mehackit

Tecnología creativa para jóvenes

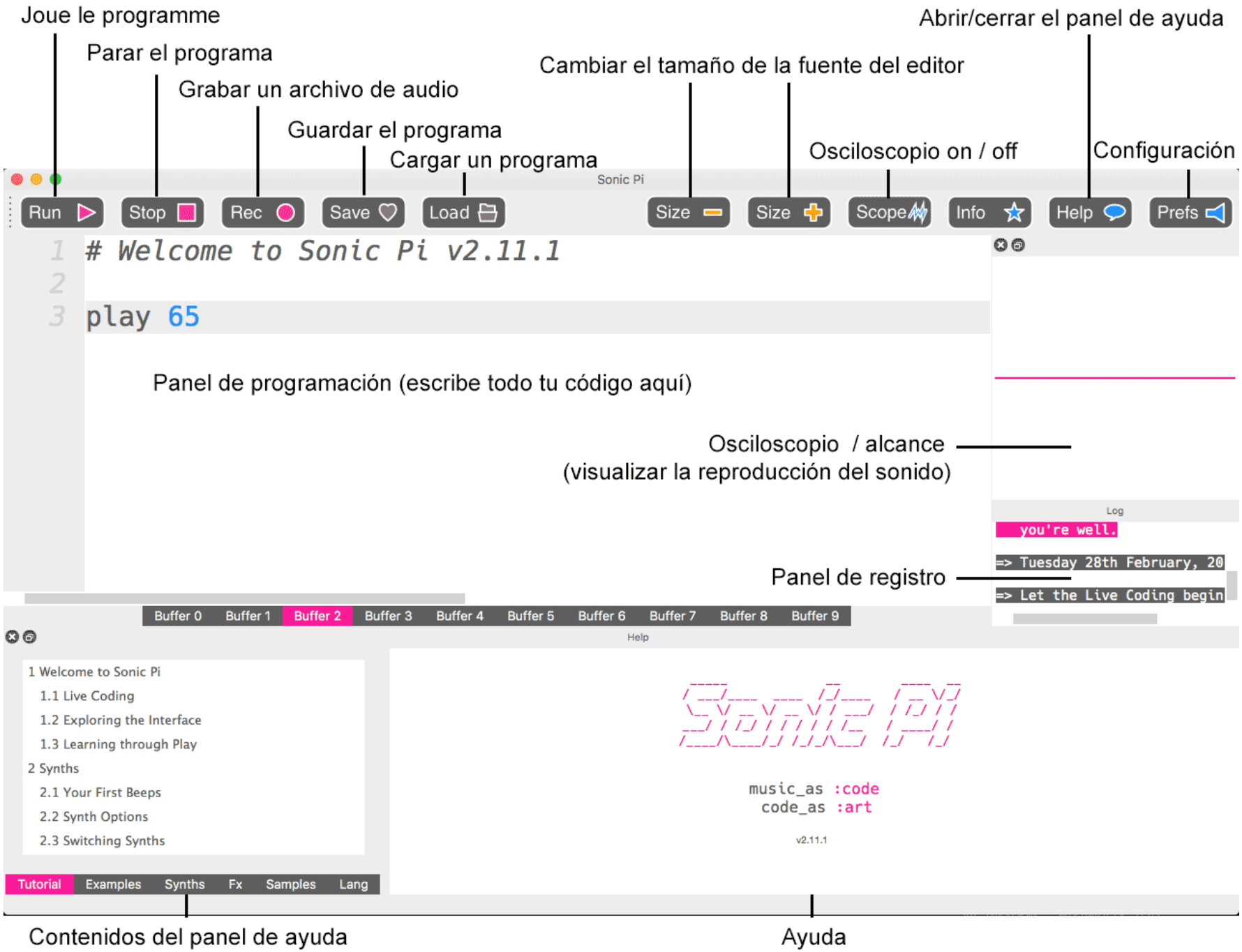
ARDUINO
Robótica & Electrónica

PROCESSING
Artes visuales &
Programación

SONIC PI
Música & Programación

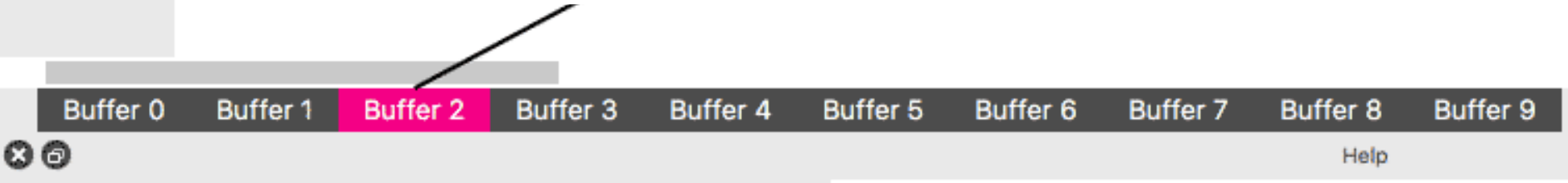
Descargar la aplicación:
sonic-pi.net

Material para auto-aprendizaje:
sonic-pi.mehackit.org


Joue le programme
Parar el programa
Grabar un archivo de audio
Guardar el programa
Cargar un programa
Cambiar el tamaño de la fuente del editor
Osciloscopio on / off
Abrir/cerrar el panel de ayuda
Configuración
Sonic Pi
Run
Stop
Rec
Save
Load
Size
Size
Scope
Info
Help
Prefs
1 # Welcome to Sonic Pi v2.11.1
2
3 play 65
Panel de programación (escribe todo tu código aquí)
Osciloscopio / alcance
(visualizar la reproducción del sonido)
Log
you're well.
=> Tuesday 28th February, 20
=> Let the Live Coding begin
Panel de registro
Buffer 0
Buffer 1
Buffer 2
Buffer 3
Buffer 4
Buffer 5
Buffer 6
Buffer 7
Buffer 8
Buffer 9
Help
1 Welcome to Sonic Pi
1.1 Live Coding
1.2 Exploring the Interface
1.3 Learning through Play
2 Synths
2.1 Your First Beeps
2.2 Synth Options
2.3 Switching Synths
Tutorial
Examples
Synths
Fx
Samples
Lang
music_as :code
code_as :art
v2.11.1
Contenidos del panel de ayuda
Ayuda

```
3 play 65
4
5
6
7
8
9
10
```

Los buffers (0-9) se pueden usar para guardar tus canciones de Sonic Pi.
¡También se pueden usar para probar rápidamente distintas ideas!



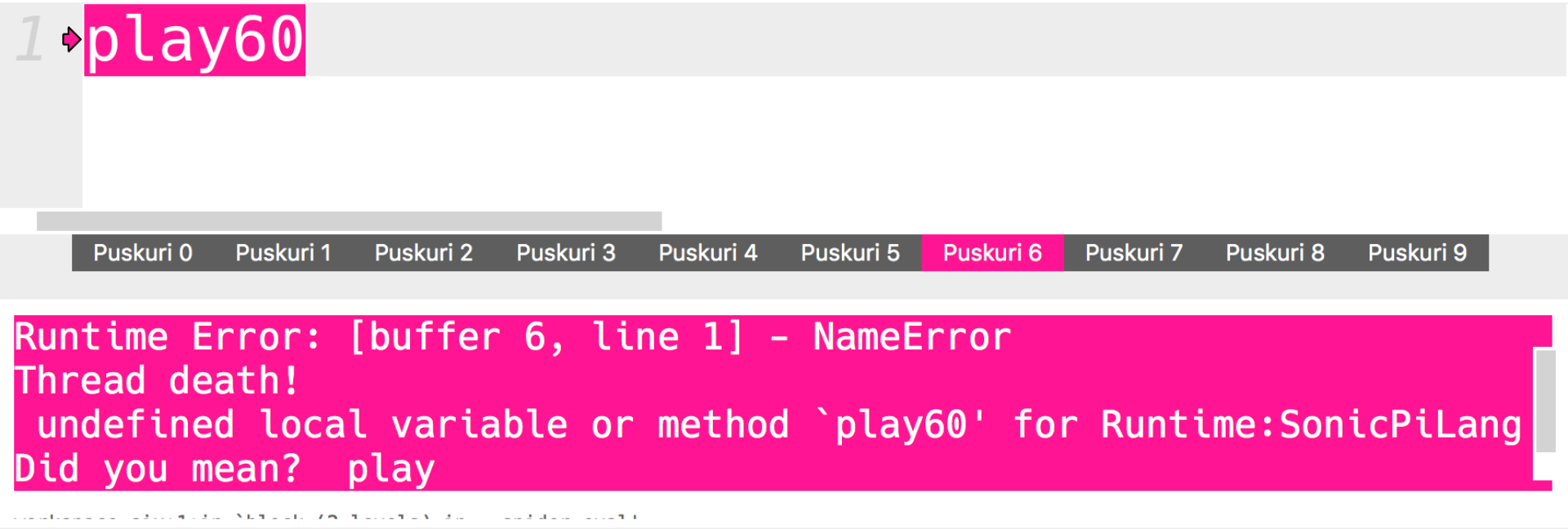
¡Tu primer Beep!

Escribe el siguiente comando:

```
play 60
```

… y presiona "RUN"

Es de humanos cometer errores...



... y la sintaxis en Sonic Pi es muy quisquillosa.

Tocando una melodía

Puedes usar números desde 0 a 127 como notas con el comando play.

Los números representan notas MIDI.

¡Consejo! Puedes tocar también tonos microtonales. ¿Alguien adivina cómo?

¡No se lo enseñéis a los niños!

Octave number	Note number											
	C	Db	D	Eb	E	F	Gb	G	Ab	A	Bb	B
-1	0	1	2	3	4	5	6	7	8	9	10	11
0	12	13	14	15	16	17	18	19	20	21	22	23
1	24	25	26	27	28	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71
5	72	73	74	75	76	77	78	79	80	81	82	83
6	84	85	86	87	88	89	90	91	92	93	94	95
7	96	97	98	99	100	101	102	103	104	105	106	107
8	108	109	110	111	112	113	114	115	116	117	118	119
9	120	121	122	123	124	125	126	127				

Pero es buenos saber de dónde
vienen estos números en el mundo
del MIDI

Tocando una melodía

```
play 60
sleep 1
play 64
sleep 1
play 67
```

Alterando el ritmo

```
play 60
sleep 1.5
play 64
sleep 0.5
play 67
```

Consejo útil # 1

Usa atajos de teclado para hacer "RUN" y "STOP"

Mac: CMD + R / CMD + S
Win: Alt + R / Alt + S

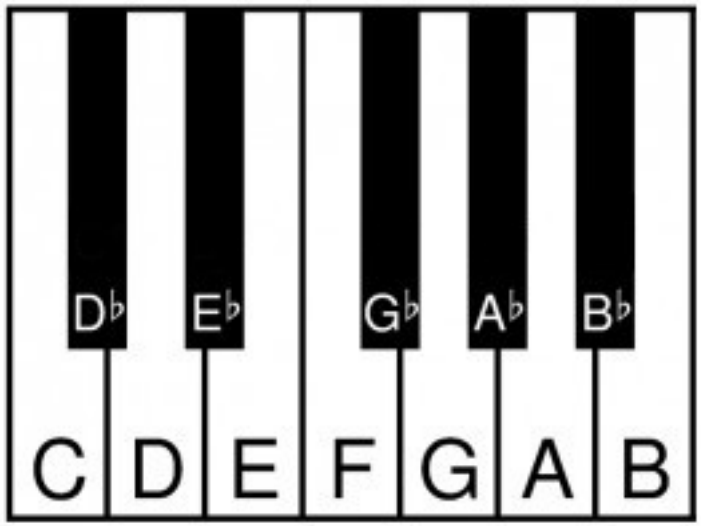
Cambiando el tiempo

Añade el siguiente comando al comienzo de tu programa:

```
use_bpm 120
```

¿Qué pasa ahora con valores como 400 or 80?

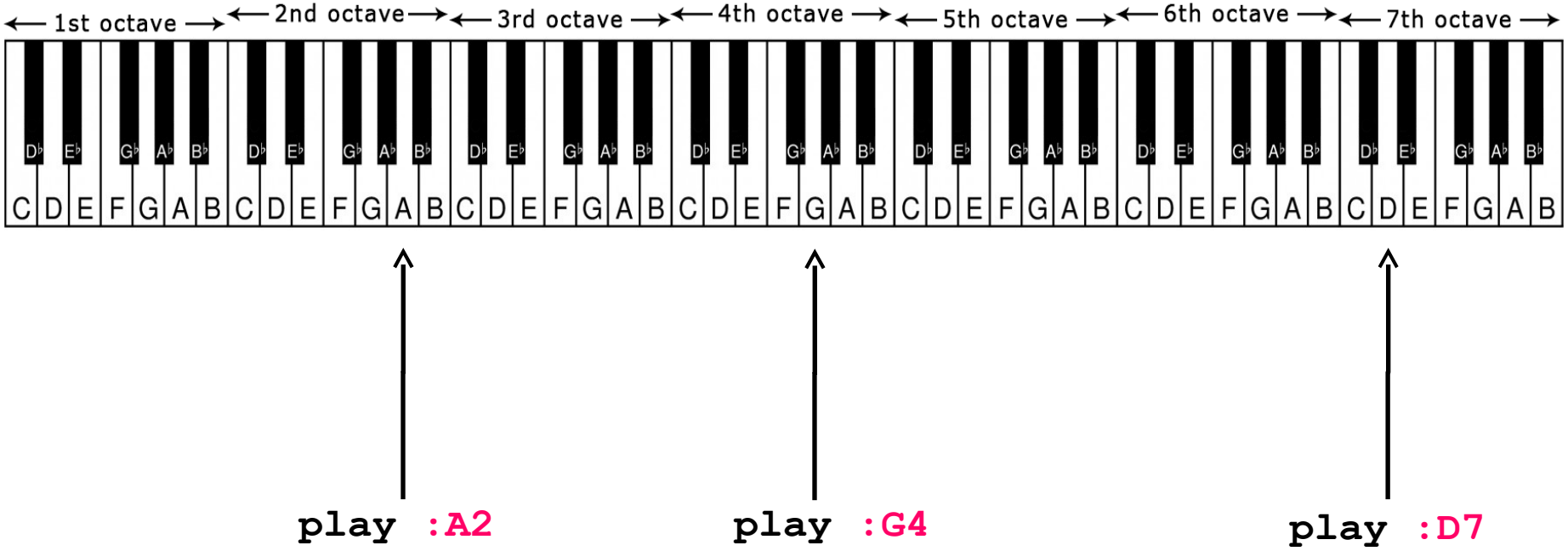
Notación en Sonic Pi



Los siguientes “símbolos de notas” se pueden usar con el comando play:

:C, :Db, :D, :Eb, :E, :F, :Gb, :G, :Ab, :A, :Bb, :B

Tocando notas de diferentes octavas.



¡Simplemente añade un número después del símbolo de la nota! Por ejemplo, play **:C4**

Usando notación en su lugar de números

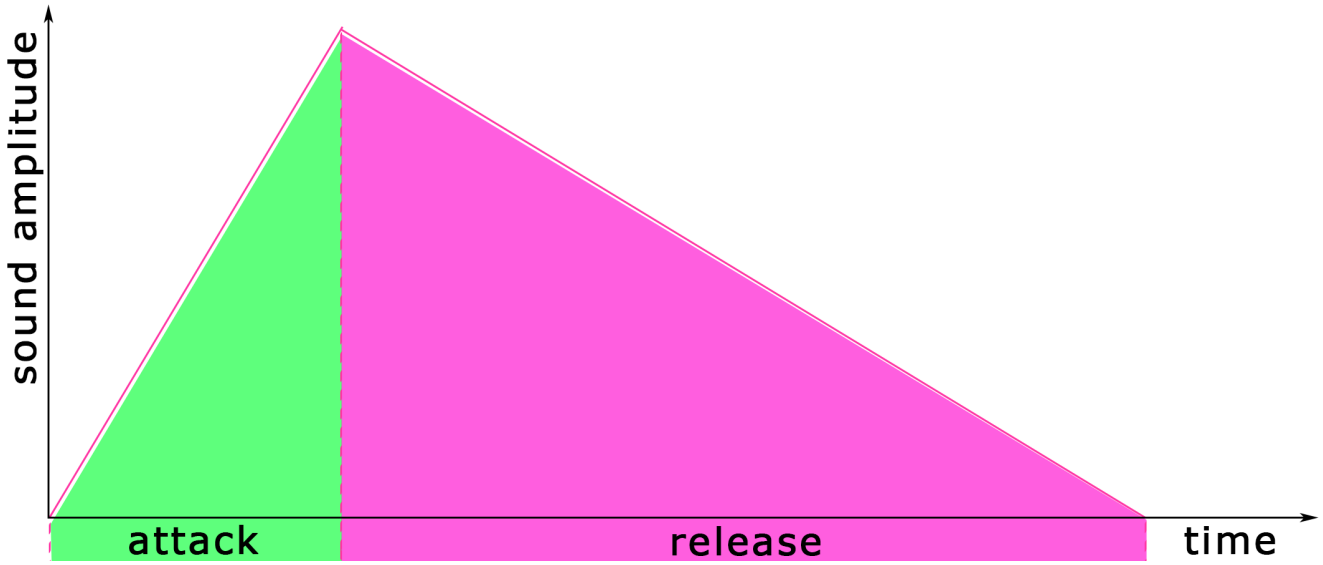
```
play :C3
sleep 1
play :E4
sleep 1
play :G5
```

Consejo útil # 2

Si para ti es más fácil escribir melodías con un teclado, puedes usar un teclado desde el navegador como este:

http://sonic-pi.mehackit.org/exercises/es/09-keys-chords-and-scales/01-piano.html

Duración de una nota




```
play :C4, attack: 1, release: 2
```

Estas son las opciones para el comando play

Cambiando tu sintetizador de sonido

```
use_synth :piano
play :C4
sleep 0.25
use_synth :pulse
play :C2
sleep 0.25
use_synth :chiplead
play :G3
sleep 0.25
```

Añadiendo algunas dinámicas con la opción amp

Por ejemplo:

```
play :C4, amp: 0.25
```
25% volumen

```
play :C4, amp: 0.5
```
50% volumen

```
play :C4, amp: 1
```
100% volumen

```
play :C4, amp: 2
```
200% volumen

”Practica tu play”

Pasa unos 15 minutos escribiendo un programa con Sonic Pi que toque una pequeña pieza melódica.

Guardando tu canción

- Puedes guardar tu canción como un archivo de audio WAV presionando el botón REC y empezando la reproducción (pulsando RUN). Una vez que se reproduzca la canción, pulse STOP y luego presione REC de nuevo.

- Si quiere hacer que el audio se escuche más fuerte y necesitas la canción en formato MP3, encontrarás instrucciones sencillas de como hacerlo en http://sonic-pi.mehackit.org y seleccionando el tema.

Repitiendo frases

```
play :C4
sleep 1
4.times do
    play :E4
    sleep 0.5
    play :G4
    sleep 0.5
end
```

Las áreas seleccionadas se llaman “bloques de código”

Bloques dentro de bloques

```
play :C4
sleep 1
4.times do
    play :E4
    sleep 0.5
    2.times do
      play :G4
      sleep 0.25
      play :B4
      sleep 0.25
    end
end
```

¡Puedes “anidar” tantos bloques dentro de otro bloque como quieras!

Tocando acordes

Por ejemplo:

```
play chord(:C4, :major)
```

or

```
play [:C4, :E4, :G4]
```

Este tipo de estructura se llama “tabla” en programación.

Tocando muestras

Por ejemplo:

```
sample :bd_fat
sample :ambi_piano
sample :ambi_choir
```

¡Pssst! ¡Usa un buffer vacío para probar las muestras!

¡Opciones de los muestras de audio que necesitas conocer!

Por ejemplo:

```
sample :bd_fat, amp: 0.5

sample :ambi_choir, rate: 0.5

sample :misc_crow, pan: -1

sample :ambi_choir, start: 0.7
```

”Una pausa de 10 minutos”

¡Vamos a pasar 10 minutos familiarizándonos con las muestras de audio de Sonic Pi! Por ejemplo, haz unos toques de tambor en un buffer vacío!

Creando bucles y tocando sonidos de forma concurrente

Concurrencia en programación

Los programas que acabamos

de escribir son programas lineales simples

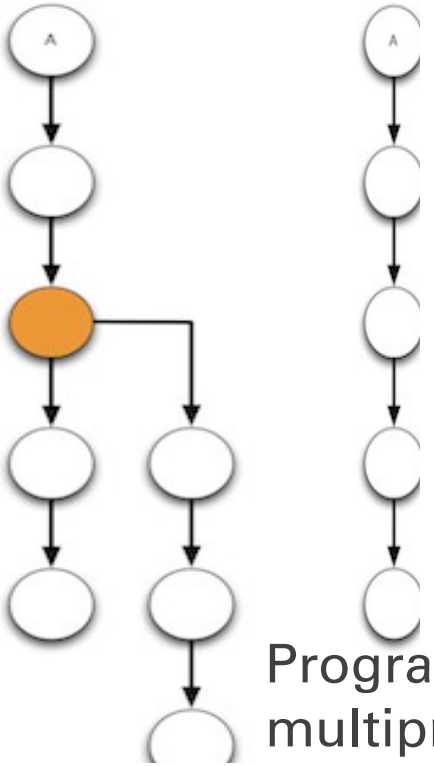
¿Es un poco problemático escribir música así?

Concurrencia en programación

Los programas que acabamos

de escribir son programas lineales simples

¿Cómo llegamos a esto?



Programación multiproceso

Creando hilos con live_loop

```
live_loop :rummut do
    sample  :bd_haus, amp: 1.5
    sleep 1
    sample  :sn_dolf
    sleep 1
end

live_loop :hihat do
    sample  :drum_cymbal_closed
    sleep 0.25
end
```

TIME IN BEATS

1

```
live_loop :kickdrum do
  sample :bd_haus
  sleep 1
end
```

```
live_loop :cowbell do
  sample :drum_cowbell
  sleep 0.5
end
```

```
live_loop :cowbell do
  sample :drum_cowbell
  sleep 0.5
end
```

```
live_loop :bassline do
  use_synth :fm
  sleep 0.5
  play :E2
  sleep 0.5
end
```

```
live_loop :guitar do
  sample :guit_em9
  sleep 2
end
```

2

```
live_loop :kickdrum do
  sample :bd_haus
  sleep 1
end
```

```
live_loop :cowbell do
  sample :drum_cowbell
  sleep 0.5
end
```

```
live_loop :cowbell do
  sample :drum_cowbell
  sleep 0.5
end
```

```
live_loop :bassline do
  use_synth :fm
  sleep 0.5
  play :E2
  sleep 0.5
end
```

3

```
live_loop :kickdrum do
  sample :bd_haus
  sleep 1
end
```

```
live_loop :cowbell do
  sample :drum_cowbell
  sleep 0.5
end
```

```
live_loop :cowbell do
  sample :drum_cowbell
  sleep 0.5
end
```

```
live_loop :bassline do
  use_synth :fm
  sleep 0.5
  play :E2
  sleep 0.5
end
```

```
live_loop :guitar do
  sample :guit_em9
  sleep 2
end
```

4

```
live_loop :kickdrum do
  sample :bd_haus
  sleep 1
end
```

```
live_loop :cowbell do
  sample :drum_cowbell
  sleep 0.5
end
```

```
live_loop :cowbell do
  sample :drum_cowbell
  sleep 0.5
end
```

```
live_loop :bassline do
  use_synth :fm
  sleep 0.5
  play :E2
  sleep 0.5
end
```

5

```
live_loop :kickdrum do
```

```
live_loop :cowbell do
```

```
live_loop :bassline do
```

```
live_loop :guitar do
```

“Bucles infinitos” - live_loop

- Puedes tener múltiples live_loops ejecutándolos simultáneamente.
- Hace posible tener múltiples hilos de código sincronizados ejecutándose en Sonic Pi.
- Cada live_loop necesita un nombre único y al menos un comando sleep

Comentarios en el código

Puedes comentar en una línea de código añadiendo el carácter # al principio de la línea. Cuando presione “Run”, los comentarios no se ejecutarán.

```
live_loop :rummut do
    #sample  :bd_haus, amp: 1.5
    sleep 1
    sample  :sn_dolf
    sleep 1
end
```

“Ejercicio con bucles”

¡Crea un programa con Sonic Pi que tenga al menos tres live_loops sonando al mismo tiempo!

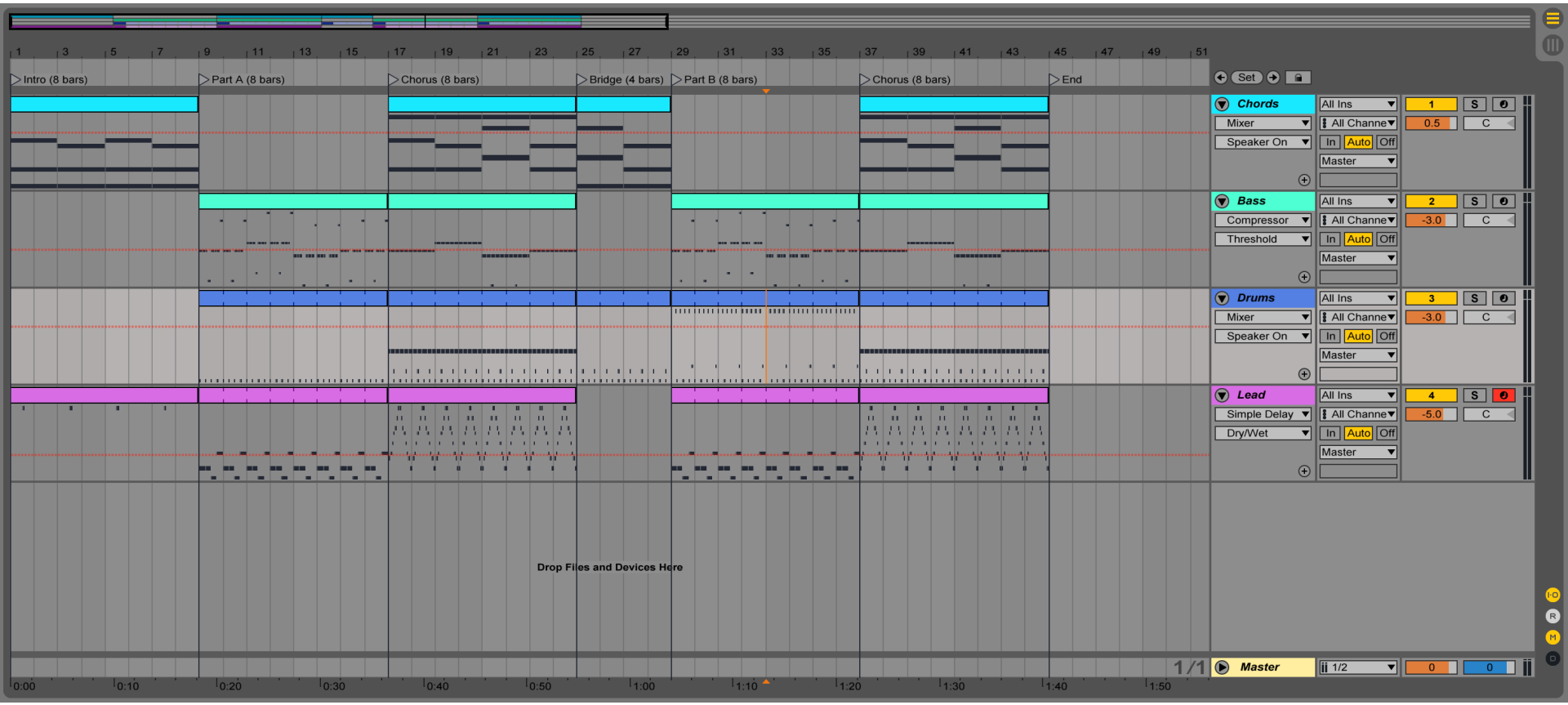
Creando hilos con in_thread

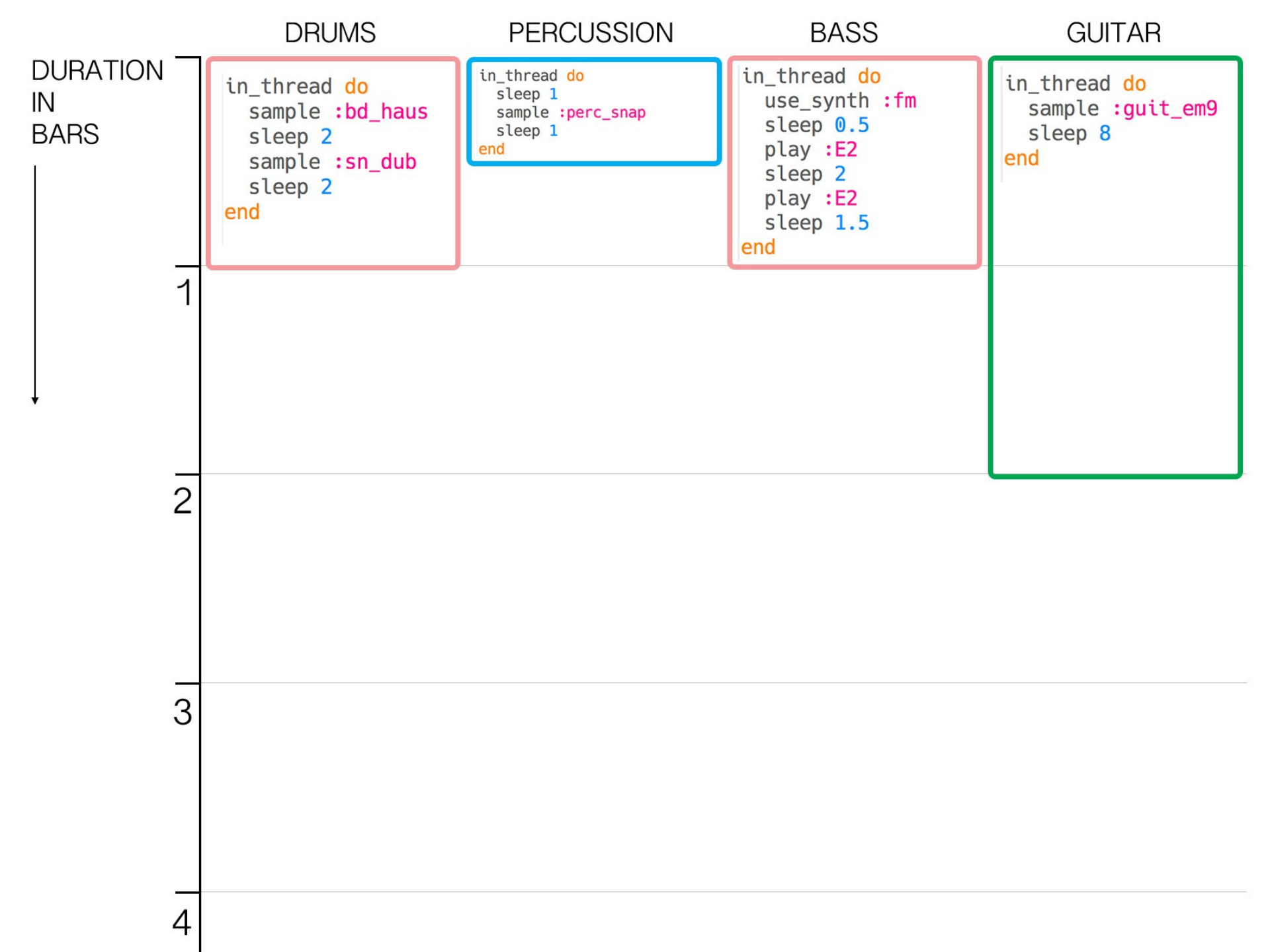
- Con in_thread puedes hacer un arreglo convencional de una forma muy poco convencional :-)

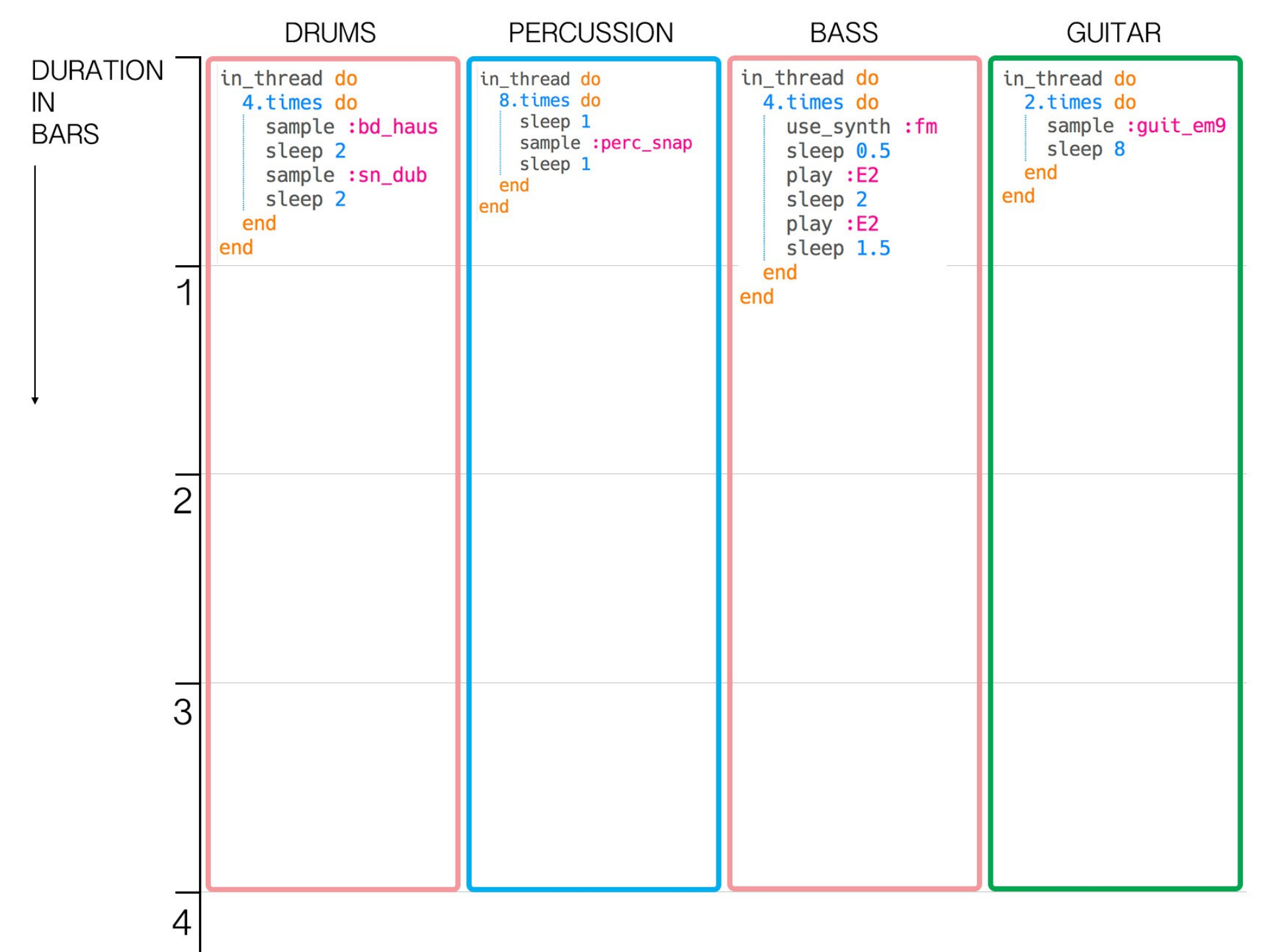
- Nota: los bloques in_thread no necesitan un nombre como con live_loops

- Construir canciones con Sonic Pi requiere algo de paciencia. ¡Necesitarás hacer un buen seguimiento de la duración de cada parte de la canción! ¡Necesitas conocer los valores de sleep!
  4 compases = sleep 16, 8 compases = sleep 32
  16 compases = sleep 64 ... y así

Arreglo normal de DAW (estación de trabajo de audio digital)




DRUMS
PERCUSSION
BASS
GUITAR
DURATION IN BARS
in_thread do
sample :bd_haus
sleep 2
sample :sn_dub
sleep 2
end
in_thread do
sleep 1
sample :perc_snap
sleep 1
end
in_thread do
use_synth :fm
sleep 0.5
play :E2
sleep 2
play :E2
sleep 1.5
end
in_thread do
sample :guit_em9
sleep 8
end
1
2
3
4


DRUMS
PERCUSSION
BASS
GUITAR
DURATION IN BARS
in_thread do
4.times do
sample :bd_haus
sleep 2
sample :sn_dub
sleep 2
end
end
in_thread do
8.times do
sleep 1
sample :perc_snap
sleep 1
end
end
in_thread do
4.times do
use_synth :fm
sleep 0.5
play :E2
sleep 2
play :E2
sleep 1.5
end
end
in_thread do
2.times do
sample :guit_em9
sleep 8
end
end
1
2
3
4

Consejo para organizar

- Puedes tener tantos in_threads como quieras ejecutándose concurrentemente. En cada parte de la canción asegúrate que sus longitudes coinciden (por ejemplo, tienen la misma cantidad de sleep)

- Edita cada parte de la canción en buffers separados y únelos luego en un buffer principal. Cuando presiones “Run” probablemente no querrás siempre escuchar la canción desde el principio. Esto te ahorrará un tiempo precioso.

- Haz comentarios para marcar los puntos de partida de cada parte diferente de la canción.

- Aunque parezca obvio: ¡guarda tu código con frecuencia!

Algunos temas más avanzados sobre Sonic Pi

Aleatorización (1/2) usando el comando choose

```
live_loop :randomMelodia do
    use_synth :chipbass
    play [:C3, :Eb5, :G4, :Bb4].choose
    sleep 0.25
end

live_loop :randomSleep do
    sample :elec_blip, amp: 2
    sleep [0.25, 0.5, 0.75].choose
end
```

Aleatorización (2/2) usando el comando rrand

```
live_loop :trance do
    use_synth :tb303
    play [:C2, :C3].choose, cutoff: rrand(50, 120), release: 0.25
    sleep 0.25
end

live_loop :hihat do
    sample :drum_cymbal_closed, amp: rrand(0,2)
    sleep 0.25
end
```

Samples externos

- Puedes tocar cualquier archivo de audio de tu computadora con Sonic Pi. De esta forma puedes usar tus propios samples guardados y destacar tu propia música

- ¡Sólo tienes que decirle a Sonic Pi donde tienes tus archivos de audio!

- Puedes encontrar un ejemplo aquí: http://sonic-pi.mehackit.org/exercises/es/05-advanced-topics-1/04-external-samples.html

Variables

En programación, una variable es como una caja o un contenedor donde pones cosas y las recuperas o las cambias más tarde. Usando variables también puedes ahorrarte el tiempo que usas repitiendo código

Por ejemplo:

```
bmin = chord(:B4, :minor)

play bmin
sleep 1
play bmin
```

Secuenciador de notas

```
live_loop :bassline do
    use_synth :tb303
    notes = [:C2, :C2, :Eb2, :Bb2].ring.tick
    play notes, release: 0.25
    sleep 0.25
end
```

Secuenciador de notas + filtro de paso-bajo

```
live_loop :bassline do
    use_synth :tb303
    notes = [:C2, :C2, :Eb2, :Bb2].ring.tick
    play notes, release: 0.25, cutoff: rrand(60, 130)
    sleep 0.25
end
```

Efectos en pocas palabras

```
with_fx :reverb do
    # your code here
end

with_fx :echo do
    # your code here
end

with_fx :distortion do
    # your code here
end
```

Ejemplos de efectos

```
with_fx :reverb, room: 0.9 do
    sample :elec_beep
    sleep 1
    sample :misc_crow
    sleep 1
end
```

El efecto se aplica a cada sonido que se toca en un bloque (entre do y end)

Pssst! Puedes también “anidar” efectos, es decir, ¡puedes crear cadenas de efectos!

play_pattern_timed

```
play :c2
sleep 0.5
play :d2
sleep 0.25
play :e2
sleep 0.75
play :d2
sleep 0.5
```

Puedes guardar muchas líneas de código usando play_pattern_timed

```
play_pattern_timed [:c2, :d2, :e2, :d2], [0.5, 0.25, 0.75, 0.5]
```

Sentencias condicionales simples

En programación esto signif ica básicamente realizar acciones dependiendo del estado de una condición. En el ejemplo siguiente, el valor de la variable playdrums establece la condición.

```
playdrums = 0

with_fx :reverb, room: 0.9 do
    sample :bd_haus if playdrums == 1
    Sleep 0.5
end
```

El tiempo en géneros musicales electrónicos

Ambient 50–100 BPM

Hip-hop 70–95 BPM

Deep house 110–130 BPM

Trance / Techno 130–145 BPM

Hard dance/hardcore 145–170 BPM

Drum and bass 160–180 BPM

Ejercicio final: ¡Haz una canción corta y diviértete!

¡Puede ser, por ejemplo, un bucle hecho con con cuatro live_loops. Un live_loop por cada instrumento: tambores, bajos, melodía de sintetizador y samples divertidos!

¡O puedes intentar organizar una canción simple con in_thread!

Consejos para la clase (1/3)

- Pinte un dibujo grande / Explique el objetivo
- Mucho de los estudiantes carecen de conocimientos básicos de TIC
- Los jóvenes probarán toda clase de experimentos raros con Sonic Pi
- No puede saberlo todo sobre Sonic Pi
- Un buen equipo de sonido es una gran ventaja

Consejos para la clase (2/3)

- Recuerde a sus estudiantes que guarden el contenido de los buffers
- Con la práctica se desenvolverá mejor con Sonic Pi
- Los alumnos pueden trabajar en parejas (usando divisores de auriculares)
- Haga un plan de clase para introducir nuevos conceptos

Consejos para la clase (3/3)

- Tenga una carpeta compartida donde intercambiar parches con sus alumnos (use por ejemplo Google Drive o Dropbox)
- Traducir canciones populares es difícil
- Diviértete, comete errores y juega con sonidos tontos
- ¡Dile a los jóvenes que Sonic Pi es libre y que se puede instalar en su computadora en casa!

Preparaciones para la clase

- Equipamiento en orden (computadoras, auriculares, divisor de auriculares (si hace falta), proyector y altavoces)
- Comprueba que Sonic Pi funciona en computadoras con el perfil del alumno. A veces los antivirus o la configuración del cortafuegos en Windows puede causar problemas.
- ¡Conecta todos los auriculares a las computadoras antes de que empiece la clase!
- Desafortunadamente los Chromebooks no son compatibles.

¿Necesitas ayuda?

- Crea un grupo en Facebook para este proyecto donde puedes hablar de Sonic Pi
- Una comunidad activa con cientos de usuarios de Sonic Pi se puede visitar en https://in-thread.sonic-pi.net/