lzOu u



**Proyecto Final**

Departamento de Sistemas Electronicos

Materia: Programacion III

Maestro: Jose De Jesus Palos Garcia

Grado y grupo: 4ºA

Fecha de entrega: Lunes 10 de junio de 2024

Alumnos:

Ariel Emilio Parra Martínez ID: 280862

Miguel Angel Batres Luna ID: 350553

**Universidad autónoma de Aguascalientes.**

**Centro de Ciencias Básicas**

**Ingeniería en Sistemas Computacionales**

**Aguascalientes, Aguascalientes.**

**“Juego Tipo Sigue el Ritmo en JAVA”**

# Cronograma de actividades

###### semana 0, del 10 al 12 de abril

La semana donde se dejo el proyecto la consideraremos la base logistica del proyecto, creando un repositorio en Github para trabajar en conjunto, creando el cronograma de actividades, creando y llenando partes del documento donde se definira la base del proyecto.

En cuestion de codigo empezaremos con el esqueleto del proyecto creando las clase “menu”, tambien implementaremos los fondos de menus y el movimiento entre menus.

###### semana 1, del 15 al 19 de abril

La primer semana de vacaciones, empezaremos la logica del menu de configuracion donde el usuario podra configurar el tamano de la ventana y el volumen.

###### semana 2, del 22 al 26 de abril

La segunda semana de vacaciones implentaremos las variables globales que se usaran el juego principal.

###### semana 3, del 29 de abril al 3 de mayo

La primer semana regresando, se hara la primer entrega donde entregaremos este documento junto con las caracteristicas del proyecto, finalizaremos las configuraciones.

###### semana 4, del 6 al 10 de mayo

Continuaremos con el modo de un jugador e implementaremos los puntajes y su logica para poder tener el 30% del proyecto.

###### semana 5, del 13 al 17 de mayo

Se hara la segunda entrega de avance del proyecto a un aproximado del 30% el dia 13 y empezaremos la logica del editor de niveles donde cargaremos una cancion, habra botones de control y se eligira el tiempo donde poner las teclas de juego.

###### semana 6, del 20 al 31 de mayo

Se avanzara el juego, se intentara divir la pantalla en dos para proceder al modo de dos jugadores.

###### semana 7, del 3 al 7 de junio

Se realzaran los detalles finales, intentaremos pulir todo lo posible y pondremos los niveles precargados, se terminara los detalles del menu, empezamos pruebas online.

###### Lunes 10 de junio

Entregaremos proyecto terminado con los puntos mencionados cubiertos

# Caracteristicas

**Modo de juego**: hay tres formas de jugar, un jugador, dos jugadres y online, en cualquier caso se puede pausar con la tecla “p” para tomar un descanso o para salir del juego , cuando se acaba la cancion o se sale voluntariamente se muestran los puntajes del juego.

**Temas**: El juego cuenta con cuatro canciones de musica estilo rock, siendo estas : Kung Fung Fu Fighting de Carl Douglas, Losing my Religion de R.E.M, Toxicity de System of a Down y Take me Out de Franz Ferdinand.

**Instrumentos**: las teclas seran notas de la guitarra como en guitarhero, usando las teclas “a,s,d,f” para el primer jugador y las teclas “h,j,k,l” para el segundo jugador.

**Escenarios**: Hay tres escenarios principales,Pang Tang Bay siendo usado por Kung Fung Fu Fighting y Toxicity, Grove Street para Losing my Religio y Amoeba Records para Take me Out.

**Modo Editor:** En el modo editor elegimos las fichas para poner usando las teclas del primer jugador “a,s,d,f” , para poner las fichas amarilla,azul,rojo,verde en dicho ordem, para quitarlas presionamos las teclas “q,w,e,r ” para quitar las fichas en el mismo orden, para poner fichas energizadas se preciona la tecla “z” y para volver a fichas normales se preciona “x”, para poder adelantar o retroceder se preciona la flechas adelante “→ ”y atras ”←” respectivamente, para pausar se presiona “p” y para guardar solo se sale del editor desde la pausa.

# Logica del Proyecto

### Explicación de clases

**Ficha:** La clase Ficha es una clase que representa una ficha del guitar hero:

* Almacena la posición de la ficha en el juego con coordenadas.
* Proporciona métodos para revalidar las posiciones de las fichas.
* Maneja la física de las fichas, probablemente cómo interactúan con el entorno o con otras fichas.

**Juego**:La clase Juego es una clase que maneja la lógica principal del juego, creando hilos de ejecución:

* Controla el estado del juego (por ejemplo, si el juego está en curso, en pausa, etc.).
* Maneja las interacciones del jugador y las entradas del usuario.
* Maneja tambien cambios del menu como lo seria con la pausa.
* Actualiza el estado del juego y los elementos del juego (como las fichas, los jugadores, etc.).
* Renderiza el juego y actualiza la interfaz del juego.

**Menu**:La clase Menu es una clase que maneja la interfaz gráfica de usuario (GUI) dentro del proyecto:

* Carga una fuente personalizada desde un archivo y la registra en el entorno gráfico.
* Crea un JLayeredPane, que es un contenedor que permite superponer componentes (como paneles).
* Crea una instancia de VideoPanel, que probablemente es un panel personalizado para mostrar videos.
* Crea una nueva instancia de Juego, que probablemente maneja la lógica del juego.
* Crea varios paneles (como menuPanel, configPanel, pausaPanel, jugarPanel, configJuego), los cuales son diferentes secciones de la interfaz de usuario del juego.

**Main:** Activa el menu y por ende todo el proyecto.

**BotonBase**: La clase BotonBase sirve para crear botones que:

* Almacena una imagen que se muestra en el botón.
* Mantiene un estado para saber si el botón está presionado o no.
* Permite cambiar la imagen y el estado del botón.
* Puede dibujar la imagen en el botón.
* Define el tamaño preferido del botón

**CustomFontLoader**: La clase CustomFontLoader es una clase que se utiliza para cargar fuentes personalizadas en el proyecto.

* Lee una fuente desde un archivo.
* Registra la fuente en el entorno gráfico de la aplicación para que pueda ser utilizada.

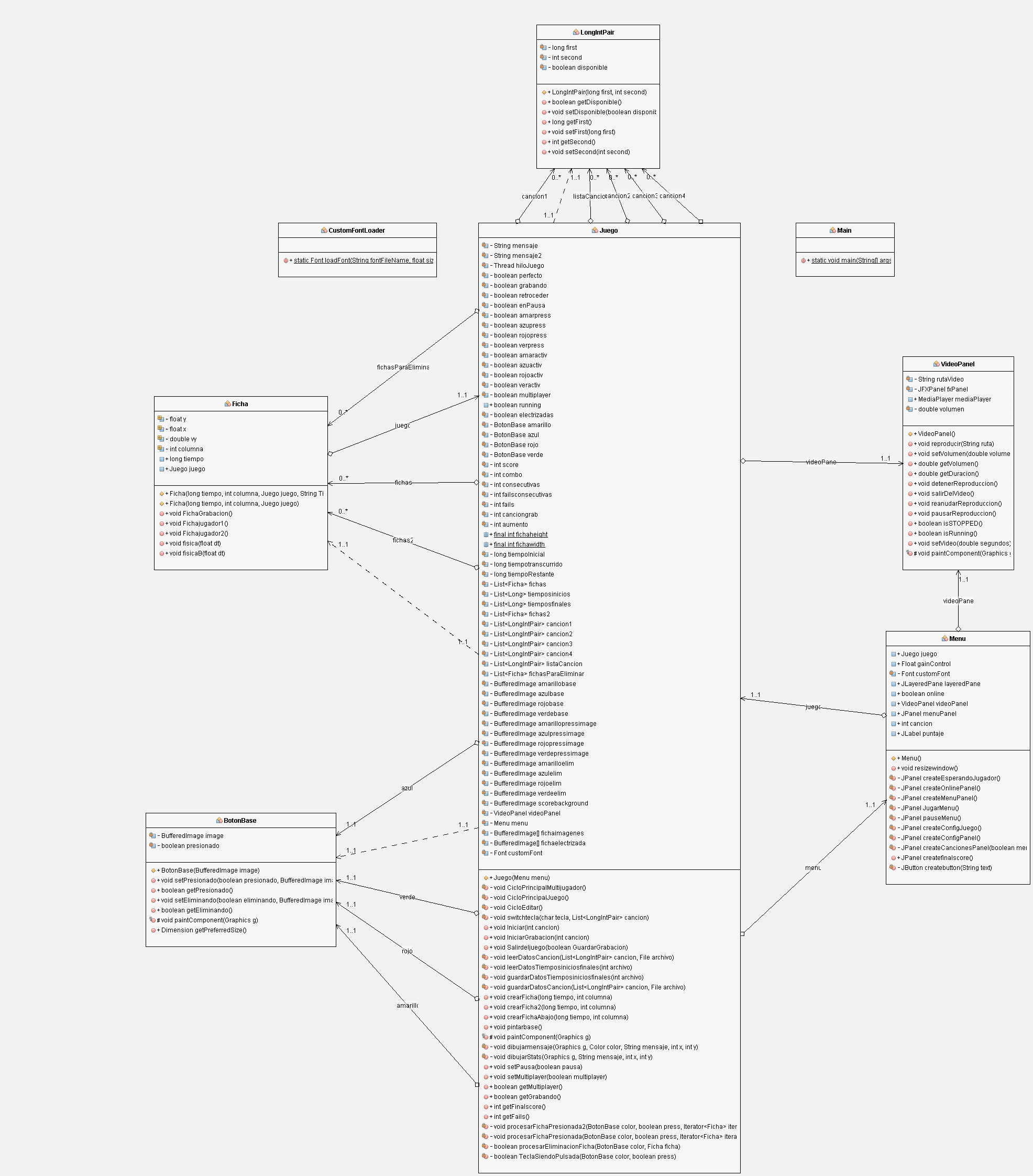
**LongIntPair**: La clase LongIntPair es una clase que almacena un par de valores, uno largo (long) y uno entero (int).

* Almacena un valor largo y un valor entero.
* Proporciona métodos para obtener y establecer estos valores.
* Implementa la interfaz Comparable para permitir la comparación entre instancias de LongIntPair basada en el valor largo.

**VideoPanel**: La clase VideoPanel es una clase personalizada que se utiliza para mostrar videos en tu aplicación a travez de JavaFX.

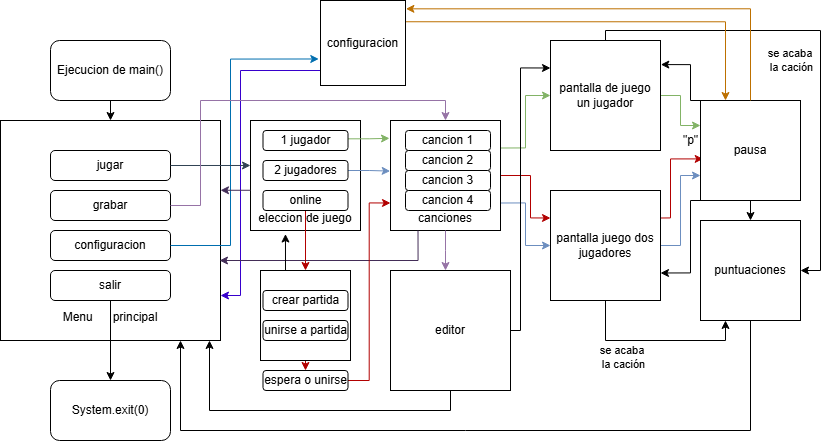
* Carga y reproduce un video.
* Puede pausar, reanudar y detener la reproducción del video.
* Maneja el redimensionamiento del video para que se ajuste al tamaño del panel.

### Jerarquia de clases UML



### Secuencia lógica de pantallas

–



# Conclusion

El proyecto que desarrollamos es un juego de ritmo al pleno estilo de Guitar Hero, donde los jugadores siguen el ritmo de varias canciones presionando teclas en el momento correcto. A lo largo del desarrollo del proyecto, abordamos múltiples aspectos técnicos y de diseño que resultaron en un juego similar al original. Establecimos un repositorio en GitHub al inicio, lo que facilitó la colaboración entre los integrantes del equipo, permitiendo un trabajo en equipo eficiente y el seguimiento detallado del progreso. La creación de una documentación detallada y un cronograma claro fueron esenciales para mantenernos enfocados y asegurar que cada etapa del proyecto se completara a tiempo. En cuanto a la interfaz grafica, diseñamos un sistema de menús intuitivo que incluye opciones de configuración, selección de modos de juego y pausas, permitiendo una navegación facil entre las secciones del juego. El juego soporta tanto el modo de un jugador como el modo de dos jugadores, así como un modo online el cual no pudimos pulir por completo debido a la falta de tiempo. También implementamos un sistema de puntajes donde hay combos que recompenzan al jugador, donde al se muestran al finalizar cada canción. Integramos cuatro canciones de rock, cada una con su propio escenario temático donde tambien se pueden editar atravez de un editor el cual permite añadir y quitar fichas con teclas específicas y ajustar el tiempo de las fichas , ofreciendo la capacidad de aumentar la dificultad al gusto del jugador. Definimos varias clases clave para manejar distintos aspectos del juego, como la lógica del juego, la interfaz de usuario y la reproducción de videos, destacando la clase Ficha indispensables para el juego y VideoPanel para la visualización de videos a travez de la libreria JavaFX22. La planificación detallada y la capacidad de adaptación ante desafíos imprevistos fueron cruciales para este proyecto, el cual no solo nos permitió aplicar conocimientos de programación y diseño de interfaces, sino que también nos enseñó la importancia de la colaboración, la planificación y el manejo del tiempo. El resultado es un juego entretenido y dinámico, sirviendo como una sólida base para futuros proyectos.

# Referencias

* Oracle. (2024). JFrame (Java Platform SE 8). Recuperado de <https://docs.oracle.com/javase/8/docs/api/javax/swing/JFrame.html>
* Oracle. (2024). AudioInputStream (Java Platform SE 8). Recuperado de <https://docs.oracle.com/javase/8/docs/api/javax/sound/sampled/AudioInputStream.html>
* Fxdocs. (s.f). JavaFX Documentation Project. Recuperado de <https://fxdocs.github.io/docs/html5/>
* OpenJDK. (2022). JavaFX 22.0.1 Release .Recuperado de <https://jdk.java.net/javafx22/>
* Vercetti95. (2022). System of a Down - Toxicity [Guitar Hero: World Tour Definitive Edition] (Perfomance Mode) (Performance Mode) [Video]. YouTube. Recuperado de <https://www.youtube.com/watch?v=xsCmE_U_-Os>
* Vercetti95. (2023). R.E.M. - Losing My Religion [Guitar Hero World Tour: Definitive Edition] (Performance Mode) [Video]. YouTube. Recuperado de <https://www.youtube.com/watch?v=IfQ4EZCHsWY>
* Video Gabe. (2023). Kung Fu Fighting - Carl Douglas [Guitar Hero World Tour Definitive Edition] [Video]. YouTube. Recuperado de <https://www.youtube.com/watch?v=c9SFxMeMZSQ>
* KawiiThug. (2022). Franz Ferdinand - Take Me Out [Guitar Hero World Tour Definitive Edition] [Video]. YouTube. Recuperado de <https://www.youtube.com/watch?v=TQTyKMZ0fFY>
* Oracle. (2024). Float Control. Recuperado de
* <https://docs.oracle.com/javase%2F7%2Fdocs%2Fapi%2F%2F/javax/sound/sampled/FloatControl.html>
* Visser, U. (s.f.). Introduction to Game Programming. Miami University. Recuperado de <https://www.cs.miami.edu/home/visser/csc329-files/Games-Threads.pdf>
* ThenisH (2018). Programación de sockets Java: programa cliente-servidor simple[Video]. Recuperado de <https://www.youtube.com/watch?v=-xKgxqG411c>
* Vargas, P. (2012). JFrame en pantalla completa en Java. Recuperado de <https://stackoverflow.com/questions/11570356/jframe-in-full-screen-java>