de líneas del ..   | 100          |
| pentagramas de piano .....                          | 98           |
| pentagramas, grupo de .....                         | 98           |
| Pentagramas, varios .....                           | 98           |
| PercentRepeat .....                                 | 83           |
| PercentRepeatCounter .....                          | 83           |
| PercentRepeatedMusic .....                          | 83           |
| percusión .....                                     | 157, 158     |
| personalización de los nombres de acorde .....      | 171          |
| Petrucci .....                                      | 180          |
| PhrasingSlur .....                                  | 72           |
| piano de precaución, alteraciones .....             | 22           |
| piano de precaución, estilo de alteraciones .....   | 22           |
| Piano templates .....                               | 146          |
| piano, alteraciones de .....                        | 22           |
| piano, estilo de alteraciones de .....              | 22           |
| PianoPedalBracket .....                             | 149          |
| PianoStaff .....                                    | 24, 77, 102  |
| picado .....                                        | 64           |
| pie .....                                           | 203          |
| piece .....                                         | 199          |
| pipeSymbol .....                                    | 57           |
| Pitch_squash_engraver .....                         | 29, 219, 303 |
| pitchedTrill .....                                  | 237, 321     |
| placing horizontal brackets around text .....       | 282          |
| placing vertical brackets around text .....         | 280          |
| plica .....                                         | 113          |
| plica de pentagrama cruzado .....                   | 148          |
| poet .....                                          | 199          |
| pointAndClickOff .....                              | 237, 321     |
| pointAndClickOn .....                               | 237, 321     |
| polifonía .....                                     | 91           |
| polimétrica .....                                   | 33           |
| polimétricas, partituras .....                      | 217          |
| polyphonic music .....                              | 90           |
| porcentaje, repeticiones de .....                   | 82           |
| portato .....                                       | 64           |
| portato .....                                       | 66           |
| portato .....                                       | 295          |
| prall .....                                         | 295          |
| prallmordent .....                                  | 64, 295      |
| prallprall .....                                    | 64, 295      |
| precaución, alteración de .....                     | 5            |
| precaución, alteraciones de piano de .....          | 22           |
| precaución, estilo de alteraciones, piano .....     | 22           |
| predeterminado, estilo de las alteraciones .....    | 18           |
| presentación, interfaz de la .....                  | 221          |
| prima volta .....                                   | 78           |
| primera y segunda vez .....                         | 78           |
| primera y segunda vez, corchetes y ligaduras en ... | 36           |
| principiantes, música para .....                    | 26           |
| printallheaders .....                               | 202          |
| PropertySet .....                                   | 225          |
| propiedades .....                                   | 225          |
| pulsos por minuto .....                             | 101          |
| puntillo, notas con .....                           | 30           |
| puntuación .....                                    | 131          |
| putting space around text .....                     | 275          |

## Q

|                        |          |
|------------------------|----------|
| quebrado, acorde ..... | 74       |
| quinta .....           | 4        |
| quoteDuring .....      | 237, 321 |

QuoteMusic ..... 106

## R

**r** ..... 37  
**R** ..... 40  
 racimo (cluster) ..... 85  
 raising text ..... 276  
 rango de alturas ..... 24  
 recordatoria, alteración ..... 5  
 redonda, silencios de, para un compás completo... 40  
 Referencia de funcionamiento interno ..... 213  
 referencing page numbers in text ..... 293  
 regulador ..... 66  
 RehearsalMark ..... 59, 123  
 relativas, especificación de octavas ..... 2  
 RelativeOctaveCheck ..... 8  
 RelativeOctaveMusic ..... 4  
 relativo ..... 2  
 relleno ..... 223  
 removeWithTag ..... 237, 321  
 repeatCommands ..... 55, 81  
 RepeatedMusic ..... 81, 82  
 RepeatSlash ..... 83  
 repetición ..... 81  
 repetición ambigua ..... 81  
 repetición, barras de ..... 54  
 repetición, ligaduras de unión en ..... 36  
 repeticiones, expandir ..... 209  
 reservados, impresión de caracteres ..... 125  
 resetRelativeOctave ..... 237, 321  
 respiraciones ..... 72  
 Rest ..... 39  
 Rest ..... 182  
 RestCollision ..... 91  
 reverseturn ..... 295  
 RevertProperty ..... 225  
 RhythmicStaff ..... 29  
 right aligning text ..... 276  
 rightHandFinger ..... 237, 321  
 rombo, cabezas de nota ..... 26  
 rotating text ..... 277

## S

**s** ..... 39  
 salto de línea ..... 48  
 SATB ..... 137  
 scaleDurations ..... 237, 321  
 scaling durations ..... 34  
 scaling text ..... 277  
 Scheme tutorial ..... 213  
 scordatura ..... 16  
 Score ..... 43, 64, 297, 299  
 scoreTitleMarkup ..... 202  
 scoreTweak ..... 237, 321  
 Script ..... 66  
 seconda volta ..... 78  
 segno ..... 58, 64, 295  
 segno sobre la línea divisoria ..... 120  
 seguimiento de voz ..... 147  
 selección del tamaño de la fuente tipográfica (de notación) ..... 108  
 self-alignment-interface ..... 222

semi-bemoles, semi-sostenidos ..... 5  
 semitono ..... 6  
 semitrino ..... 64, 295  
 semitrino ascendente ..... 64  
 semitrino descendente ..... 64, 295  
 separate text ..... 124  
 séptima, acordes de ..... 165  
 set-accidental-style ..... 18  
 setting extent of text objects ..... 294  
 setting horizontal text alignment ..... 270  
 setting subscript in standard font size ..... 261  
 setting superscript in standard font size ..... 262  
 shapeNoteStyles ..... 27  
 shiftDurations ..... 237, 321  
 shifting voices ..... 90  
 side-position-interface ..... 222  
 Silencio invisible ..... 39  
 Silencios ..... 37  
 silencios antiguos ..... 182  
 silencios condensados ..... 43  
 Silencios de compás completo ..... 40  
 silencios eclesiásticos ..... 41  
 Silencios multicomás ..... 40  
 simple text strings ..... 263  
 simple text strings with tie characters ..... 288  
 simplificada, notación ..... 26  
 simultáneas, notas y alteraciones ..... 24  
 SkipMusic ..... 40  
 slashed digits ..... 293  
 Slur ..... 71  
 Sol, clave de ..... 11  
 sonido ..... 206  
 soprano, clave de ..... 11  
 sostenido ..... 4  
 sostenido ..... 6  
 sostenido, doble ..... 4  
 spacing-spanner-interface ..... 316, 317  
 spacingTweaks ..... 237, 321  
 SpanBar ..... 55  
 staccatissimo ..... 64, 295  
 staccato ..... 64  
 staccato ..... 66  
 staccato ..... 295  
 stacking text in a column ..... 267  
 Staff ..... 24, 25  
 Staff ..... 39  
 Staff ..... 55, 101, 102, 117, 187, 297  
 Staff.midiInstrument ..... 208  
 StaffGroup ..... 56, 98  
 StaffSymbol ..... 100  
 standalone text ..... 124  
 StanzaNumber ..... 145  
 Stem ..... 114, 184  
 stem, direction ..... 114  
 stem, down ..... 114  
 stem, neutral ..... 114  
 stem, up ..... 114  
 stem-interface ..... 114  
 Stem\_engraver ..... 114  
 stemLeftBeamCount ..... 52  
 stemRightBeamCount ..... 52  
 StemTremolo ..... 84  
 stopped ..... 295  
 storePredefinedDiagram ..... 237, 321  
 StringNumber ..... 151

|                           |     |
|---------------------------|-----|
| <b>StrokeFinger</b> ..... | 155 |
| styles, voice.....        | 87  |
| subdivideBeams.....       | 48  |
| subscript text.....       | 264 |
| subsubtitle.....          | 199 |
| subtitle.....             | 199 |
| suggestAccidentals.....   | 197 |
| superscript text.....     | 265 |
| sus.....                  | 167 |
| SustainPedal.....         | 149 |
| SystemStartBar.....       | 99  |
| SystemStartBrace.....     | 99  |
| SystemStartBracket.....   | 99  |

## T

|                                                     |                   |
|-----------------------------------------------------|-------------------|
| <b>Tab_note_heads_engraver</b> .....                | 153               |
| tablatura.....                                      | 150               |
| tablatura predeterminada.....                       | 151               |
| tablatura, principios básicos.....                  | 151               |
| tablaturas personalizadas.....                      | 152               |
| <b>TabNoteHead</b> .....                            | 152               |
| <b>TabStaff</b> .....                               | 151, 152          |
| <b>TabVoice</b> .....                               | 151, 152          |
| <b>tag</b> .....                                    | 237, 321          |
| <b>tagline</b> .....                                | 199               |
| tagline.....                                        | 202               |
| <b>Tempo</b> .....                                  | 101               |
| tenor, clave de.....                                | 11                |
| tenuto.....                                         | 64                |
| tenuto.....                                         | 66                |
| tenuto.....                                         | 295               |
| tesitura.....                                       | 24                |
| tesitura.....                                       | 25                |
| text columns, left-aligned.....                     | 274               |
| text columns, right-aligned.....                    | 276               |
| text spanners.....                                  | 119               |
| text, standalone.....                               | 124               |
| <b>text-interface</b> .....                         | 222, 292          |
| <b>text-script-interface</b> .....                  | 222               |
| texto sobre silencios multicompás.....              | 41                |
| texto, elementos no vacíos de.....                  | 118               |
| texto, en otros idiomas.....                        | 118               |
| Texto, inscripciones de.....                        | 118               |
| texto, marcado de.....                              | 125               |
| <b>TextScript</b> .....                             | 66, 119, 124, 126 |
| <b>TextSpanner</b> .....                            | 120, 233          |
| <b>textSpannerDown</b> .....                        | 120               |
| <b>textSpannerNeutral</b> .....                     | 120               |
| <b>textSpannerUp</b> .....                          | 120               |
| textual, inscripción, sobre silencios multicompás.. | 41                |
| Thorough bass.....                                  | 174               |
| thumb marking.....                                  | 295               |
| thumb-script.....                                   | 109               |
| <b>Tie</b> .....                                    | 37                |
| <b>TieColumn</b> .....                              | 37                |
| tiempo, administración.....                         | 63                |
| tiempo, control del (dentro de la partitura).....   | 63                |
| ties and chords.....                                | 35                |
| <b>TimeScaledMusic</b> .....                        | 33                |
| <b>TimeSignature</b> .....                          | 45, 185           |
| timing, internal.....                               | 59                |
| <b>Timing_translator</b> .....                      | 45, 64, 299       |
| tipografía Feta.....                                | 244               |

|                                         |          |
|-----------------------------------------|----------|
| tipografiar texto.....                  | 125      |
| <b>title</b> .....                      | 199      |
| títulos.....                            | 203      |
| <b>tocItem</b> .....                    | 237, 321 |
| top-level text.....                     | 124      |
| translating text.....                   | 277      |
| <b>Translation</b> .....                | 221      |
| transparentes, notas.....               | 111      |
| transporte.....                         | 8        |
| <b>transposedCueDuring</b> .....        | 238, 321 |
| <b>TransposedMusic</b> .....            | 11       |
| transposición.....                      | 8        |
| transposición de alturas.....           | 8        |
| transposición de MIDI.....              | 17       |
| transposición instrumental.....         | 17       |
| <b>transposition</b> .....              | 238, 321 |
| transpositor, instrumento.....          | 17       |
| transpositoras, claves.....             | 12       |
| transpositores, instrumentos.....       | 9        |
| traste.....                             | 151      |
| trastes, diagramas de.....              | 153      |
| tremolo, indicaciones de.....           | 84       |
| <b>tremoloFlags</b> .....               | 84       |
| <b>tresillo</b> .....                   | 33       |
| tresillos.....                          | 31       |
| triadas.....                            | 165      |
| trill.....                              | 295      |
| <b>TrillSpanner</b> .....               | 78, 233  |
| trino.....                              | 64       |
| <b>trino</b> .....                      | 78       |
| trino.....                              | 295      |
| trinos.....                             | 77       |
| trinos con altura.....                  | 77       |
| triplet formatting.....                 | 32       |
| trucos.....                             | 227      |
| tuplet formatting.....                  | 32       |
| <b>TupletBracket</b> .....              | 33       |
| <b>TupletNumber</b> .....               | 32, 33   |
| <b>tupletNumberFormatFunction</b> ..... | 32       |
| <b>tupletSpannerDuration</b> .....      | 32       |
| turn.....                               | 295      |
| <b>tweak</b> .....                      | 238, 321 |
| <b>Tweaking methods</b> .....           | 33       |

## U

|                                    |          |
|------------------------------------|----------|
| underlining text.....              | 266      |
| <b>UnfoldedRepeatedMusic</b> ..... | 81, 82   |
| <b>unfoldRepeats</b> .....         | 238, 321 |
| upbow.....                         | 295      |

## V

|                                              |          |
|----------------------------------------------|----------|
| varcoda.....                                 | 64, 295  |
| Vaticana, Editio.....                        | 180      |
| <b>Vaticana_ligature_engraver</b> .....      | 181      |
| <b>VaticanaStaffContext</b> .....            | 196      |
| <b>VaticanaVoiceContext</b> .....            | 196      |
| verso, número de.....                        | 140      |
| <b>VerticalAxisGroup</b> .....               | 101      |
| vertically centering text.....               | 278      |
| violín, clave de.....                        | 11       |
| <b>Visibility and color of objects</b> ..... | 216      |
| <b>Vocal ensembles</b> .....                 | 137, 220 |

voces, alteraciones en varias ..... 21  
**Voice** ..... 25, 29  
**Voice** ..... 39, 92  
**Voice** ..... 94, 95, 106, 134, 135, 189, 207, 223  
voice styles ..... 87  
**VoiceFollower** ..... 148, 233  
**Voices contain music** ..... 87, 91  
volta ..... 78  
volta, prima ..... 78  
volta, seconda ..... 78

**Volta\_engraver** ..... 170  
**VoltaBracket** ..... 81, 82  
**VoltaRepeatedMusic** ..... 81, 82

## W

**whichBar** ..... 55  
**Within-staff objects** ..... 230  
**withMusicProperty** ..... 238, 321