



Especificación

DISEÑO E IMPLEMENTACIÓN DE UN LENGUAJE

v1.0.0

1. Equipo	1
2. Repositorio	1
3. Dominio	2
4. Construcciones	2
5. Casos de Prueba	3
6. Ejemplos	3

1. Equipo

Nombre	Apellido	Legajo	E-mail
Valentin	Ontivero	60034	vontivero@itba.edu.ar
Pablo	Germano	62837	pgermano@itba.edu.ar
Mauro	Vella	62134	mvella@itba.edu.ar
Federico	Kloberdanz	62890	fkloberdanz@itba.edu.ar

2. Repositorio

La solución y su documentación serán versionadas en:
<https://github.com/valenontivero/TP-TLA>

3. Dominio

El dominio de este trabajo consiste en un lenguaje específico para la creación de patrones musicales en una caja de ritmos (también conocido como Drum Machine).

El lenguaje permite definir instrumentos de percusión (bombo, snare, hi-hat, platillo, etc.), instrumentos melódicos (bajo, teclado, etc.), especificar el patrón rítmico de cada uno, el volumen de cada uno (decibelios) y especificar el tempo general del programa (BPM). También soportará bucles (loops), variaciones, silencios, y la posibilidad de concatenar o superponer patrones para construir ritmos más complejos.

La idea es que cualquier ritmo programado pueda ser transformado a un formato de audio (.wav, MIDI), permitiendo a músicos y programadores generar patrones rítmicos a partir de un lenguaje (código), probar variaciones y experimentar con infinitas combinaciones de beats, además de la implementación de algunos instrumentos musicales (piano, bajo) para ampliar aún más las posibilidades de este dominio.

4. Construcciones

El lenguaje desarrollado debería ofrecer las siguientes construcciones, prestaciones y funcionalidades:

1. Definir tempo, indicando los Beats Por Minuto (BPM) (e.g. tempo 120)
2. Soportar la exportación de los patrones finalizados en audio.
3. Declarar instrumentos de percusión (e.g. kick, snare, hihat)
4. Declarar instrumentos melódicos (e.g. teclado, bajo)
5. Crear patrones rítmicos con pasos de compás (pattern{...}).
6. Soportar silencios (-), simples o prolongados
7. Soportar intensidad de golpe (con operadores relacionales)
8. Concatenar patrones (+)
9. Iterar patrones (*n repeticiones)
10. Aplicar transformaciones rítmicas (e.g. doble tempo, medio tempo)
11. Definir variaciones dentro de patrón (e.g. alternar un golpe en cada repetición con función)
12. Reutilizar patrones con variables (pattern groove = {...})
13. Guardar y cargar patrones desde archivos para reutilizarlos en distintos programas.
14. Crear patrones rítmicos con pasos de compás (pattern con subdivisiones en corcheas, semicorcheas, etc.)

5. Casos de Prueba

Se proponen los siguientes casos iniciales de prueba de aceptación:

- (I). Un programa que crea un ritmo con tempo 120 bpm y en 4/4.
- (II). Un programa que agrega ritmos de percusión: bombo y de snare.
- (III). Un programa que agrega ritmos melódicos: bajo y teclado.
- (IV). Un programa que agrega ritmos mixtos: bajo y kick.
- (V). Un programa que carga un sample de kick para utilizar en su ritmo.
- (VI). Un programa que pueda definir patrones particulares para reutilizar.
- (VII). Un programa que soporte una progresión con instrumentos y silencios.
- (VIII). Un programa que combina dos patrones guardados en un solo ritmo.
- (IX). Un programa que permita el agregado (+) de diferentes ritmos para el mismo instrumento.
- (X). Un programa que permita la iteración (*) de diferentes ritmos para el mismo instrumento.
- (XI). Un programa que tenga patrones con diferentes figuras musicales (*e.g.* corcheas y negras).
- (XII). Un programa que exporte como .WAV el ritmo creado.

Además, los siguientes casos de prueba de **rechazo**:

- (I). Un programa que genere acordes o una composición compleja de notas *e.g.* ligaduras de guitarra.
- (II). Un programa que genere un ritmo sin instrumentos asignados.
- (III). Un programa que concatene patrones de distinta métrica.
- (IV). Un programa que defina un compás ó tempo inválido.
- (V). Un programa que defina el mismo instrumento en múltiples patrones.
- (VI). Un programa que contenga un patrón fuera de la métrica 4/4.

6. Ejemplo conceptual

Creación de un patrón que imita el ritmo de la canción “We Will Rock You” de Queen. Este patrón sonará en loop hasta finalizar el programa. Este programa tiene las funcionalidades más fáciles y rápidas de implementar.

```
tempo{160};
compasses{1};
steps{1};
instruments{
  kick{
    Pattern { [x,x,-,-] };
    Duration{1};
  }
  snare{
    pattern{ [-,-,x,-] };
    duration{1};
  }
}
```

Creación de un patrón que imita la melodía y ritmo de la canción “Smoke on the Water” de Deep Purple. Este programa crea un patrón con bombo (kick) y bajo (bass) y los patrones sonarán en loop hasta finalizar el programa .

```
tempo{115};
compasses{4};
steps{1};
Instruments{
  kick{
    pattern{[x, -, x, -] * 4};
    duration{1};
  }
  bass{
    Pattern { [E2,G2,A2, -] + [E2,G2,A#2,A2] + [E2,G2,A2, -] + [G2, E2, -, -] };
    duration{1};
  }
}
```

Aclaración de las palabras reservadas:

- “tempo” es la velocidad general a la que sucederá el ritmo.
- “compasses” la cantidad de compasses que se loopearán en total.
- “steps” es la cantidad de golpes utilizables por cada pulso (beat).
- “instrumentos” es aquí se listan los instrumentos que aparecerán dentro del ritmo.
- “pattern” es el patrón de notas (o golpes) que seguirá el instrumento. En este caso el kick es 4 veces golpes en el primer y tercer pulso de cada compás.
- “duration” es la cantidad de "steps" que mantiene sonando cada símbolo del "pattern".