# Proyecto individual: Batería Virtual en Python

# Alejandro Brenes Calderón

C21319

**Curso: CA-0305** 

### 1. Introducción

Para el desarrollo del proyecto individual se tuvo la intención inicial de enfocarlo a la parte musical, más específicamente, a la parte de percusión. Se buscaba implementar en Python una herramienta que permitiera al usuario entender más a fondo los conceptos detrás de la estructura musical, de manera simple y visualmente atractiva. Así, el proyecto desarrollado es una herramienta didáctica y pedagógica diseñada para reproducir sonidos de batería de manera intuitiva. La idea principal es proporcionar una plataforma para experimentar con ritmos de percusión, ayudando tanto a principiantes como a músicos experimentados a mejorar su comprensión y habilidades en la percusión. La herramienta está desarrollada en Python utilizando la librería Pygame, lo que permite crear una interfaz gráfica interactiva y reproducir sonidos de batería en tiempo real, apoyándose además en Mixer de Pygame.

# 2. Objetivos

- Crear una herramienta interactiva y visualmente atractiva para reproducir sonidos de batería.
- Facilitar el aprendizaje de ritmos de percusión mediante una interfaz intuitiva.
- Proporcionar una plataforma flexible donde los usuarios puedan crear y guardar sus propios ritmos.

# 3. Metodología

# 3.1. Estructura del Código

El código está organizado en una clase principal llamada "Bateria\_virtual", la cual contiene toda la funcionalidad de la herramienta. Esta clase maneja la inicialización de Pygame, la carga de sonidos, la creación de la interfaz gráfica y la lógica para la reproducción de ritmos.

### 3.2. Librerías Utilizadas

- **Pygame**: Librería especializada en crear la interfaz gráfica y manejar la reproducción de sonidos.
- **Pygame.mixer**: Utilizada específicamente para manejar los sonidos de los instrumentos de batería, los cuales vienen en formato WAV.

### 3.3. Funcionalidades Principales

#### Inicialización

La inicialización de la aplicación, al principio de la clase, implica configurar el tamaño de la pantalla, definir una paleta de colores para la interfaz, y cargar los sonidos de los diferentes elementos de la batería (hi-hat, redoblante, bombo, crash, tom y una simulación de un aplauso). Esta configuración inicial es esencial para asegurar que la aplicación funcione correctamente y que la interfaz sea visualmente coherente y atractiva.

### Configuración de la Interfaz Gráfica

La interfaz gráfica de la Batería Virtual está diseñada para ser intuitiva y fácil de usar. La pantalla principal presenta una cuadrícula en la que cada celda representa un beat específico para un sonido de un instrumento en particular. El usuario puede activar o desactivar estos beats haciendo clic en las celdas correspondientes, permitiéndoles crear y modificar ritmos de manera interactiva. Es decir, funcionan a manera de botón que la persona puede estripar dependiendo si desea que haya un sonido en ese espacio. La interfaz también incluye controles para ajustar los BPM (beats por minuto) y el número de beats en loop, proporcionando un alto grado de personalización.

## Reproducción de Ritmos

La lógica de reproducción de los ritmos está implementada de manera que los sonidos se reproduzcan de forma sincronizada y sin interrupciones. El código se encarga de iterar sobre los beats activados en la cuadrícula y reproducir los sonidos correspondientes en los momentos adecuados, basándose en los BPM seleccionados por el usuario.

### 4. Resultados

La herramienta desarrollada cumple con los objetivos planteados, proporcionando una interfaz gráfica intuitiva y funcional para la creación y reproducción de ritmos de batería. Los usuarios pueden fácilmente interactuar con la cuadrícula de beats, ajustar el BPM, y

guardar o cargar ritmos. La implementación de clases y la organización del código en métodos específicos para cada funcionalidad proporciona un orden y una lógica más estructurada en el código.

#### 6. Conclusión

El proyecto de la Batería Virtual en Python demuestra el potencial de Pygame para crear herramientas educativas interactivas. La combinación de una interfaz gráfica intuitiva con la capacidad de reproducir sonidos en tiempo real ofrece una plataforma valiosa para el aprendizaje y la experimentación en el ámbito de la percusión. Con futuras mejoras y expansiones, tales como ampliar esta idea a más instrumentos como guitarra o piano, esta herramienta puede convertirse en un recurso aún más poderoso para músicos y profesores de música.

### 7. Referencias bibliográficas

```
pygame news. (2024). Pygame.org. https://www.pygame.org/news

pygame.mixer — pygame v2.6.0 documentation. (2023). Pygame.org.

https://www.pygame.org/docs/ref/mixer.html
```

Real Python. (2019, Setiembre 16). PyGame: A Primer on Game Programming in Python. Realpython.com; Real Python. https://realpython.com/pygame-a-primer/

Python Programming Tutorials. (2024). Pythonprogramming.net. https://pythonprogramming.net/adding-sounds-music-pygame/

Cervera, A. (2022, May 25). ¿Qué Es Un Archivo WAV Y Cómo Se Usa? Wondershare.es;

Wondershare Recoverit. https://recoverit.wondershare.es/audio-recovery/what-is-wav-file.html

plemaster01/LeMasterTechYT. (2024). GitHub.

https://github.com/plemaster01/LeMasterTechYT

 $pygame.draw - pygame\ v2.6.0\ documentation.\ (2023).\ Pygame.org.$ 

https://www.pygame.org/docs/ref/draw.html

rasql/pygame-tutorial: Introduction to making object-oriented apps with Pygame. (2024).

GitHub. https://github.com/rasql/pygame-tutorial