



**UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS  
FACULTAD 4**

# **Desarrollo del componente “Clave del Son” para el videojuego “Rescatando al Son”.**

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

**Autor:**

Kevin Padrón Domínguez

**Tutores:**

Ing. Jorge Evelio Valdivia Hernández

M.Sc. PA Yanetsi Millet Lombida

La Habana, diciembre de 2023

Año 65 de la Revolución

## DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título **Desarrollo del componente “Clave del Son” para el videojuego “Rescatando al Son”** concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los días 8 del mes de diciembre del año 2023.

**Kevin Padrón Domínguez**



---

Firma del Autor

**Ing. Jorge Evelio Valdivia Hernández**



---

Firma del Autor

**M.Sc. PA Yanetsi Millet Lombida**



---

Firma del Tutor

## **DATOS DE CONTACTO**

Jorge Evelio Valdivia Hernández, graduado de Ingeniería en Ciencias Informáticas en 2019. Profesor durante 2 años de asignaturas pertenecientes al Dpto. de Matemáticas y actual Especialista "A" en Ciencias Informáticas. Se desempeña como programador en los proyectos desarrollados por el Centro de Tecnologías Interactivas.

Correo electrónico: [jevaldivia@uci.cu](mailto:jevaldivia@uci.cu)

Yanetsi Millet Lombida, graduada de Ingeniería en Ciencias Informáticas en el año 2007. Máster en Calidad de Software y profesora auxiliar con más de 15 de años de experiencia en la asignatura de Proyecto de Investigación y Desarrollo. Vicedecana de la Facultad 4 de la Universidad de las Ciencias Informáticas. Líneas de Investigación: Ingeniería y Calidad de Software.

Correo electrónico: [ymillet@uci.cu](mailto:ymillet@uci.cu)

## **AGRADECIMIENTOS**

A mis padres por darme su apoyo en todo momento y dedicarme tanta preocupación, por ser mis guías y mi razón de ser.

A mis abuelos y tíos que siempre han estado ahí luchando conmigo a mi lado para culminar mis estudios.

A mi hermano leo que lo quiero de corazón.

En general a toda mi familia ya que en lo poco y en lo mucho siempre he tenido su apoyo.

A todas mis amistades y en especial a las de mi carrera ya que sin ellos esta meta no hubiera sido posible, gracias por darme esos momentos y sentir lo que es verdaderamente una universidad (la mejor etapa de la vida).

A mis tutores por la ayuda inquebrantable en cada aspecto de esta tesis.

## **DEDICATORIA**

A mi abuelo Padrón, por su amor incondicional.

Espero que este orgulloso desde el cielo.

## **RESUMEN**

El desarrollo de videojuegos ha experimentado un crecimiento significativo, ofreciendo no solo entretenimiento sino también oportunidades lucrativas y formas efectivas de comunicación. Los videojuegos se perfilan como una alternativa atractiva para reflejar valores y principios en la sociedad cubana. En el ámbito de la Universidad de las Ciencias Informáticas, el enfoque en el desarrollo de juegos de ritmos como Clave del Son se alinea con esta tendencia. Este juego se centra en la habilidad y la coordinación al tocar las teclas de un piano en un patrón específico, ofreciendo una experiencia interactiva única. El desarrollo de Clave del Son, guiado por un marco ingenieril para juegos, se basa en el uso de C# como lenguaje de programación y el motor de juegos Unity. Este enfoque estratégico se alinea con la tendencia actual de integrar servicios en línea para enriquecer la experiencia del usuario en juegos, promoviendo la interacción y la competitividad entre los jugadores.

### **PALABRAS CLAVE**

videojuegos, Son, juegos de ritmos

## **ABSTRACT**

*Video game development has seen significant growth, offering not only entertainment but also lucrative opportunities and effective forms of communication. Video games are emerging as an attractive alternative to reflect values and principles in Cuban society. In the field of the University of Computer Sciences, the focus on the development of games such as Clave del Son aligns with this trend. This game focuses on skill and coordination when playing the keys of a piano in a specific pattern, offering a unique interactive experience. The development of Clave del Son, guided by an engineering framework for games, is based on the use of C# as a programming language and the Unity game engine. This strategic approach aligns with the current trend of integrating online services to enrich the user experience in games, promoting interaction and competitiveness among players.*

### **KEYWORDS**

*video games, Son, Rhythm Games*

**ÍNDICE**

INTRODUCCIÓN.....	1
CAPÍTULO I: Fundamentación Teórica .....	6
1.1 Videojuego.....	6
1.2 Videojuego de ritmo .....	8
1.3 Rescatando el Son.....	9
1.4 Sistemas homólogos.....	10
1.5 Metodología de desarrollo de software .....	17
1.6 Herramientas y tecnologías .....	18
Conclusiones del capítulo .....	21
CAPÍTULO II: Conceptualización y diseño de la solución .....	22
2.1 Propuesta de solución .....	22
2.2 Conceptualización de la propuesta de solución.....	25
2.3 Diseño del Videojuego .....	26
2.4 Especificación de mecanismos.....	31
2.5 Requisitos no funcionales .....	36
2.6 Integración con el juego macro.....	37
2.7 Paquetes de mecanismos.....	37
2.8 Estilo arquitectónico.....	39
2.9 Diagrama de clases .....	41
2.10 Patrones de diseño .....	42
2.11 Representación de comportamiento .....	44
Conclusiones del capítulo .....	45
CAPÍTULO III: Implementación y pruebas de la solución .....	46
3.1 Estándar de codificación .....	46
3.2 Assets y recursos utilizados.....	47
3.3 Diagramas de componentes .....	48
3.4 Pruebas .....	49
3.5.1 Pruebas Alpha .....	49

3.5.2 Pruebas Beta .....	51
Conclusiones del capítulo .....	52
CONCLUSIONES .....	54
RECOMENDACIONES .....	55
REFERENCIAS BIBLIOGRÁFICAS .....	56
ANEXOS .....	59
A.1 Diagrama de clases .....	59
A 1.1 Diagrama de clases del componente Clave del Son. ....	59
A 1.2 Diagrama de clases del mecanismo M1 Control del Menú Principal. ....	59
A 1.3 Diagrama de clases del mecanismo M2 Control del Menú Partida. ....	60
A 1.4 Diagrama de clases del mecanismo M3 Generación de notas. ....	60
A 1.5 Diagrama de clases del mecanismo M4 Nota. ....	61
A 1.6 Diagrama de clases del mecanismo M5 Puntuación. ....	61
A 1.7 Diagrama de clases del mecanismo M6 Efecto de sonido. ....	62
A 1.8 Diagrama de clases del mecanismo M7 Interacción con las notas. ....	62
A.2 Diagrama de estado .....	63
A 2.1 Diagrama de estado del menú principal. ....	63
A 2.2 Diagrama de estado de la partida.....	63
A.3 Diagramas de componentes.....	63
A 3.1 Diagrama de componentes de mecanismo M1 Control del menú principal. ....	63
A 3.2 Diagrama de componentes de mecanismo M2 Control de partida. ....	64
A 3.3 Diagrama de componentes de mecanismo M3 Generación de notas. ....	64
A 3.4 Diagrama de componentes de mecanismo M4 Nota. ....	64
A 3.5 Diagrama de componentes de mecanismo M5 Puntuación. ....	65
A 3.6 Diagrama de componentes de mecanismo M6 Efecto de sonido.....	65
A 3.7 Diagrama de componentes de mecanismo M7 Interacción con las notas. ....	65
A 4 No conformidades registradas durante las pruebas Beta .....	66

**ÍNDICE DE TABLAS**

Tabla 1. Características de los sistemas homólogos..... 16  
Tabla 2. Jugador.....26  
Tabla 3. Objetivos.....27  
Tabla 4. Elementos de pantallas gráficas elementales.....29  
Tabla 5. Especificación de mecanismos.....31

## ÍNDICE DE FIGURAS

Figura 1. Videojuego Magic Tiles 3.....	11
Figura 2. Videojuego Tiles Hop.....	13
Figura 3. Videojuego Tap Tap Reborn 2.....	14
Figura 4. Videojuego Cytus II.....	16
Figura 5. Propuesta de solución. ....	23
Figura 6: Diagrama de paquetes de mecanismos .....	38
Figura 7. Capas de la arquitectura de la propuesta de solución. ....	40
Figura 8. Diagrama de clases del mecanismo M1 Control de menú principal. ....	41
Figura 9. Patrón de diseño Fachada implementado en la solución. ....	44
Figura 10. Diagrama de estado del mecanismo M2 Control de partida. ....	45
Figura 11. Diagrama de componentes M1 Control del menú principal. ....	48
Figura 12 No conformidades.....	52

## INTRODUCCIÓN

El uso de las Tecnologías de Información y las Comunicaciones (TIC) en los últimos años ha permitido extender la informática a muchos sectores, ocupando cada vez más espacio en la vida y funcionamiento de la sociedad. Uno de los procesos que mayor impacto ha tenido en la sociedad es el desarrollo de videojuegos. Hoy en día, la industria del videojuego se ha consolidado como la principal contribuidora en la economía del entretenimiento y ocio, llegando a ser más relevante que industrias del mismo tipo como la música y el cine. Los videojuegos son una vía de entretenimiento interactivo en el que uno o varios usuarios, mediante un dispositivo de entrada, se comunican con un sistema que posea imágenes de video. La plataforma en la que se desarrolle (o sistema) puede ser una computadora, consola, o incluso un celular. La interactividad usuario–sistema está dada por la capacidad del equipo de trabajo de planificar un sistema en el que el usuario se sienta cómodo y controle la situación (Norrie, 2022).

A lo largo de la historia de los videojuegos, sus creadores han dado lugar a una variedad creciente de géneros en las distintas plataformas disponibles. Estos géneros se han conformado en torno a factores como: la representación gráfica, el tipo de interacción entre el jugador y la máquina, la ambientación, y su sistema de juego, siendo este último el criterio más habitual a tener en cuenta. Los géneros o categorías de videojuegos son una forma de clasificar los videojuegos en función fundamentalmente de su mecánica de juego. Sin embargo, existen otros factores como la estética o la temática que pueden influir también a la hora de definir ciertos géneros (Stephen, 2023). Es difícil que las personas no encuentren un videojuego capaz de satisfacer sus preferencias. Aquellos que son amantes de la música también encuentran su espacio dentro del mercado en los juegos conocidos como rítmicos. Los videojuegos rítmicos les proporcionan una experiencia centrada en la música y el ritmo.

El son cubano es un género musical tradicional de Cuba que surgió a finales del siglo XIX como una fusión de ritmos afro caribeños y europeos. Con sus características percusiones de clave y bongó, melodías pegadizas y letras que hablan de la vida cotidiana y el amor, el son se convirtió rápidamente en la música nacional de Cuba. Comenzando en las regiones rurales

de Oriente, donde ritmos como el changüí se mezclaron con la trova, el son cubano adoptó los típicos patrones de llamada y respuesta entre las voces del solista improvisando y el coro. Músicos legendarios como Miguel Matamoros y los grupos Trío Matamoros y Sexteto Habanero llevaron estas primeras formas de son a La Habana, fusionándolas con otros géneros populares de la época. Para mediados del siglo XX, el son cubano había adquirido sus elementos definitorios como el montuno, los sones en clave y una prominente sección de metales. Por lo tanto, en septiembre de 2012 fue declarado patrimonio cultural de Cuba (Corazón Tierra, 2019).

Con el surgimiento y popularización de nuevos géneros musicales como el reggaetón, hip hop, rock y pop han capturado la atención de los jóvenes cubanos. Muchos de ellos prefieren estas músicas más contemporáneas y comerciales sobre el son tradicional, por lo cual el interés de las nuevas generaciones por el son ha disminuido drásticamente lo que ha causado una pérdida considerable de la presencia en la cultura popular de este género (Kuss, 1994). Por esta razón el ministerio de Educación Superior unido al Ministerio de Cultura se ha dado a la tarea de crear un proyecto extensionista. El proyecto utiliza la clasificación de juegos serios para brindar a las generaciones más jóvenes los conocimientos relacionados a esta arista de la cultura nacional. Permite que de manera interactiva y entretenida los jugadores puedan familiarizarse con los detalles y las formas que dan vida a este importante fragmento de la cultura cubana.

Como alternativa interactiva se desarrolla en colaboración con el Centro de Tecnologías Interactivas (VERTEX) de la Universidad de las Ciencias Informáticas (UCI) y el Instituto Superior de Diseño (ISDi), este último encargado de toda la parte de recursos visuales, un videojuego llamado Rescatando el Son. Tiene como objetivo llevar a los jugadores por un viaje cultural en el que pueden conocer, aprender y descubrir todo lo relacionado con el género mientras que se entretienen y divierten. Para esto se utilizan un conjunto de minijuegos de agilidad mental que cambian su estética y jugabilidad para reflejar los elementos principales del son en cada provincia. Sin embargo, Isla del Son carece de un medio para brindarle al jugador una forma de obtener información y comprobar sus conocimientos, tampoco logra

familiarizarlo con los acordes y notas musicales que caracterizan al ya mencionado género. Esto trae consigo que el jugador no logre obtener nuevos conocimientos acerca del son y no se familiarice con sus sonidos más representativos lo que dificulta el objetivo principal del juego de volver a crear interés por dicho género musical.

Teniendo en cuenta la situación problemática antes descrita, se identifica como **problema de investigación**: ¿cómo contribuir al videojuego Rescatando el Son mecanismos para obtener conocimientos e identificar los ritmos musicales de los subgéneros del son?

Como **objeto de estudio** se define: los videojuegos de ritmos, mientras que el **campo de acción** es los videojuegos de estilo piano tile.

Para dar solución al problema de investigación planteado, se define **como objetivo general**: Desarrollar un componente en forma de minijuego para el videojuego Rescatando el Son perteneciente al género de ritmo, para el mejoramiento de la coordinación, el oído musical y el sentido del tempo de los jugadores.

Para darle cumplimiento al objetivo planteado se proponen las siguientes **tareas de investigación**:

1. Elaboración del marco teórico de la investigación a partir del estado del arte de los videojuegos de ritmos.
2. Caracterización del desarrollo de videojuegos pertenecientes al género de ritmo.
3. Análisis de las herramientas y metodologías para el desarrollo de la propuesta de solución.
4. Implementación del minijuego como componente propuesto.
5. Validación a través de pruebas al minijuego como componente implementado.

El desarrollo de la investigación se apoya en algunos métodos científicos que son muy útiles para guiarla. A continuación, se explican los métodos de investigación que se utilizan, los cuales se pueden clasificar en métodos teóricos y empíricos.

### **Métodos teóricos:**

- **Histórico-lógico:** se utiliza para Develar, identificar o descubrir las tendencias de la evolución a través del tiempo de los videojuegos.
- **Analítico-sintético:** se emplea para analizar las interrelaciones entre sujetos, objetos, medios y condiciones externas para sintetizar las esencias presentes en los diferentes conceptos y definiciones relacionados con la investigación.
- **Sistémico** se utiliza para determinar los componentes que forman parte del proceso de desarrollo de videojuegos y analizar la interacción que existen entre estos.

### **Métodos empíricos:**

- **Observación:** se emplea para estudiar distintos videojuegos. Estos sirvieron como objeto de análisis y comparación para establecer las características y elementos fundamentales que deben estar presente en un prototipo de videojuego del mismo tipo y género.

El presente trabajo se compone de tres capítulos, estructurados de la siguiente forma:

- **Capítulo 1 Fundamentación Teórica:** En este capítulo se definen los elementos teóricos necesarios para el desarrollo de la investigación y los principales conceptos que se emplean durante todo el trabajo. Se realiza un análisis de las soluciones similares y se selecciona la metodología de desarrollo de software, las herramientas y tecnologías a utilizar.
- **Capítulo 2 Conceptualización y diseño de la solución:** en este capítulo se describe la propuesta de solución y los artefactos generados de acuerdo a la metodología seleccionada para el desarrollo de videojuegos, estos son: documento de diseño del videojuego, especificación de mecanismos, descripción de requisitos no funcionales y arquitectura.
- **Capítulo 3 Implementación y pruebas de la solución:** en este capítulo se realiza una representación de cada uno de los elementos que conformaran la solución, así como los estándares de codificación utilizados en su implementación. Finalmente se presenta el resultado de las pruebas realizadas a la aplicación para validar su correcto funcionamiento.

## CAPÍTULO I: Fundamentación Teórica

En este capítulo, se establecen los fundamentos conceptuales que respaldan el ámbito de investigación, abordando tanto los videojuegos como los juegos de ritmos. Se efectúa un análisis del juego principal, Rescatando el Son, el cual constituye el punto focal de la propuesta. Se subraya la importancia del género de ritmo en este contexto. Asimismo, se examinan ejemplos pertinentes de sistemas homólogos, detallando sus características distintivas. Se proporciona una visión detallada de las tecnologías y herramientas primordiales que podrían emplearse para la materialización de la solución propuesta. Además, se expone la metodología de desarrollo de software que se planea emplear.

### 1.1 Videojuego

Un videojuego es una forma de entretenimiento interactivo que combina elementos visuales, auditivos y mecánicos para permitir a los usuarios participar en una experiencia virtual. Los videojuegos se ejecutan en dispositivos electrónicos, como computadoras, consolas de videojuegos, teléfonos móviles y tabletas. En un videojuego, los jugadores asumen el papel de un personaje o controlan elementos en un entorno digital creado específicamente para el juego. Los videojuegos pueden presentar una variedad de géneros, que van desde la acción y la aventura hasta los rompecabezas, los juegos de rol, los juegos de estrategia y los juegos de deportes, entre otros. Los videojuegos suelen contar con objetivos claros que los jugadores deben alcanzar, ya sea completando misiones, resolviendo acertijos, ganando competencias o explorando mundos virtuales. Para lograr estos objetivos, los jugadores interactúan con el juego a través de controles, como teclados, joysticks, controladores de movimiento o pantallas táctiles, dependiendo del dispositivo utilizado. Además, los videojuegos pueden ofrecer elementos narrativos, donde se cuenta una historia a través del juego, y también pueden incluir aspectos sociales, permitiendo a los jugadores interactuar y competir con otros jugadores en línea (Julián Pérez Porto y Ana Gardey, 2021).

Un juego se define como una actividad recreativa que se lleva a cabo dentro de un contexto definido, con reglas establecidas y un objetivo o meta determinada. Los juegos son realizados

por una o más personas, quienes participan voluntariamente y se divierten en el proceso (Julián Pérez Porto y Ana Gardey, 2021).

### **Clasificación y géneros de los videojuegos**

Los videojuegos se clasifican en diversos géneros según sus características temáticas, mecánicas de juego y estilo visual. A continuación, se presentan algunos de los géneros más comunes (Stephen, 2023):

- **Acción:** Los juegos de acción se centran en la rapidez, los reflejos y el combate. Incluyen subgéneros como los juegos de disparos en primera persona (FPS), juegos de acción en tercera persona, juegos de plataformas y juegos de lucha.
- **Aventura:** Los juegos de aventura se enfocan en la exploración y la resolución de acertijos en un mundo narrativo. Pueden ser de acción-aventura, donde se combina la acción con elementos de exploración y resolución de acertijos, o de aventura gráfica, que se centran más en la narrativa y los rompecabezas.
- **Rol (RPG):** Los juegos de rol permiten a los jugadores asumir el papel de un personaje y tomar decisiones que afectan el desarrollo de la historia. Suelen incluir elementos de progresión de personajes, niveles de experiencia y personalización.
- **Estrategia:** Los juegos de estrategia requieren planificación y toma de decisiones estratégicas para alcanzar la victoria. Pueden ser juegos de estrategia en tiempo real (RTS), donde los jugadores toman decisiones en tiempo real, o juegos de estrategia por turnos, donde los jugadores toman decisiones por turnos.
- **Deportes:** Los juegos de deportes simulan actividades deportivas, como fútbol, baloncesto, tenis o carreras. Los jugadores pueden controlar equipos o atletas individuales y participar en competiciones virtuales.

- **Simulación:** Los juegos de simulación recrean situaciones o actividades de la vida real, como simuladores de vuelo, simuladoras de construcción de ciudades o simuladoras de vida. Estos juegos buscan brindar una experiencia realista en un contexto virtual.
- **Puzzle:** Los juegos de puzzle se basan en desafíos lógicos o rompecabezas que requieren la resolución de problemas para avanzar. Pueden ser juegos de lógica, juegos de combinaciones o juegos de plataformas basados en la resolución de acertijos.
- **Agilidad mental:** Estos son juegos donde el jugador debe pensar y agilizar la mente. El objetivo es resolver ejercicios con dificultad progresiva para desarrollar la habilidad mental. Algunos videojuegos de este género son: Brain Academy, La Neurona.
- **Ritmo:** Estos juegos están diseñados para desafiar y entretener a los jugadores al seguir el ritmo de la música y realizar acciones precisas en el momento adecuado.

Estos son solo algunos de los géneros más comunes, y existen muchos otros que abarcan una amplia variedad de temas y mecánicas de juego. Además, es común que los juegos combinen elementos de varios géneros, creando nuevas categorías o subgéneros en la industria de los videojuegos. Los videojuegos de ritmo son aquellos que desafían a los jugadores a seguir el ritmo de la música y realizar acciones precisas en sincronía con los elementos en pantalla. Pertenecen al género de música o ritmo y pueden presentar subgéneros como juegos de baile, música instrumental, mezcla de música y simulación de instrumentos. En estos últimos se enfocará la investigación con el fin de dar solución al objetivo general planteado.

## 1.2 Videojuego de ritmo

Un videojuego de ritmo es un tipo de juego en el que los jugadores interactúan con la música a través de acciones en pantalla. Estos juegos desafían a los jugadores a seguir el ritmo de la

música y realizar acciones precisas en sincronía con los elementos visuales que aparecen en pantalla. En los videojuegos de música o ritmo, los jugadores suelen enfrentarse a una serie de notas, botones o comandos que se presentan en el momento adecuado de acuerdo con la música que se reproduce. Los jugadores deben presionar los botones o realizar acciones específicas en el momento exacto para obtener la puntuación más alta posible. Estos juegos a menudo ofrecen una variedad de canciones y estilos musicales para que los jugadores elijan, lo que les permite disfrutar de diferentes géneros musicales mientras juegan. Algunos videojuegos de música también permiten a los jugadores desbloquear nuevas canciones, desafíos o modos de juego a medida que avanzan en el juego. Uno de los ejemplos más conocidos de videojuegos de ritmo es la serie "Guitar Hero", en la que los jugadores usan controladores de guitarra de plástico para tocar notas en sincronía con las canciones. Otro ejemplo popular es "Dance Revolution", en el que los jugadores deben seguir los pasos de baile que aparecen en una plataforma de baile física. Estos juegos de ritmo no solo ofrecen diversión y entretenimiento, sino que también pueden ayudar a desarrollar habilidades de coordinación mano-ojo, memoria y ritmo. Además, algunos videojuegos de música también incluyen modos multijugador, lo que permite a los jugadores competir entre sí o colaborar en desafíos musicales (Martínez Cruz, 2023).

### 1.3 Rescatando el Son

En el documento de diseño del videojuego Rescatando el Son entregado por el cliente se define como: videojuego que agrupa un compendio de minijuegos para celular. Estos minijuegos están aglutinados alrededor del son cubano, su historia, sus principales intérpretes y sus tradiciones en general. Cada provincia tiene un escenario embellecido con los elementos tradicionales e instrumentos típicos de la región. Este escenario es la "Casa Llamo al Son" y la ambientación se devela según se vayan ganando puntos ("claves de son"). Las claves de son se obtienen completando satisfactoriamente los niveles de la provincia. Una de las mecánicas más importante de estos minijuegos es el cambio en sus escenarios y sus mecánicas dependiendo la provincia que seleccione el jugador en el juego base, teniendo que reflejar en estos cambios características únicas de dicha provincia. Para esto, los minijuegos tienen acceso a la información de la provincia del juego base y también envían al juego base los datos

referentes a la puntuación obtenida por el jugador. Los juegos son variados y de corta duración que van escalando en dificultad en la medida en que progresa el viaje; estos son: runner “Son de máquina”, juego de memoria “afinación de instrumentos”, juego de ritmo “clave de son” y una un juego de preguntas y respuestas llamado “Sabios del son”. Estos dos últimos pertenecientes al género de agilidad mental y suelen estar asociados a los juegos serios, los cuales cumplen con las características educativas que se requieren en la propuesta.

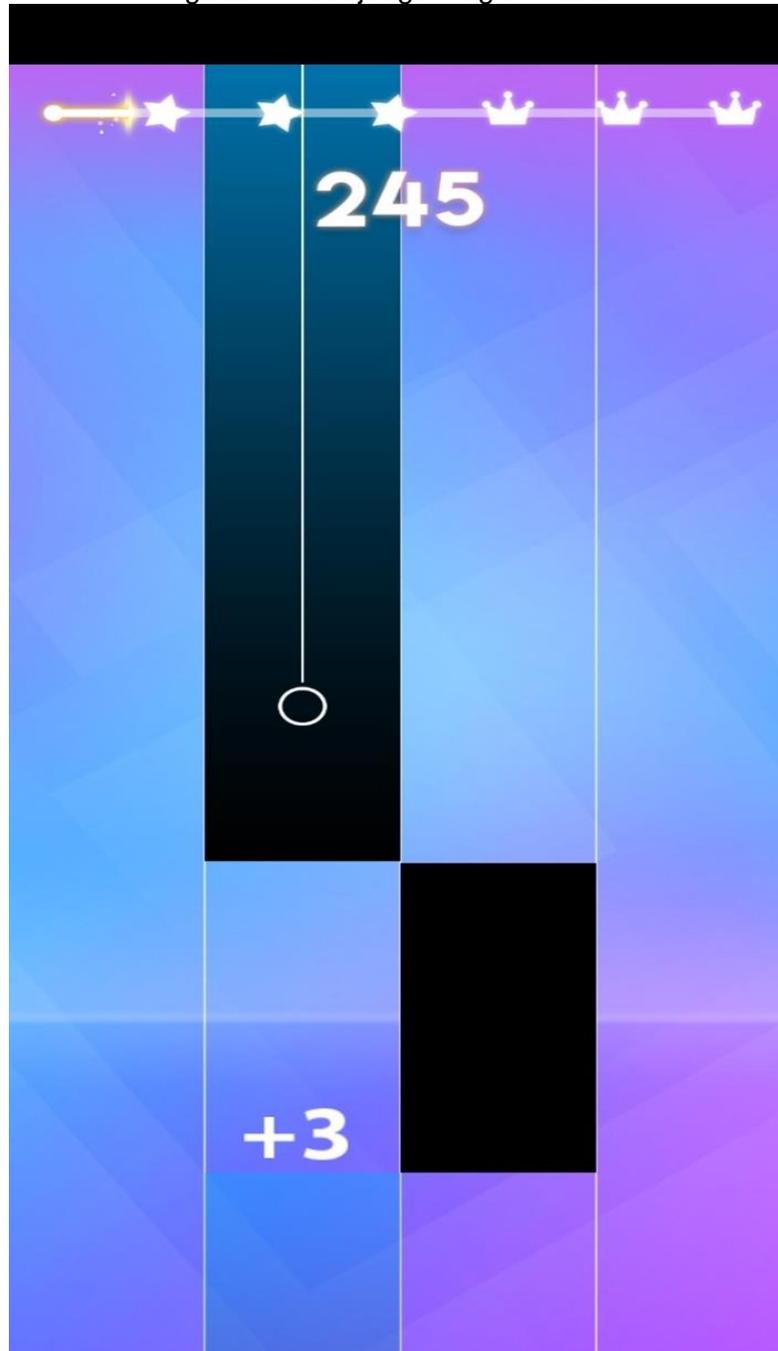
#### 1.4 Sistemas homólogos

Los videojuegos de ritmo constituyen un género fascinante que combina la destreza musical con la acción y la diversión interactiva. A lo largo de los años, varios sistemas homólogos han surgido para llevar la experiencia de juego a nuevas alturas, proporcionando a los jugadores la oportunidad de sumergirse en mundos virtuales donde la música guía la acción. En esta introducción, exploraremos algunos de los sistemas homólogos más destacados en el género de los videojuegos de ritmo.

**Magic Tiles 3:** Magic Tiles 3 es un popular juego de ritmo que se asemeja a Piano Tiles (Andrés López, 2023) en su mecánica de juego. En este juego, los jugadores deben tocar las teclas de piano en el momento adecuado para seguir el ritmo de la música y avanzar en el juego. Magic Tiles 3 ofrece una amplia selección de canciones en diferentes géneros musicales, desde música clásica hasta pop y rock. Los jugadores pueden elegir entre una variedad de modos de juego, incluyendo el modo clásico, el modo arcade, el modo zen y el modo batalla. Cada modo de juego ofrece una experiencia única y desafiante. El objetivo principal en Magic Tiles 3 es tocar las teclas de piano en el momento exacto en el que las notas llegan a la línea de tiempo en la parte superior de la pantalla. Los jugadores deben ser precisos y tener buen ritmo para obtener una puntuación alta. Si tocan una tecla incorrecta o se retrasan, perderán el juego y deberán volver a intentarlo. A medida que los jugadores progresan en el juego, desbloquean nuevas canciones y niveles más difíciles. Magic Tiles 3 también cuenta con un sistema de clasificación en línea que permite a los jugadores comparar sus puntuaciones con otros jugadores de todo el mundo. El juego ofrece una experiencia visual atractiva con colores vibrantes y efectos visuales cautivadores. Además, Magic Tiles 3 cuenta

con una interfaz sencilla y fácil de usar, lo que lo hace accesible para jugadores de todas las edades y niveles de habilidad (Andrés López, 2023).

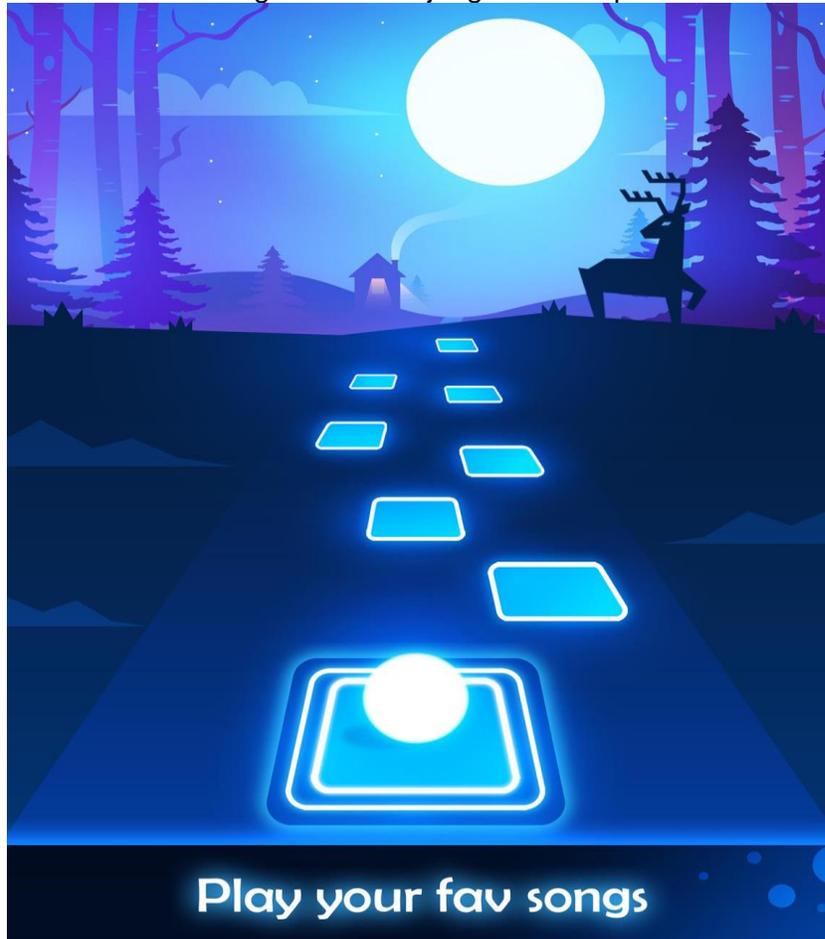
Figura 1. Videojuego Magic Tiles 3.



Fuente. Andrés López, U. T. (2023). Magic Tiles 3 (Android). Uptodown. <https://magic-tiles-3.uptodown.com/android>.

**Tiles Hop:** Tiles Hop es otro popular juego de ritmo en el que los jugadores controlan una bola que rebota sobre las teclas de piano en un camino en constante movimiento. El objetivo del juego es tocar la pantalla en el momento preciso para saltar de una tecla a otra y seguir el ritmo de la música. En Tiles Hop, los jugadores deben mantener el ritmo y la precisión mientras navegan por el camino de teclas de piano en constante cambio. Cada tecla tiene una nota musical asociada, y los jugadores deben tocar la pantalla en el momento adecuado para que la bola salte a la siguiente tecla en la secuencia correcta. El juego presenta una amplia variedad de canciones de diferentes géneros musicales, desde música pop y electrónica hasta hip-hop y música clásica. Los jugadores pueden desbloquear nuevas canciones a medida que avanzan en el juego o incluso importar sus propias canciones para una experiencia personalizada. Tiles Hop también cuenta con diferentes modos de juego, como el modo de desafío, en el que los jugadores deben intentar superar niveles cada vez más difíciles, y el modo infinito, donde pueden disfrutar de una experiencia relajada sin restricciones de tiempo. El juego ofrece una experiencia visual atractiva con gráficos coloridos y efectos visuales llamativos. Además, los jugadores pueden personalizar la apariencia de la bola y el ambiente visual del juego mediante la elección de diferentes temas. Tiles Hop es un juego adictivo y desafiante que combina la diversión del juego de ritmo con la emoción de saltar y seguir el ritmo de la música. Es adecuado para jugadores de todas las edades y ofrece una experiencia entretenida y estimulante para los amantes de la música y el ritmo (Uptodown Content Team, 2023).

Figura 2. Videojuego Tiles Hop.



Fuente. Uptodown Content Team, U. T. (2023). Tiles Hop (Android). Uptodown. <https://tiles-hop.uptodown.com/android>.

**Tap Tap Reborn 2:** Tap Tap Reborn 2 es un juego de ritmo en el que los jugadores deben tocar las notas en el momento exacto para seguir el ritmo de la música. Es la secuela del popular juego Tap Tap Reborn y ofrece una experiencia de juego emocionante y desafiante. En Tap Tap Reborn 2, los jugadores se enfrentan a una serie de notas que aparecen en la pantalla en sincronía con la música. Las notas pueden ser tocadas tocando, deslizando o manteniendo presionadas diferentes áreas de la pantalla, dependiendo del patrón de notas que se presente. Los jugadores deben ser precisos y seguir el ritmo de la música para obtener la puntuación más alta posible. El juego presenta una amplia variedad de canciones en

diferentes géneros musicales, que van desde música electrónica y pop hasta rock e hip-hop. Los jugadores pueden desbloquear nuevas canciones a medida que avanzan en el juego o incluso importar sus propias canciones para una experiencia personalizada. Tap Tap Reborn 2 ofrece diferentes modos de juego para mantener la diversidad y el desafío. Puede incluir modos como el modo clásico, el modo arcade o el modo desafío, cada uno con su propia mecánica de juego y dificultad. Además, el juego también puede tener tablas de clasificación en línea donde los jugadores pueden competir con otros jugadores de todo el mundo. El juego presenta una interfaz visual atractiva con efectos visuales dinámicos y vibrantes que responden al ritmo de la música. También puede ofrecer una personalización visual, como temas o skins que permiten a los jugadores modificar la apariencia del juego según sus preferencias (Uptodown Content Team, 2023).

Figura 3. Videojuego Tap Tap Reborn 2.



Fuente. Uptodown Content Team, U. T. (2023). Tap Tap Reborn 2 (Android). Uptodown. <https://tap-tap-reborn-2.uptodown.com/android>.

**Cytus II:** es un popular juego de ritmo que combina una narrativa profunda con mecánicas de juego emocionantes. Es la secuela del exitoso juego Cytus y ha ganado reconocimiento por su estilo visual único y su enfoque en la música electrónica. En Cytus II, los jugadores se sumergen en un mundo futurista en el que deben seguir el ritmo de la música y tocar las notas en el momento preciso. Las notas aparecen en la pantalla en diferentes patrones y los jugadores deben tocar, deslizar o mantener presionadas las notas según las indicaciones para obtener una puntuación alta. El juego cuenta con una amplia selección de canciones originales de diversos géneros, principalmente música electrónica. Los jugadores pueden desbloquear nuevas canciones a medida que progresan en el juego, lo que les permite experimentar con diferentes estilos musicales y desafíos. Una característica distintiva de Cytus II es su narrativa inmersiva. A medida que los jugadores avanzan en el juego, se les presenta una historia intrigante que se desarrolla a través de las interacciones con los personajes y el descubrimiento de pistas y eventos en el mundo del juego. El juego ofrece varios modos de juego, como el modo historia, en el que los jugadores desbloquean nuevos capítulos y exploran la trama, y el modo libre, en el que pueden jugar canciones específicas sin restricciones. Cytus II se destaca por su estilo visual único, con gráficos impresionantes y diseños de personajes detallados. El juego ofrece una experiencia audiovisual cautivadora, con efectos visuales sincronizados con la música y una interfaz de usuario intuitiva. Además, Cytus II ha sido elogiado por su enfoque en la dificultad progresiva, lo que permite a los jugadores desafiar constantemente sus habilidades de ritmo y coordinación a medida que se enfrentan a canciones más complicadas (Rayark Games, 2023).

Figura 4. Videojuego Cytus II.



Fuente. Rayark Games. (2023, septiembre 18). Cytus II | Cytus Wiki | Fandom. [https://cytus.fandom.com/es/wiki/Cytus\\_II](https://cytus.fandom.com/es/wiki/Cytus_II).

A continuación, se muestra un resumen de las características de interés para la propuesta presentes en los sistemas homólogos estudiados:

Tabla 1. Características de los sistemas homólogos.

Videojuego	Pulsación del jugador en notas	Varios tipos de géneros de canciones	Varios modos de juego	Variación de dificultad	Experiencia visual atractiva	Generación procedural de contenido	Servicios en línea
Magic Tiles 3	si	si	si	si	si	Si	si
Tiles Hop	no	si	si	si	si	No	no
Tap Tap Reborn 2	si	si	si	si	si	Si	si

Cytus II	si	si	si	si	si	Si	no
----------	----	----	----	----	----	----	----

*Fuente. Elaboración propia.*

En base al análisis realizado, los videojuegos de ritmo y pulsación de teclas como Magic Tiles 3, Tiles Hop, Tap Tap Reborn 2 y Cytus II incluyen características importantes que podrían incorporarse en un videojuego de son cubano para lograr una experiencia entretenida y educativa. Elementos clave como la interacción directa del jugador pulsando sobre las notas, una amplia variedad de canciones y géneros, múltiples modos de juego, progresión en dificultad, atractivos gráficos y sonidos, y funcionalidades en línea para compartir puntajes y conectar con otros jugadores; son comunes en los títulos exitosos de este género.

### 1.5 Metodología de desarrollo de software

Para el desarrollo de los videojuegos, Vertex utiliza el Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos. El mismo se centra en las mecánicas que definen el comportamiento en un videojuego, lo que posibilita una adecuada comprensión por parte del equipo de análisis y desarrollo del atributo de calidad y jugabilidad. De igual forma se acoge a un modelo de desarrollo incremental para garantizar la retroalimentación y entrega gradual del videojuego. Se compone por cinco etapas que complementan el proceso de desarrollo de los videojuegos: Conceptualización, Diseño, Implementación, Prueba y Mantenimiento (Hernández P., Andy; Pérez T. Karina; Correa M., Omar, 2017).

A continuación, se define la estructura de la metodología:

- Etapa 1: Conceptualización
  - o Definir el género sobre el cual se desarrollará el videojuego.
  - o Describir la mecánica del videojuego.
  - o Especificar las metas para la experiencia del jugador.
- Etapa 2: Diseño
  - o Describir los elementos formales que definen la estructura del videojuego.
  - o Describir los elementos dramáticos que definen el entretenimiento del videojuego.

- o Diseñar las pantallas gráficas elementales que forman la estructura del videojuego.
- o Describir los elementos dinámicos que definen las mecánicas o mecanismos del videojuego.
- o Validar los mecanismos teniendo en cuenta criterios técnicos para su implementación.
- o Modelar el diagrama de paquetes de mecanismos teniendo en cuenta la distribución arquitectónica.
- o Describir la concepción de los mecanismos sobre la distribución arquitectónica diseñada.
- o Modelar el comportamiento de los mecanismos mediante diagramas de transición de estado.
- o Mantener una trazabilidad bidireccional entre cada elemento del videojuego.
- o Describir las características no funcionales del videojuego.
- Etapa 3: Implementación
  - o Diseñar los componentes que encapsulan la implementación.
  - o Desarrollar las mecánicas especificadas y diseñadas.
- Etapa 4: Prueba
  - o Desarrollar pruebas Alpha.
  - o Desarrollar pruebas Beta.
  - o Registrar defectos durante las pruebas realizadas.
- Etapa 5: Mantenimiento
  - o Realizar análisis Postmortem.
  - o Retomar la etapa de Diseño.

## I.6 Herramientas y tecnologías

Para el desarrollo de la solución son necesarios el empleo de un conjunto de herramientas y tecnologías que permitan guiar este proceso, así como permitir y facilitar su diseño e implementación. A continuación, se detallan las alternativas seleccionadas teniendo en cuenta

los videojuegos realizados en el centro VERTEX y que se usan actualmente en el proyecto Rescatando el Son.

### **Lenguaje de modelado UML**

Lenguaje Unificado de Modelado (UML) es un lenguaje de propósito general que ayuda a especificar, visualizar y documentar modelos de sistemas software, incluido su estructura y diseño, de tal forma que se unifiquen todos sus requerimientos. El objetivo principal del UML es estandarizar el modelado de sistemas software. UML proporciona el vocabulario y reglas necesarias para combinar y construir representaciones, modelos conceptuales y físicos del sistema y permite representar varios modelos, compartiendo características específicas de cada uno como los nombres de clases, atributos y métodos (lucidchart, 2023).

### **Herramienta de modelado**

Para el modelado de la solución se utilizará Visual Paradigm en su versión 8.0, ya que es una de las herramientas consideradas como completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Fue creada para el ciclo vital completo del desarrollo de software que lo automatiza y acelera, permitiendo la captura de requisitos, análisis, diseño e implementación. La herramienta cuenta con funciones que permiten generar código a partir de la representación de los modelos y viceversa (en el documento Desarrollo de un Generador de Datos para el proyecto SIGEP., 2022).

### **Motor de videojuego Unity**

Para la elaboración de la solución se propone como herramienta, la utilización de Unity (específicamente en su versión 2021.3.4f1) Unity es lo que se conoce como un motor de desarrollo o motor de juegos. El término motor de videojuegos (game engine) hace referencia a un software, el cual tiene una serie de rutinas de programación que permiten el diseño, la creación y el funcionamiento de un entorno interactivo, en este caso, de un videojuego (Alfredo y johayana, 2021).

Dentro de las funcionalidades típicas que tiene un motor de videojuegos se encuentran las siguientes:

- Motor gráfico para dibujar gráficos 2D y 3D.
- Motor físico que simule las leyes de la física.
- Animaciones.
- Iluminación.
- Sonidos.
- Inteligencia Artificial.
- Programación o scripting.

### **Lenguaje de programación**

Como lenguaje de programación principal se emplea C# en su última versión estable, ya que es uno de los tres soportados por la plataforma Unity 3D. Además, es orientado a objetos y posee un gran grupo de funcionalidades, así como librerías que serán de gran utilidad en la elaboración del código fuente de la aplicación. Posee características como la gestión automática de memoria, la eficiencia y la seguridad de tipos que le brindan robustez y limpieza al resultado, permitiendo que la aplicación quede eficazmente optimizada (Albahari, 2022).

### **Entorno integrado de desarrollo (IDE)**

Microsoft Visual Studio 2019 es un entorno de desarrollo integrado (IDE) creado por Microsoft para el desarrollo de aplicaciones en Windows, web y cloud. Algunas de sus características principales son(Microsoft, 2023):

- Soporte multiplataforma para crear aplicaciones para Windows, Mac, Linux, iOS, Android, web y cloud.
- Editor de código avanzado con finalización inteligente de código, resaltado de sintaxis, integración con control de versiones y depuración paso a paso.
- Posibilidad de utilizar lenguajes como C#, C++, Python, JavaScript, TypeScript, Visual Basic, etc.
- Herramientas de diseño visual para aplicaciones web, Mobile y de escritorio. Permite crear interfaces de usuario arrastrando controles a un lienzo.

- Potentes herramientas de depuración que permiten ejecutar aplicaciones paso a paso e inspeccionar valores de variables.
- Integración con bases de datos como SQL Server, PostgreSQL, MySQL para el desarrollo de aplicaciones de back-end.
- Compatibilidad con tecnologías web como ASP.NET, Node.js, Python Django, React y Angular para crear aplicaciones robustas.
- Integración con repositorios de código (GitHub, Azure DevOps) para facilitar el trabajo en equipo.
- Generación de informes de pruebas, análisis de código, rendimiento y seguridad.

### Conclusiones del capítulo

En este capítulo, se establecen los cimientos teóricos esenciales que respaldan el ámbito de investigación, centrando la atención tanto en los videojuegos como en los juegos de ritmo. Se realiza un análisis del juego principal, "Rescatando el Son", que sirve como punto focal para la propuesta en estudio. Se destaca la relevancia del género de ritmo dentro de este contexto, explorando su impacto y singularidades. Además, se examinan detalladamente ejemplos representativos de sistemas homólogos, desglosando sus características distintivas. Este análisis comparativo proporciona una comprensión más profunda de las tendencias y particularidades que influyen en el diseño y la experiencia de juego. Se arroja a luz las tecnologías y herramientas fundamentales que podrían desempeñar un papel crucial en la materialización de la solución propuesta. Al identificar estas herramientas, se sientan las bases para la implementación efectiva del proyecto. Asimismo, se expone la metodología de desarrollo de software a utilizar. Este enfoque metodológico establece la estructura y los pasos clave que guiarán el proceso de creación del juego, asegurando una planificación adecuada y una ejecución eficiente.

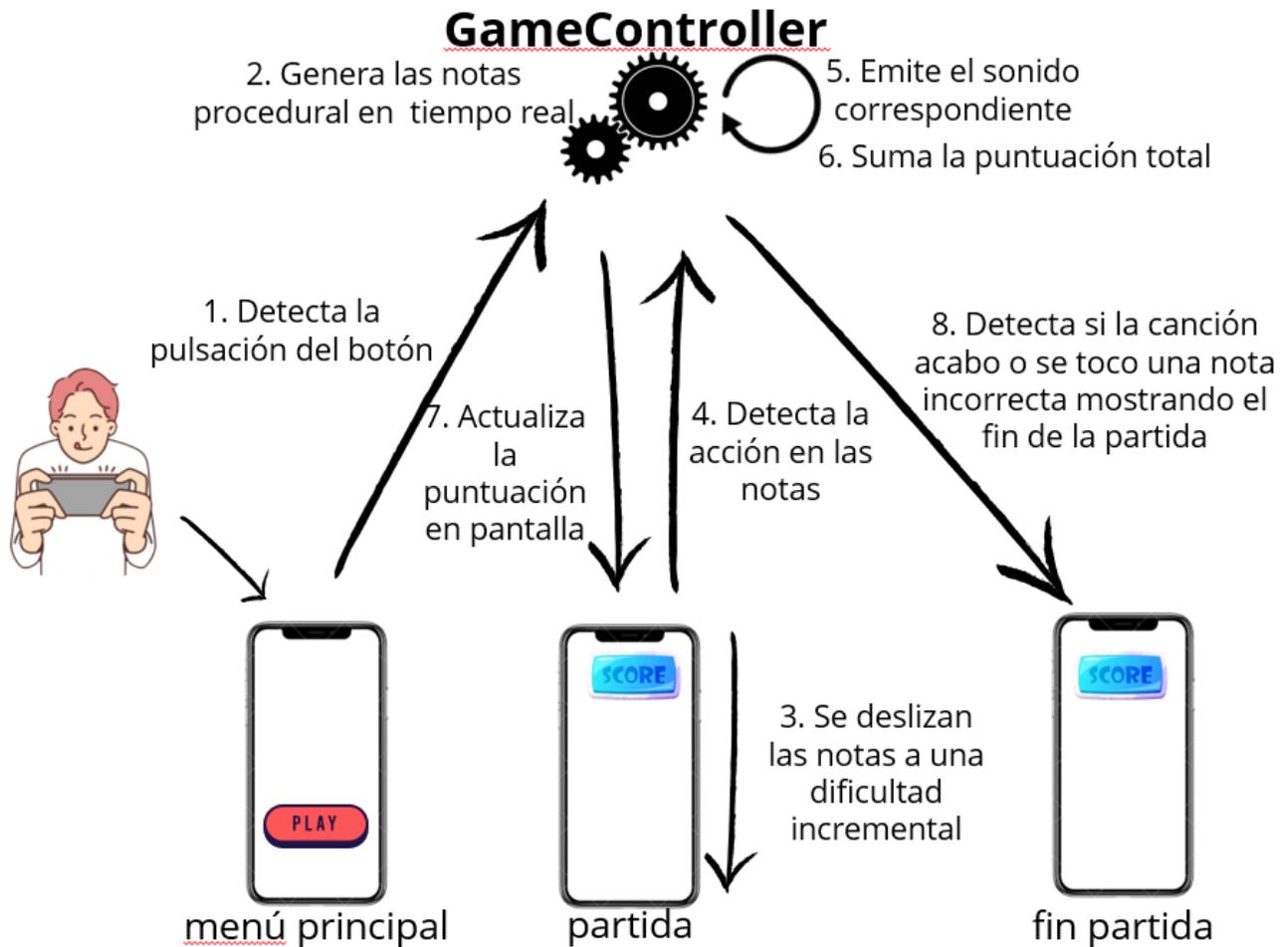
## CAPÍTULO II: Conceptualización y diseño de la solución

En el siguiente capítulo se da a conocer la descripción detallada de la propuesta de solución. Los procesos y actividades descritas se realizan según lo propuesto por el Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos. Se realiza la definición de los elementos formales y dramáticos, la especificación de los mecanismos, la descripción de los patrones de diseño aplicados y la representación de los diagramas de clases del diseño y la arquitectura utilizada.

### 2.1 Propuesta de solución

Para dar respuesta al objetivo de esta investigación, luego de analizar los sistemas homólogos del videojuego y realizar la selección de las herramientas para la implementación, se definen los elementos a desarrollar. El componente Clave del Son debe contar con las mecánicas de control del menú principal, control de la partida, mostrar la puntuación, generación de notas procedural, efectos de sonido y control de la pantalla de fin de juego. En la siguiente imagen se describen las mecánicas que debe contener el componente:

Figura 5. Propuesta de solución.



Fuente. Elaboración propia.

Como se muestra en la figura 5 el componente Clave del Son propuesto está compuesto por mecanismos y mecánicas que se definen a continuación:

**Control del menú principal** la cual contendrá todo lo relacionado con el menú principal, constará con un botón de iniciar la partida. Luego de iniciar la partida empieza la **generación procedural de las notas**. Para este se analiza la canción dividiéndola en beat (es requerido que la canción este en formato MIDI para un mejor análisis). Luego a cada beat se le hace corresponder una nota, capturando la información de la canción en ese momento guardándola en la nota, por cada vez que se detecta un beat, en el caso que no tenga canción se generara una nota vacía. Luego de unos segundos las notas empiezan a caer desde la parte superior

de la pantalla hacia abajo, el jugador ira tocando las notas y estas empezaran a emitir el sonido guardado en esta. Mediante la **detección de interacción con las notas** que es el encargado de detectar y responder a los clics del mouse en notas dentro del juego, Se verifica si se ha presionado el botón izquierdo del mouse (clic). Se crea la variable que guarda y convierte la posición del clic del mouse a coordenadas del mundo. Se crea otra variable que guarda y realiza un rayo desde la posición del clic en dirección cero (no se necesita una dirección específica). Si el rayo golpea algo (es decir, si hay una colisión). Se crea la variable `gameObject` que obtiene el objeto `GameObject` que fue golpeado por el rayo. Comprueba si el objeto tiene una etiqueta llamada "Note". Obtiene el componente "Note" asociado al `GameObject` y llama al método que **reproduce el sonido** en el componente "Note". este método está relacionado con la lógica del juego para manejar la interacción con las notas. En él se Declaran de variables necesarias en la clase "GameController", siendo una Variable de tipo entera que almacena el identificador de la última nota reproducida, inicializado en 0 y otra variable con propiedad reactiva de tipo entera que almacena el puntaje del jugador. Se verifica si el juego ha comenzado y no ha terminado, también se verifica si la nota es visible, si lo está se verifica si la nota no ha sido reproducida y si es la siguiente nota en secuencia. De ser así se marca la nota como reproducida, se actualiza el identificador de la última nota reproducida y se incrementa el puntaje del jugador. Además, llama al método encargado de la reproducción de una parte de la canción. Y por último se reproduce una animación de la nota. Si la nota no es visible, inicia una corrutina para finalizar el juego y reproduce una animación de nota incorrecta. Para **reproducir la canción de la nota**, comprueba si el audio no se está reproduciendo y si no es la última nota, de ser así se inicia la reproducción de la canción Comprueba si la diferencia entre la duración total de la canción y el tiempo actual de reproducción es menor o igual a la longitud de un segmento de la canción de cumplirse la comparación marca que esta es la última nota. Se verifica si ya existe una corrutina de reproducción de segmento de canción, detiene esa corrutina e inicia una nueva corrutina. Comprueba si es la última nota. Marca que el jugador ha ganado e inicia una nueva corrutina Moviéndote al escenario final de la partida (Para finalizar el juego). Para la **dificultad** se crea una corrutina que guarde el tiempo que lleva el juego ejecutándose y la comparamos en dependencia en que momentos queremos que varíe la dificultad aumentamos la velocidad de

las notas al caer. También al no presionar una nota y esta llegar al final de la pantalla esta terminara la partida montándose la pantalla final del juego.

## 2.2 Conceptualización de la propuesta de solución

Se propone como solución el minijuego Clave del Son que actuará como componente del videojuego Rescatando el Son, el cual se concibe como un videojuego de ritmo para plataformas móviles, por lo tanto, posee un argumento y jugabilidad sencilla. Como el nombre sugiere, el jugador va pulsando baldosas del piano según vayan deslizándose por la pantalla emitiendo así sonidos de canciones del género del son cubano. El jugador debe tener reflejos rápidos para tocar las baldosas en el momento exacto, ya que el escenario de juego es finito hasta que complete la canción. El jugador no podrá equivocarse al pulsar la pantalla porque esto conlleva a perder la partida. Una vez terminada la canción se obtendrá la puntuación alcanzada por el jugador y se le brindara una serie de recompensas desbloqueando así nuevas provincias y características de la misma relacionada con las grandes personalidades del son cubano. Al ser un juego para dispositivos con sistema Android, se desea que la visión de la pantalla se obstruya lo menos posible por lo que el jugador solo debe pulsar las baldosas con sus manos, ya que este estará programado para que caigan por su cuenta.

### **Jugabilidad**

El jugador tiene la opción de seleccionar la provincia en el menú principal de la cual desea conocer elementos, para ello tendrá que ganar una partida en “Clave del Son”. Estas decisiones afectan en la dificultad de la partida ya que hay provincias con canciones con ritmos más rápidos por lo tanto tendrán mayor complejidad, aportando variación al desarrollo de esta. El jugador también tiene la posibilidad de hacer uso del servicio de ranking de Cosmox, para lo cual deberá identificarse en la sección del perfil. Además, puede acceder a la sección de ayuda para obtener información útil sobre el juego y consultar los autores de los recursos utilizados en la sección de créditos.

Al ser un videojuego perteneciente al género Música o ritmo, la perspectiva será desde la parte superior (Top View). El jugador podrá pulsar las baldosas que aparezcan en la pantalla. Este posicionamiento de la vista de juego permitirá al jugador observar con precisión la aparición de nuevas baldosas. El juego contará con 4 columnas por las que se deslizan las notas, estas imitan la forma del mástil de una guitarra, el cual obligara al jugador a pulsar en solo las cuatro líneas en la pantalla.

Metas para la experiencia del jugador:

- El jugador recurrirá a la experiencia del juego como método rápido y sencillo de entretenimiento.
- El jugador será incentivado a ver cuánto tiempo puede sobrevivir.
- El jugador estará incentivado a mejorar su puntuación personal en cada partida.
- El jugador querrá mejorar su puntuación en el ranking de puntos.

### 2.3 Diseño del Videojuego

A continuación, se definen los elementos principales que conforman el juego: los elementos formales, los elementos dramáticos y prototipos de las pantallas gráficas, así como la descripción de los mecanismos y los artefactos que definen la arquitectura del software.

#### Elementos formales

Los elementos formales definen las características, objetos y comportamientos del juego

#### Jugadores

Tabla 2. Jugador.

<b>Aspecto</b>	<b>Descripción</b>
Invitación a Jugar	Botón de comienzo “Tocar para iniciar”
Cantidad de Jugadores	Un jugador
Roles	Es quien interactúa con el minijuego
Patrón de Interacción	Individual vs Juego



Fuente. Elaboración propia.

## Objetivos

Tabla 3. Objetivos.

Objetivos	Descripción
Tocar la mayor cantidad de baldosas	El objetivo principal del juego es tocar la mayor cantidad de baldosas negras en el tablero. Cumplir con este objetivo permite al jugador obtener una puntuación más alta.
Evitar tocar otros lados de la escena que no sean las baldosas	El jugador debe evitar tocar otros puntos de la pantalla que no sean las baldosas. Tocar otro punto de la pantalla puede resultar en la pérdida del juego.
Mejorar la precisión	El jugador puede esforzarse por mejorar su precisión y evitar tocar baldosas incorrectas. Esto puede ser importante para mantener una cadena de toques exitosos.
Superar el propio récord	Un objetivo común es superar la mejor puntuación o el mejor desempeño anterior del jugador. Esto fomenta la competencia interna y la mejora continua.

Fuente. Elaboración propia.

## Procedimientos

- La partida comienza al seleccionar el minijuego en la provincia del país que se encuentre el jugador.

- Se le presentará un botón para comenzar y acto seguido comenzarán a caer baldosas de piano que al presionarla estas emitirán sonido relacionadas a las canciones típicas del son en esa región.
- El jugador deberá completar el nivel pulsando las baldosas correctas sin equivocarse hasta terminar la canción y sumar la puntuación máxima.
- Al culminar el nivel correctamente este desbloqueará elementos de la casa Llamo correspondiente para seguir avanzando en el juego.

## **Reglas**

- El jugador gana la partida al presionar todas las notas que aparecen en pantalla. Las notas aparecen en la parte superior de la pantalla y se deslizan hasta desaparecer en la parte inferior.
- Si el jugador omite presionar una nota antes que desaparezca en pantalla o equivoca al presionar en otra ubicación pierde la partida y debe empezar del principio.
- En pantalla pueden aparecer elementos sorpresa además de las notas, estos revelan espacios secretos en la casa Llamo al Son. Completar todos los espacios secretos en todos los niveles pueden resultar en premios vía ranking.

## **Conflictos**

El jugador debe presionar la mayor cantidad de baldosas de piano sin equivocarse.

## **Frontera o Límite**

Presionar todas las baldosas correctamente.

## **Resultado**

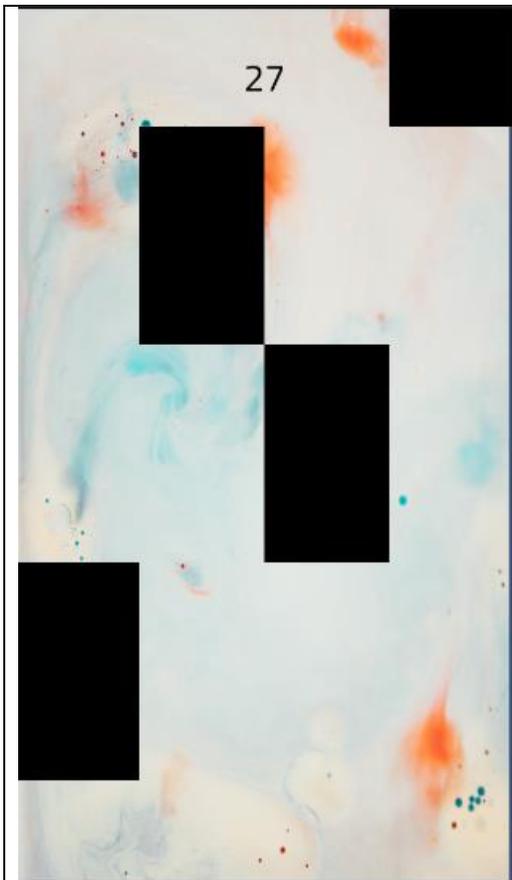
Alcanzar la mayor puntuación en el ranking.

## **Elementos de pantallas gráficas elementales**

A continuación, se muestran prototipos de las pantallas gráficas, así como la descripción de los elementos y comportamientos de estas.

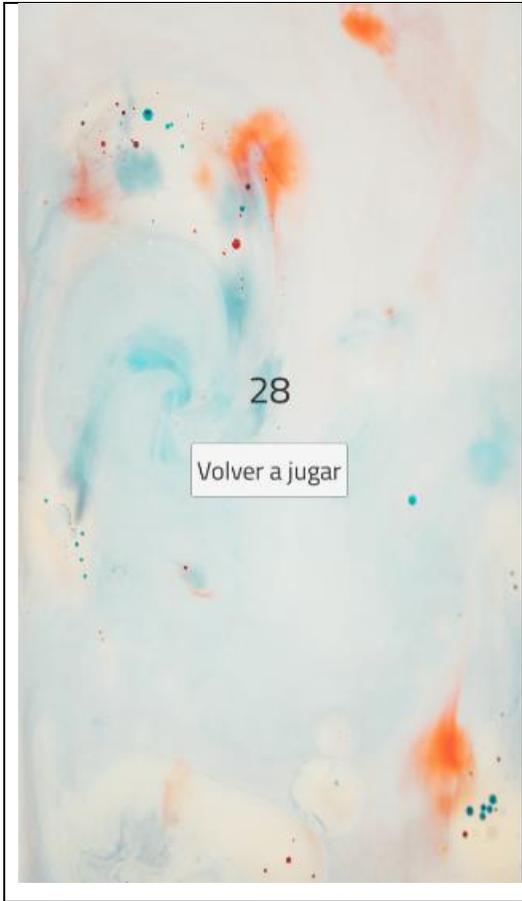
Tabla 4. Elementos de pantallas gráficas elementales.

<b>Diseño</b>	<b>Descripción</b>
	<p>Interfaz inicial del juego. En ella el jugador tiene acceso a:</p> <ul style="list-style-type: none"><li>• Toca para Comenzar: comienza el juego.</li></ul>



Interfaz durante el juego. En ella el jugador tiene acceso a:

- Baldosas de piano: permite al jugador presionar sobre estas.

	<p>Interfaz mostrada cuando el jugador termina su partida. En ella el jugador tiene acceso a:</p> <ul style="list-style-type: none"> <li>• Volver a jugar: reinicia la partida.</li> </ul>
--	--

Fuente. Elaboración propia.

## 2.4 Especificación de mecanismos

Los mecanismos son los elementos mediante los cuales se logra el cumplimiento de un objetivo específico. Los mecanismos controlan distintos aspectos del juego y en conjunto definen el correcto funcionamiento de este (nigromante, 2017). A continuación, se definen los mecanismos necesarios para el videojuego.

Tabla 5. Especificación de mecanismos.

<b>N o</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Organización arquitectónica</b>
1	Mecanismo M1 Control del menú principal	<p><b>Objetos:</b> Botón [Toca para comenzar]</p> <p><b>Propiedades:</b></p> <ul style="list-style-type: none"> <li>• El botón es de color negro.</li> </ul>	Mecanismo núcleo

		<p><b>Comportamiento:</b> una vez iniciado el juego, el jugador podrá interactuar con los elementos de interfaz de usuario, los cuales permitirán comenzar la partida.</p> <p><b>Relaciones:</b> M2</p>	
2	Mecanismo M2 Control de partida	<p><b>Objetos:</b> Baldosas de piano, puntuación.</p> <p><b>Propiedades:</b></p> <ul style="list-style-type: none"> <li>• Todas las baldosas son de color negro.</li> <li>• Las baldosas al ser presionadas reproducen un sonido, parpadean de color azul y desaparecen</li> <li>• Puntos: almacena la cantidad de puntos adquirida por el jugador.</li> </ul> <p><b>Comportamiento:</b> al iniciar la partida aparecen baldosas de pianos aleatoriamente en la pantalla. El mecanismo permite al jugador presionar en la baldosa adecuada para ir sumando puntos y continuar reproduciendo la canción.</p> <p><b>Relaciones:</b>M1, M3, M4, M6, M7</p>	Mecanismo núcleo
3	Mecanismo M3 Mecanismo de generación de notas	<p><b>Objetos:</b> Nota</p> <p><b>Propiedades:</b></p> <p><b>Comportamiento:</b> cuando empieza la partida se generan notas ajustándose al tiempo de la pista de audio seleccionada. Estas se generan en filas con cuatro posiciones posible y se le asigna un ID único a las notas visibles.</p>	Mecanismo núcleo

		<b>Relaciones:</b> M2, M4	
<b>4</b>	Mecanismo M4 Nota	<p><b>Objetos:</b> Nota visible, Nota invisible, Nota fallado</p> <p><b>Propiedades:</b></p> <ul style="list-style-type: none"> <li>• Tipo de nota: las notas de tipo visible son las que el jugador debe tocar, mientras que las notas invisibles deben evitarse ya que al presionarlas sería una nota de fallado.</li> <li>• Posición inicial: Las notas se generan en la parte superior de la pantalla y se desplazan hacia abajo. La posición inicial determina cuánto tiempo tiene el jugador para reaccionar antes de que la nota alcance la parte inferior de la pantalla.</li> <li>• Color: el color de las notas visibles es de color negro mientras que las invisibles son transparente y la de fallado rojo.</li> </ul> <p><b>Comportamiento:</b> cuando comienza la partida este mecanismo se encarga de que las notas se generen de una forma continua y a una velocidad constante. También controla las animaciones de cada nota al interactuar con estas y emitir su sonido.</p> <p><b>Relaciones:</b> M2, M3, M5, M6, M7</p>	Mecanismo núcleo
<b>5</b>	Mecanismo Puntuación M5	<p><b>Objetos:</b> Puntuación</p> <p><b>Propiedades:</b></p> <ul style="list-style-type: none"> <li>• Puntuación acumulativa: la puntuación es acumulativa. Cada vez que un jugador toca</li> </ul>	Mecanismo núcleo

		<p>una nota correctamente, se suman puntos a la puntuación total.</p> <p><b>Comportamiento:</b> al comenzar la partida la puntuación es de cero e irá aumentando cada vez que el jugador toca una nota correctamente al equivocarse se le mostrara la puntuación que hizo hasta ese momento.</p> <p><b>Relaciones:</b>M2, M4, M7</p>	
6	Mecanismo M6 Efectos de sonido	<p><b>Objetos:</b> efecto de sonido</p> <p><b>Propiedades:</b></p> <ul style="list-style-type: none"> <li>• Archivo de sonido: almacena la referencia al archivo de sonido que se reproducirá cuando el jugador toque una nota con éxito. Cada nota puede tener un archivo de sonido asociado.</li> </ul> <p><b>Comportamiento:</b> cuando el jugador toca una nota con éxito, se reproduce el sonido asociado a esa nota. Esto proporciona una retroalimentación auditiva inmediata.</p> <p><b>Relaciones:</b> M2, M4, M7</p>	Mecanismo núcleo
7	Mecanismo M7 Interacción con las notas	<p><b>Objetos:</b> nota, área de interacción, detector de interacción, puntuación, efecto sonoro, finalización de partida.</p> <p><b>Propiedades:</b></p> <ul style="list-style-type: none"> <li>• Posición del clic: almacena la posición del clic del jugador en la pantalla. Esto se compara con la posición de la nota para verificar si el toque fue preciso.</li> <li>• Precisión de Clic: mide la precisión del toque del jugador. Basándose en la</li> </ul>	Mecanismo núcleo

		<p>diferencia entre la posición del clic y la posición de la nota, así como en el tiempo del clic en relación con el tiempo de la nota.</p> <ul style="list-style-type: none"> <li>• Estado de Toque: Un indicador que registra si el toque del jugador fue válido o no. Esta propiedad se utiliza para determinar si se debe sumar puntuación al jugador.</li> <li>• Efecto sonoro: se activa cuando un toque es válido, cambiando la apariencia de la nota y reproduciendo su sonido asociado.</li> <li>• Estado de Partida: indica si la partida está en curso, finalizada o en pausa. Esta propiedad se utiliza para controlar cuándo se pueden realizar toques y cuándo se detiene el juego.</li> </ul> <p>Comportamiento: en el transcurso de la partida cuando el jugador hace clic en una nota en la pantalla se inicia la detección del clic, se verifica si la posición del clic coincide con la posición de la nota. Si es la correcta se considera un toque válido aumentando la puntuación, reproduciendo el sonido asociado de la nota. Si toca la incorrecta como una nota invisible, se considera como una pérdida de la partida y se finaliza.</p> <p><b>Relaciones:</b>M2, M4, M5, M6</p>	
--	--	---	--

*Fuente. Elaboración propia.*

## 2.5 Requisitos no funcionales

Los requerimientos no funcionales representan características generales y restricciones de la aplicación o sistema que se esté desarrollando. Teniendo en cuenta la norma ISO/IEC 25010 se definieron los siguientes requisitos no funcionales(visuresolutions, 2023):

- **RnF 1. Usabilidad**
  - o **Objetivo:** Asegurar que el juego sea fácil de entender y jugar, proporcionando una experiencia intuitiva para los usuarios.
  - o **Métricas:** Tasa de éxito en la realización de acciones, tiempo de aprendizaje para nuevos usuarios.
- **RnF 2. Seguridad**
  - o **Objetivo:** Proteger los datos del usuario y garantizar la integridad del juego.
  - o **Métricas:** Niveles de encriptación de datos, resistencia a ataques externos.
- **RnF 3. Compatibilidad**
  - o **Objetivo:** Garantizar que el juego pueda interactuar adecuadamente con otros sistemas o juegos
  - o **Métricas:** Grado de compatibilidad con estándares.
- **RnF 4. Fiabilidad**
  - o **Objetivo:** Asegurar la estabilidad del juego, minimizando fallos y errores.
  - o **Métricas:** Tiempo medio entre fallos (MTBF), porcentaje de éxito en la ejecución del juego.
- **RnF 5. Portabilidad**
  - o **Objetivo:** Garantizar que el juego sea compatible y se ejecute eficientemente en diferentes plataformas y dispositivos.
  - o **Métricas:** Niveles de compatibilidad con distintos sistemas operativos y dispositivos.

## 2.6 Integración con el juego macro

Como parte de los requisitos de la solución es necesario que los minijuegos se integren con el videojuego Rescatando el Son. De esto se encarga principalmente el juego base, pero hay dos funcionalidades que es necesario implementar en los minijuegos. La primera es la que permite obtener la información de la provincia desde la que el jugador accede al minijuego, y la segunda proporcionarle al juego base la puntuación del jugador al terminar la partida. Para esto se utilizó una clase llamada FileManager que posee dos funciones, una se encarga de leer un archivo que provee el juego base con un valor asociado a cada provincia lo que permite cambiar los elementos de la interfaz y la canción a emitir. La otra se encarga de crear un archivo que guarde la puntuación obtenida en la partida para luego ser cargado por el juego base. En este caso el GenerateScoreFile() se encarga de generar el archivo con la puntuación y guardarlo en la dirección en la que posteriormente será importado por el juego base. La función ReadDifficultyFile() es la que lee el valor del archivo proporcionado por el juego base para cambiar los elementos del minijuego según la provincia.

## 2.7 Paquetes de mecanismos

Un paquete es un mecanismo que permite organizar los elementos modelados con UML, facilitando de esta forma el manejo de los modelos de un sistema complejo. Permiten dividir un modelo para agrupar y encapsular sus elementos en unidades lógicas individuales. Los paquetes pueden estar anidados unos dentro de otros, y unos paquetes pueden depender de otros paquetes. Se pueden utilizar para plantear la arquitectura del sistema a nivel macro. Muestra cómo está estructurado el sistema, cada paquete puede contener otros paquetes o clases, que tienen interfaces y realizan cierta funcionalidad. También se pueden mostrar algunas clases dentro de los paquetes, así como las relaciones de dependencia de estas clases con otras clases o paquetes (López, 2023)

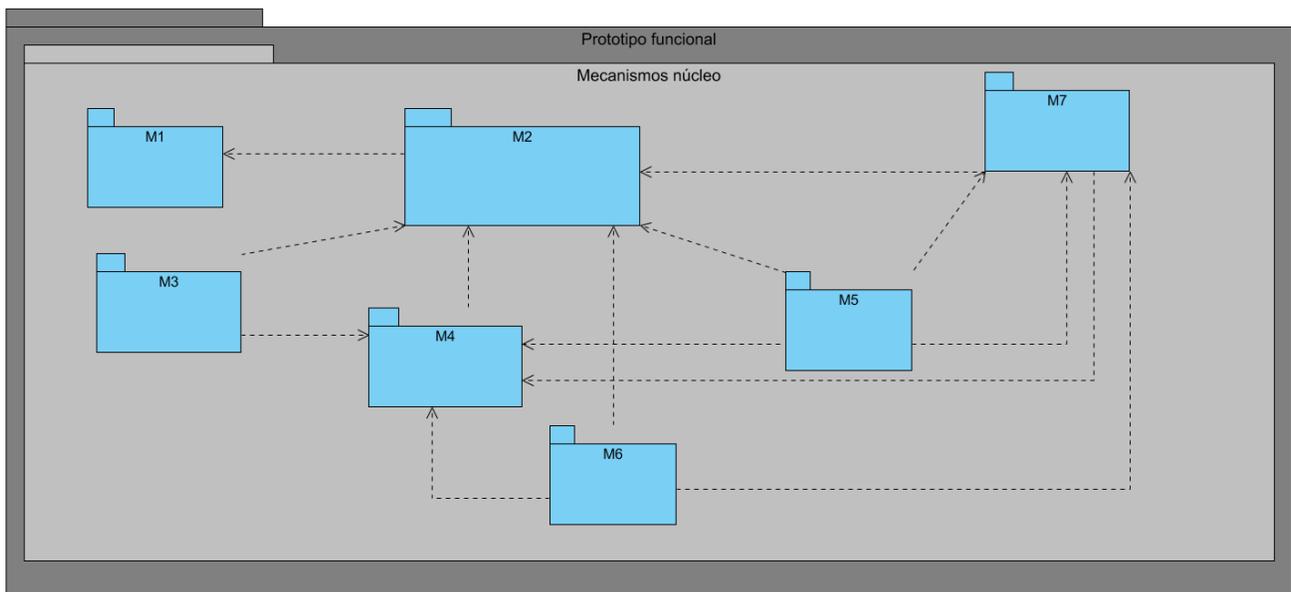
- Mecanismos Núcleo: representan a los mecanismos centrales del videojuego que constituyen el medio para captar la atención jugador y el propósito esencial del juego. Permiten la interacción del jugador con cada uno de los elementos que

contienen las escenas, así como la navegación por ellas, con el propósito de cumplir con los niveles del juego.

- Mecanismo Alternativo: representan a los mecanismos que aportan al propósito general del juego, pero no constituyen elementos fundamentales de este para su funcionamiento. Permiten la incorporación de elementos que enriquecen las opciones de interacción con el videojuego.

A continuación, se muestra el diagrama de paquetes de mecanismos donde se representa el propósito general del juego:

Figura 6: Diagrama de paquetes de mecanismos



Fuente. Elaboración propia.

- M1 Control del menú principal.
- M2 Control de partida.
- M3 Generación de notas.
- M4 Nota.
- M5 Puntuación.
- M6 Efectos de sonidos.
- M7 Interacción con las notas.

## 2.8 Estilo arquitectónico

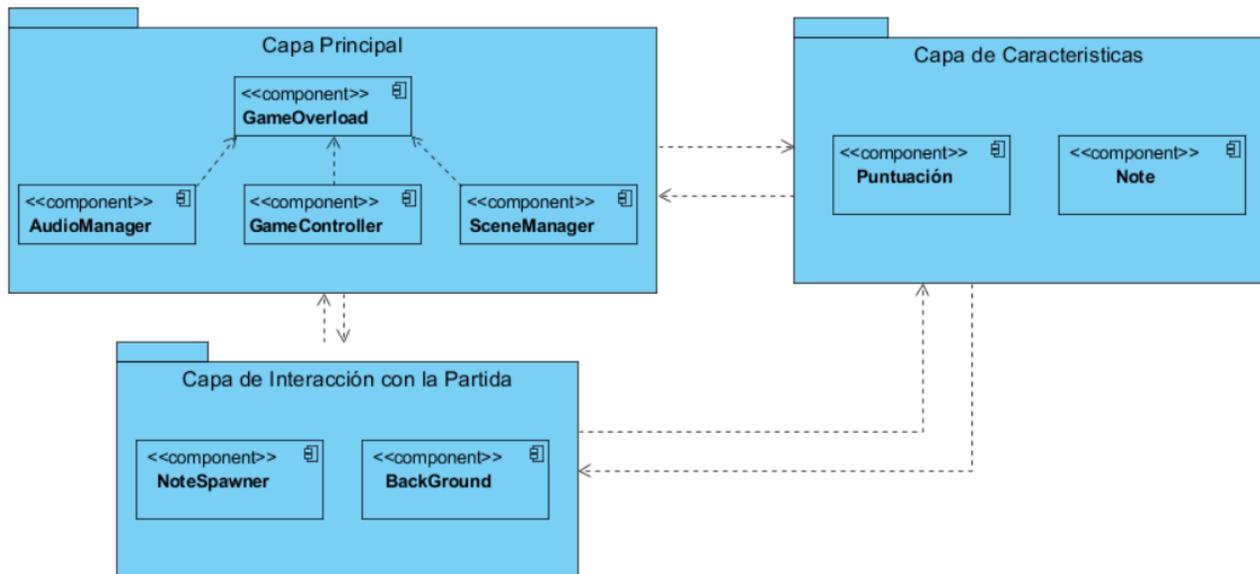
Un estilo arquitectónico es un conjunto coordinado de restricciones arquitectónicas que restringe los roles o rasgos de los elementos arquitectónicos y las relaciones permitidas entre ellos dentro de la arquitectura que se conforma a ese estilo. El estilo llamada y retorno se caracteriza por la descomposición jerárquica en subrutinas (componentes) que solucionan una tarea o función definida. Los datos son pasados como parámetros y el manejador principal proporciona un ciclo de control sobre las subrutinas. Este estilo arquitectónico permite al diseñador del software construir una estructura de programa relativamente fácil de modificar y ajustar a escala y se basa en la bien conocida abstracción de procedimientos, funciones y métodos (en el documento Diseño e implementación de funcionalidades que se llevan a cabo en los registros mercantiles: Solicitudes de expedientes, copias de documentos y sellado de libros, 2023).

### **Arquitectura de software**

La arquitectura de la solución propuesta incluye los elementos fundamentales de la tesis “Arquitectura de software para videojuegos desarrollados sobre el motor de juego Unity3D” (en documento Diseño e implementación de funcionalidades que se llevan a cabo en los registros mercantiles: Solicitudes de expedientes, copias de documentos y sellado de libros, 2023). La arquitectura está compuesta por la combinación de los patrones arquitectónicos: arquitectura basada en capas y arquitectura basada en componentes, para estructurar los elementos del videojuego. Los componentes de cada capa se comunican con otros componentes en otras capas a través de interfaces definidas o instancias de clases. Para cada componente se definió una clase con sus atributos y operaciones para ejecutar los métodos y acciones relevantes. Para que las clases puedan comunicarse con otras en dependencia de la capa donde se encuentren fueron definidas las interfaces e instancias necesarias.

Distribución de las capas presentes en la arquitectura:

Figura 7. Capas de la arquitectura de la propuesta de solución.



Fuente. Elaboración propia.

**Capa Principal:** en esta capa se encuentran los controladores principales del juego.

- **GameController:** es el controlador encargado del funcionamiento del juego, así como los elementos de la interfaz de esta. Se implementa como un `GameObjec` en la escena de juego. Guarda instancias de las notas y los elementos de la interfaz.
- **SceneManager:** es el controlador encargado de gestionar las escenas. Tiene las funcionalidades para cambiar o recargar las escenas.
- **AudioManager:** es el controlador encargado de reproducir la música y los efectos de sonido cuando sean necesarios.
- **GameOverlord:** es el controlador encargado del funcionamiento del juego. Guarda instancias de los elementos necesarios en todas las escenas y las mantiene disponibles para su uso. Se encarga de las transiciones entre escenas y el tratamiento de estos.

**Capa de Interacción con la Partida:** en esta capa se encuentran los componentes que permiten el desarrollo del videojuego mediante la interacción del jugador.

- NoteSpawner: es el controlador encargado de iniciar las formaciones de las notas generadas aleatoriamente.
- BackGround: es el controlador encargado de instanciar, mover y destruir el fondo del juego.

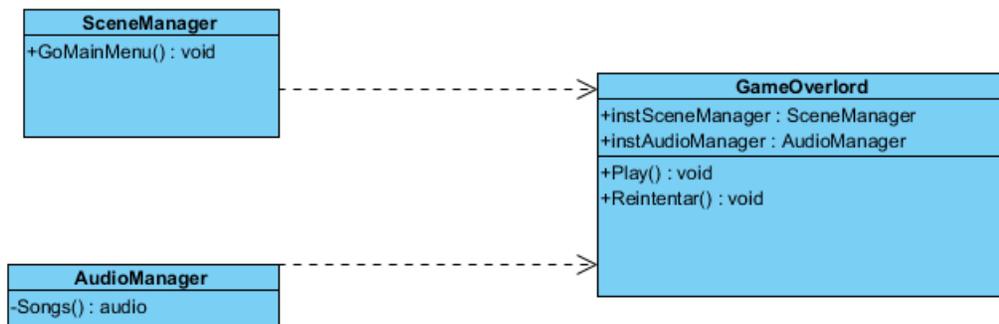
**Capa de Características:** en esta capa se encuentran los componentes que permiten el control de los elementos en el videojuego.

- Puntuación: es el controlador encargado de almacenar la puntuación del jugador
- Note: es el controlador que se encarga de gestionar las notas musicales en el juego.

## 2.9 Diagrama de clases

Para una mejor comprensión de la forma en que se estructura el sistema de clases de la propuesta de solución (miro, 2023), se muestra el siguiente diagrama de clases perteneciente al mecanismo M1 Control del Menú Principal en el Juego. Los diagramas de clases del resto de mecanismos se encuentran en el anexo A1.

Figura 8. Diagrama de clases del mecanismo M1 Control de menú principal.



*Fuente. Elaboración propia.*

## 2.10 Patrones de diseño

Un patrón de diseño proporciona un esquema para refinar los subsistemas o componentes de un sistema software, o las relaciones entre ellos. Describe estructuras repetitivas de comunicar componentes que resuelven un problema de diseño en un contexto particular. Es una información nombrada, instructiva e intuitiva que captura la estructura esencial y la perspicacia de una familia de soluciones probadas con éxito para un problema repetitivo que surge en un cierto contexto y sistema. En el diseño de la solución se utilizarán los Patrones Generales de Software de Asignación de Responsabilidad (GRAPS) y la Banda de Cuatro (GOF) (PDFs de programación » GUÍA DE CONSTRUCCIÓN DE SOFTWARE EN JAVA CON PATRONES DE DISEÑO, 2018).

Patrones GRASP usados en la solución:

- Experto en información: Suele conducir a diseños donde los objetos se hacen responsables de las mismas operaciones. Tiene como beneficio que se mantiene el encapsulamiento de la información, lo que posibilita menor acoplamiento. Se distribuye el comportamiento entre las clases que contienen la información requerida, posibilitando clases más ligeras logrando mayor cohesión (Botero Tabares, 2010). El uso de la herramienta Unity facilita el uso de este patrón ya que se puede asignar como un elemento de los GameObjects el fragmento de código que establecerá su comportamiento o tareas como en el caso de SpawnNotes.
- Alta cohesión: Es una medida que determina cuán relacionadas y adecuadas están las responsabilidades de una clase, de manera que no realice un trabajo superior al estrictamente necesario. Una clase con baja cohesión tiene muchas responsabilidades con poca relación entre sí, haciéndola difícil de comprender, reutilizar y conservar, mientras que una alta cohesión favorece la claridad del sistema y disminuye el acoplamiento (Botero Tabares, 2010). En la implementación de la solución esto se evidencia en la separación de las responsabilidades de la

clase GameController que dieron lugar a la implementación de las clases SpawnNotes y Background para especializar sus responsabilidades.

- Bajo acoplamiento: Es una medida de la fuerza con la que se relacionan y del grado de focalización de las responsabilidades de un elemento. Una clase con baja cohesión tiene muchas responsabilidades, difíciles de entender, reutilizar, mantener, delicadas y frágiles. Tiene como beneficio que incrementa la claridad y comprensión del sistema, se simplifica el mantenimiento, favorece una alta cohesión y facilita la reutilización (Botero Tabares, 2010). Por ejemplo, las notas y la puntuación no son controlados por estos elementos, esa responsabilidad se divide en Nota y Puntuación, guardando relación con el uso del patrón Experto en información.
- Controlador: Este patrón se manifiesta mediante un objeto responsable del manejo de los eventos del sistema, que no pertenece a la interfaz de usuario. El controlador recibe la solicitud del servicio desde la interfaz y coordina su realización delegando a otros objetos. En el proceso de desarrollo se utiliza la clase GameOverlord, que se encarga de las acciones en el software.
- Creador: Este patrón es el responsable de asignarle a una clase determinada la responsabilidad de crear una instancia de las demás clases referentes al sistema. El mismo se evidencia en la clase SpawnNotes, donde la misma es responsable de crear las instancias de clases necesarias que están implicadas en la creación de las notas.

Patrones GOF usados en la solución:

- Única instancia (Singleton): El patrón de diseño Singleton se asegura de que una clase solo tenga una instancia y provee acceso global a ella. El uso de este patrón se evidencia en las clases GameController, GameOverlord, SceneManager y SoundManager, ya que son las clases con información de los elementos del videojuego que se necesitan estén accesibles en todo momento, permitiendo la optimización de código y la sobrecarga de los recursos.  
En la solución se implementa el patrón Singleton en la clase GameController, ya que esta es la encargada de brindar un medio de comunicación entre distintas clases y

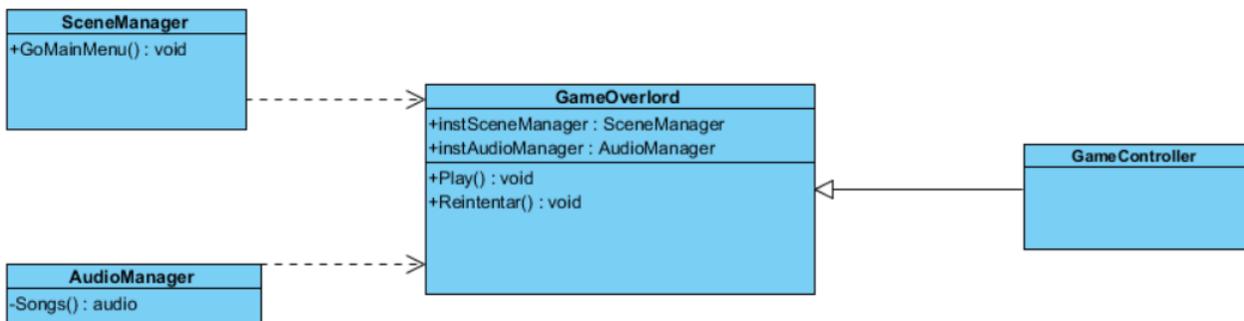
de mantener una instancia única de sí misma para que no haya datos duplicados. El empleo de este patrón permite la persistencia de los datos entre escenas y los mantiene públicos para su uso por otras clases del sistema.

```
private void DetectStart()
{
    if (!GameController.Instance.GameStarted.Value && Input.GetMouseButtonDown(0))
    {
        GameController.Instance.GameStarted.Value = true;
    }
}
```

Figura 8. Patrón de diseño Singleton implementado en la solución.

- Fachada (Facade): El patrón Fachada provee una interfaz a una serie de clases para evitar el uso de varias instancias, dejando a las subclases accesibles para ser usadas directamente. Se implementa en el GameOverlord, que junto al patrón Singleton es la encargada de gestionar los elementos principales del videojuego, permitiendo el acceso al resto de las clases asociadas a ella.

Figura 9. Patrón de diseño Fachada implementado en la solución.



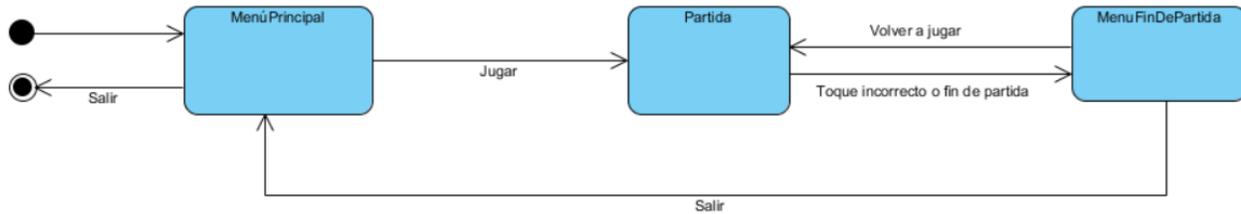
Fuente. Elaboración propia.

## 2.11 Representación de comportamiento

Para representar el comportamiento del videojuego según la acción del jugador se emplean los diagramas de estados (Celka, 2022). En la figura 9 se muestra el diagrama de estados del

mecanismo M2 Control de partida, donde se muestran las posibles acciones que puede realizar el jugador y los estados destino a los que puede llegar en este escenario.

Figura 10. Diagrama de estado del mecanismo M2 Control de partida.



*Fuente. Elaboración propia.*

### Conclusiones del capítulo

Con la finalización de este capítulo, que inicia con la definición de los elementos formales y dramáticos, desentrañando las complejidades que dan forma a la experiencia de juego. A través de esta revelación, se busca no solo comprender la mecánica del videojuego, sino también capturar la esencia que cautivará a los usuarios. La especificación de los mecanismos representa un paso crucial hacia la materialización de la solución. Cada engranaje y componente se disecciona con precisión, delineando su función específica y su contribución al conjunto armonioso del juego. Asimismo, se detalla la aplicación de patrones de diseño, revelando las decisiones estratégicas que dan coherencia y eficiencia al desarrollo del videojuego. Estos patrones, cuidadosamente seleccionados, reflejan no solo buenas prácticas, sino también una comprensión profunda de las necesidades específicas del proyecto. La culminación de este capítulo se alcanza con la representación de los diagramas de clases del diseño y la arquitectura utilizada. Estos diagramas sirven como mapas detallados, guiando la implementación con precisión y ofreciendo una visión clara de la estructura que sostiene la propuesta de solución.

## CAPÍTULO III: Implementación y pruebas de la solución

En el presente capítulo se representa la estructura del videojuego a nivel de componentes. Se describe el estándar de codificación y se explican los resultados obtenidos en las pruebas alfa, beta y de rendimiento.

### 3.1 Estándar de codificación

El objetivo de los estándares de codificación de software es inculcar prácticas de programación comprobadas que conduzcan a un código seguro, confiable, comprobable y mantenible. Por lo general, esto significa evitar prácticas de codificación no seguras conocidas o código que pueda causar un comportamiento impredecible (Hamilton, 2020).

En la implementación de la solución se usó el estilo de escritura CamelCase. El nombre viene porque se asemeja a las dos jorobas de un camello, y se puede dividir en dos tipos (Leino, 2016):

- UpperCamelCase: usado en la declaración de métodos y clases de la solución.
- LowerCamelCase: usada en la declaración de variables.

Vista del código

Definición de clases: Las clases comienzan con inicial mayúscula al inicio de la palabra y en caso de estar conformada por palabras compuestas, la definición debe ser continua y cada palabra debe iniciar con mayúscula siguiendo el estándar determinado. Un ejemplo de ello se muestra a continuación:

```
public class GameController : MonoBehaviour
```

Declaración de métodos: El nombre de los métodos debe comenzar con mayúscula y en caso de que el nombre esté compuesto por más palabras cada una debe iniciar con mayúscula. A continuación, se presenta un ejemplo de cada caso:

```
public void SpawnNotes()
```

```
private void OnTriggerEnter2D(Collider2D collision)
```

Declaración de variables: Los nombres de las variables comienzan con minúscula y en caso de estar compuesto por más palabras la segunda debe empezar con mayúscula siguiendo el estándar determinado, como se muestra en los siguientes ejemplos:

```
private float songSegmentLength = 0.8f;
private bool lastNote = false;
private bool lastSpawn = false;
```

Estilos de implementación: El estilo utilizado en la implementación de la propuesta de solución es propio para lenguajes de programación que usan llaves para delimitar bloques lógicos de códigos. A continuación, se presenta un ejemplo de su utilización:

```
private void DetectNoteClicks()
{
    if (Input.GetMouseButtonDown(0))
    {
        var origin = Camera.main.ScreenToWorldPoint(Input.mousePosition);
        var hit = Physics2D.Raycast(origin, Vector2.zero);
        if (hit)
        {
            var gameObject = hit.collider.gameObject;
            if (gameObject.CompareTag("Note"))
            {
                var note = gameObject.GetComponent<Note>();
                note.Play();
            }
        }
    }
}
```

### 3.2 Assets y recursos utilizados

#### **Assets**

Como parte de la implementación del mecanismo M4, se utiliza un elemento de animación para cuando se presiona correctamente una nota o falles, así como para que se hagan invisible bajo una licencia estándar de usuario final (EULA). Los elementos conocidos como Invisible, Missed, Note y Played forma parte del paquete de Animación, disponible en la Unity Assets Store, el cual está pensado para incorporar de forma rápida y sencilla estas animaciones para videojuegos basados en el sistema Android (Delgado et al., 2018).

#### **Efectos de sonido**

Los efectos de sonidos fueron obtenidos de [samplefocus.com](http://samplefocus.com), un portal web que proporciona ejemplos de audio, sonidos y loops gratuitos para descargar, libres de derechos para usar en producciones musicales, videos, juegos, y más. Cuentan con una amplia librería de sonidos organizados en categorías como pianos, guitarras, baterías, sintetizadores, voces, Foley, etc (Sample Focus, 2023).

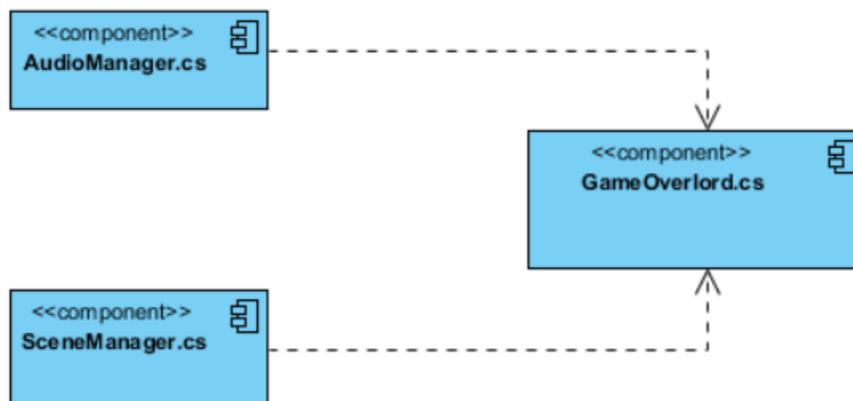
## Imágenes

Las imágenes utilizadas para el diseño de la interfaz de usuario fueron obtenidas en [Vecteezy.com](http://Vecteezy.com) bajo la licencia Creative Commons attribution (*Download Free Vectors, Clipart Graphics, Vector Art & Design Templates*, s. f.). Este sitio ofrece una gran variedad de imágenes vectoriales de todo tipo. Las imágenes seleccionadas fueron modificadas en la herramienta GIMP para añadir transparencia y cambiarle el color a blanco, con el fin de aumentar su versatilidad dentro del motor de juego Unity.

### 3.3 Diagramas de componentes

En la Figura 10 se presenta el diagrama de componentes (diagramasuml, 2023) perteneciente al mecanismo M1 Control del Menú Principal. Con el mismo se puede apreciar desde una vista más cercana al código, cómo el videojuego es estructurado en diferentes componentes que se relacionan entre sí. La representación de los demás mecanismos se encuentra en el anexo.

Figura 11. Diagrama de componentes M1 Control del menú principal.



Fuente. Elaboración propia.

### 3.4 Pruebas

Al concluir el proceso de desarrollo del videojuego y encontrarse en estado funcional, se desarrolla un conjunto de pruebas con el fin de validar todos los mecanismos. Las pruebas aplicadas se hicieron tomando en cuenta los pasos descritos por la metodología seleccionada, que trabaja sobre la base del análisis del prototipo compilado (Alex Zap Chernyak, 2023).

#### **Pruebas alfa y beta**

- Pruebas Alpha: estas pruebas las realiza el desarrollador. Se usa el software de forma natural con el desarrollador como observador del usuario y registrando los errores y problemas de uso. Las pruebas alfa se hacen en un entorno controlado (Alex Zap Chernyak, 2023).
- Pruebas Beta: estas pruebas las realizan los usuarios finales del software en los lugares de trabajo de los clientes. A diferencia de la prueba alfa, el desarrollador no está presente normalmente (Alex Zap Chernyak, 2023).

#### 3.5.1 Pruebas Alpha

A continuación, se describe el registro de no conformidades (NC) encontrados durante las pruebas Alpha:

- Iteración 1: se realizan pruebas al mecanismo M1 Control del menú principal.
  - NC 1: No realiza la acción esperada el botón de jugar.
    - Nombre del probador: Kevin Padrón Domínguez.
    - Localización del error: Menú Principal.
    - Tipo de error: app-error de interfaz.
    - Impacto: alto.
    - Estado del defecto: resuelto.
    - Respuesta del equipo: resuelto.
  - NC 2: Error ortográfico en la palabra “comenzar” está escrita con la letra s.
    - Nombre del probador: Kevin Padrón Domínguez.
    - Localización del error: Menú Principal.

- Tipo de error: app-Doc/ortografía.
  - Impacto: alto.
  - Estado del defecto: resuelto.
  - Respuesta del equipo: resuelto.
- Iteración 2: Se realizan pruebas al mecanismo M2 Control de partida, M3 Generación de notas, M4 Nota, M5 Puntuación, M6 efecto de sonido, M7 Interacción con las notas.
  - NC 1 Las notas no se destruyen al salir de la pantalla.
    - Nombre del probador: Kevin Padrón Domínguez.
    - Localización del error: Pantalla de juego.
    - Tipo de error: app-funcionalidad.
    - Impacto: alto.
    - Estado del defecto: resuelto.
    - Respuesta del equipo: resuelto.
  - NC 2 Solo se genera una nota.
    - Nombre del probador: Kevin Padrón Domínguez.
    - Localización del error: Pantalla de juego.
    - Tipo de error: app-funcionalidad.
    - Impacto: alto.
    - Estado del defecto: resuelto.
    - Respuesta del equipo: resuelto.
  - NC 3 Todas las notas emiten el mismo sonido.
    - Nombre del probador: Kevin Padrón Domínguez.
    - Localización del error: Pantalla de juego.
    - Tipo de error: app-funcionalidad.
    - Impacto: alto.
    - Estado del defecto: resuelto.
    - Respuesta del equipo: resuelto.
  - NC 4 El juego no termina cuando se da un toque erróneo.
    - Nombre del probador: Kevin Padrón Domínguez.

- Localización del error: Pantalla de juego.
  - Tipo de error: app-funcionalidad.
  - Impacto: alto.
  - Estado del defecto: resuelto.
  - Respuesta del equipo: resuelto.
- o NC 5 Las notas presionadas no suman puntos al total de puntos de la partida.
    - Nombre del probador: Kevin Padrón Domínguez.
    - Localización del error: Pantalla de juego.
    - Tipo de error: app-funcionalidad.
    - Impacto: alto.
    - Estado del defecto: resuelto.
    - Respuesta del equipo: resuelto.

### 3.5.2 Pruebas Beta

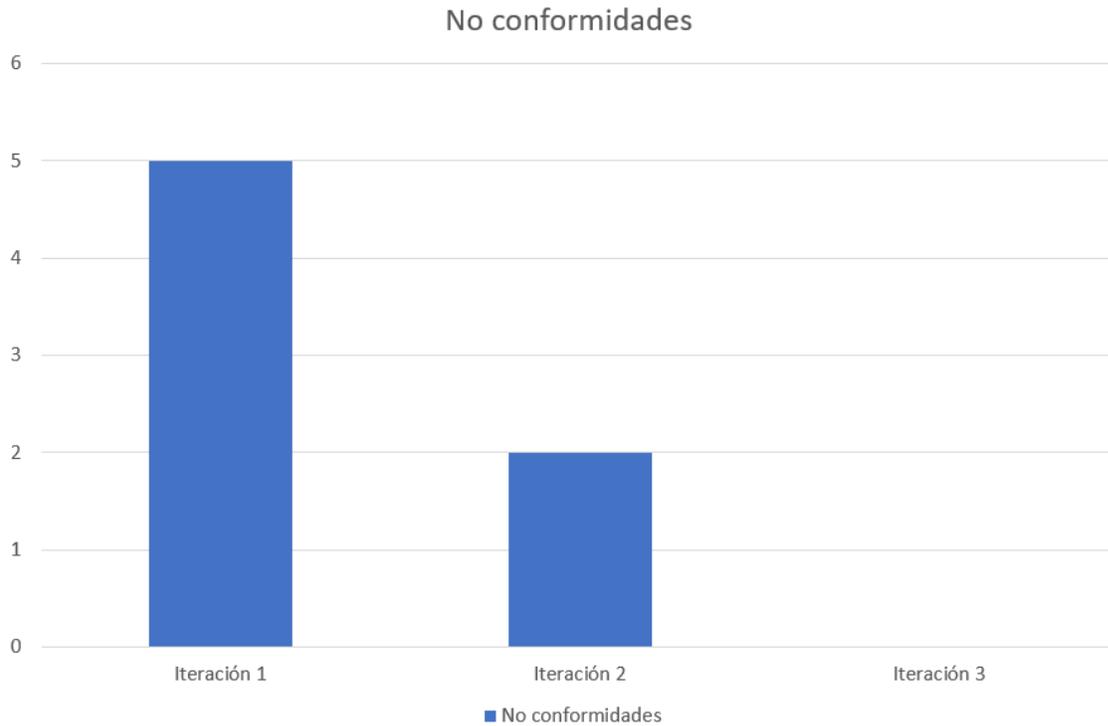
Para la confección de las pruebas Beta, se selecciona un equipo de jugadores para que prueben el funcionamiento del videojuego, los cuales se relacionan a continuación:

- Elier Evian Miranda Abreu (estudiante de 5to año en Ingeniería en Ciencias Informáticas facultad 4).
- Josué Hernández Martínez (estudiante de 5to año en Ingeniería en Ciencias Informáticas facultad 4).
- Leandro Ruiz Domínguez (Estudiante de medicina).
- Johan Chirino Montano (Ingeniero hidráulico).
- David Ricardo Delgado Pérez (estudiante de arquitectura).

Se realizaron tres iteraciones de las pruebas Beta, cuyos resultados pueden ser consultados en la sección de los anexos. Las no conformidades arrojadas por las pruebas en una iteración fueron resueltas antes de iniciar la siguiente. En la tercera iteración se registran cero no

conformidades por parte del personal que realizó las pruebas, por lo que no fue necesario realizar otra iteración. En la Figura 11 se muestra un resumen de este proceso de validación.

Figura 12 No conformidades.



Fuente. Elaboración Propia.

### Conclusiones del capítulo

En este capítulo, hemos explorado la esencia estructural del videojuego a través de la representación detallada de sus componentes. A medida que desglosamos la arquitectura, se revelaron las conexiones entre cada elemento, proporcionando una visión profunda de la maquinaria que impulsa la experiencia de juego. Se abordó el estándar de codificación, destacando la importancia de seguir prácticas coherentes y eficientes en el proceso de desarrollo. La consistencia en el código no solo promueve la legibilidad y mantenibilidad, sino que también contribuye significativamente a la estabilidad y eficacia del videojuego. Las pruebas alfa, beta y de rendimiento han sido pilares fundamentales en la validación de nuestra propuesta. Los resultados obtenidos no solo reflejan la robustez de la implementación, sino

que también proporcionan valiosas perspectivas sobre áreas de mejora y refinamiento. Las pruebas alfa ofrecieron una visión temprana y detallada, permitiendo ajustes iniciales, mientras que las pruebas beta destacaron la capacidad del juego para enfrentar condiciones variadas y la eficiencia de los recursos utilizados.

## CONCLUSIONES

Tras culminar la investigación y el proceso de desarrollo de la propuesta de solución se arribaron a las siguientes conclusiones:

- Se ha llevado a cabo una exhaustiva revisión del estado del arte de los videojuegos de ritmo, proporcionando un marco teórico sólido. Este análisis ha permitido identificar tendencias, características y desafíos clave en el género, sentando las bases conceptuales para la propuesta de solución.
- La investigación ha profundizado en la caracterización del desarrollo de videojuegos pertenecientes al género de ritmo. Se han explorado los elementos formales y dramáticos, los mecanismos específicos y los patrones de diseño aplicados, ofreciendo una visión completa del proceso de desarrollo específico para este género.
- Se ha llevado a cabo un análisis exhaustivo de las herramientas y metodologías pertinentes para el desarrollo de la propuesta de solución. Este análisis ha permitido seleccionar y aplicar las herramientas más apropiadas, así como adoptar una metodología de desarrollo de software que garantice la eficiencia y la calidad del producto final.
- Se ha logrado la implementación exitosa del minijuego como componente propuesto. Este proceso involucró la aplicación de los patrones de diseño y la arquitectura definida, culminando en un componente funcional que refleja la visión conceptual delineada en las fases anteriores.
- La validación del minijuego como componente implementado se ha llevado a cabo mediante pruebas alfa y beta. Los resultados obtenidos han proporcionado valiosos visión sobre la eficacia y estabilidad del juego, permitiendo ajustes y refinamientos finales para garantizar una experiencia de usuario óptima.

## RECOMENDACIONES

A partir de esta primera entrega y como indica la metodología empleada, se puede retomar la parte de diseño y conceptualización para mejorar y añadir contenido al videojuego.

En vista a futuras investigaciones en el campo de videojuegos de ritmos se recomienda:

- Investigar cómo diseñar y mejorar experiencias multijugador en videojuegos de ritmo, ya sea a través de competiciones en tiempo real o colaboraciones que fomenten la interacción social.
- Investigar cómo facilitar y mejorar la creación de contenido por parte de los usuarios en videojuegos de ritmo, permitiendo a la comunidad contribuir con nuevas canciones, patrones y niveles.

## REFERENCIAS BIBLIOGRÁFICAS

- Albahari, J. (2022). C# 10 in a Nutshell. O'Reilly Media, Inc.
- Idas Mena, D. E., & Andrade Cadena, M. A. (2011). Guía práctica para el uso de patrones de diseño en el desarrollo de software [B.S. thesis]. QUITO/EPN/2011.
- Norrie, D. (2022, mayo 17). The History Of Gaming Consoles—An Industry Decades In The Making. *Tell Me Best*. <https://www.tellmebest.com/history-of-gaming-consoles/>
- Hamilton, D. (2020). Estándares de codificación de software y pautas de programación. Parasoft. [https://es.parasoft.com/blog/an-ounce-of-prevention-software ...](https://es.parasoft.com/blog/an-ounce-of-prevention-software...)
- Kuss, M. (1994). Review of Essays on Cuban Music: North American and Cuban Perspectives [Review of Review of Essays on Cuban Music: North American and Cuban Perspectives, por P. Manuel]. *Notes*, 50(3), 934-941. <https://doi.org/10.2307/898544>
- diagramasuml. (2023). ▷ *Diagrama de componentes. Teoría y ejemplos*. DiagramasUML.com. <https://diagramasuml.com/componentes/>
- López, S. (2023). ▷ *Diagrama de paquetes*. DiagramasUML.com. <https://diagramasuml.com/paquetes/>
- Delgado, J. C. S.-, Alvarado, M. A. C.-, Delgado, J. C. S.-, & Alvarado, M. A. C.-. (2018). Análisis comparativo de juegos móviles educativos basados en posicionamiento. *InterSedes*, 19(39), 146-171. <https://doi.org/10.15517/isucr.v19i39.34075>
- desarrolloweb. (2023). C#. <https://desarrolloweb.com/home/c>
- microsoft. (2023, noviembre 16). C# | Lenguaje de programación moderno y de código abierto para .NET. <https://dotnet.microsoft.com/es-es/languages/csharp>
- microsoft. (2023). *Características de desarrollo de Visual Studio*. Visual Studio. <https://visualstudio.microsoft.com/es/vs/features/>
- Pérez, Mariana. (2023, julio 25). *ConceptoDefinición | ¿Qué es el Juego? - Su Definición y Significado [2023]*. <https://conceptodefinicion.de/juego/>
- Centro de Tecnologías Interactivas. (2023). *COSMOX | Portal de Videojuegos Cubanos*. <https://cosmox.uci.cu/>
- Rayark Games. (2023, septiembre 18). *Cytus II | Cytus Wiki | Fandom*. [https://cytus.fandom.com/es/wiki/Cytus\\_II](https://cytus.fandom.com/es/wiki/Cytus_II)
- In document FACULTAD DE INGENIERÍA Y ARQUITECTURA. (2022). *Desarrollo de videojuegos—Bases teórico – científicos*. <https://1library.co/article/desarrollo-de-videojuegos-bases-te%C3%B3rico-cient%C3%ADficos.q5mx8dri>
- miro. (2023). *Diagrama de clases: Qué es, cómo hacerlo y ejemplos | Miro*. <https://miro.com/.https://miro.com/es/diagrama/que-es-diagrama-clases-uml/>
- Celka. (2022, octubre 26). Diagrama de Estado—Definición y explicación. *TechEdu*. <https://techlib.net/techedu/diagrama-de-estado/>
- López Raventós, C. (2016). El videojuego como herramienta educativa. Posibilidades y problemáticas acerca de los serious games. *Apertura (Guadalajara, Jal.)*, 8(1), 0-0.
- In document Diseño e implementación de funcionalidades que se llevan a cabo en los registros mercantiles: Solicitudes de expedientes, copias de documentos y sellado de libros. (2023). *Estilo*

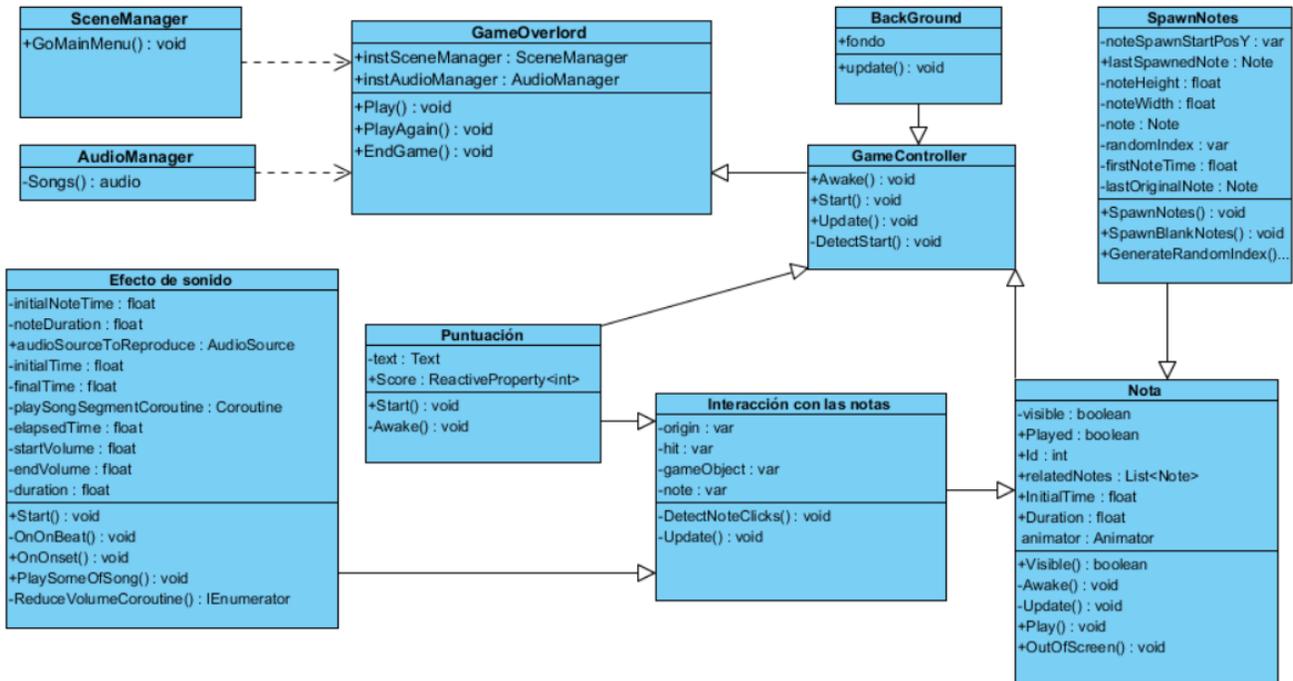
- de Llamada y Retorno—Estilo Arquitectónico. <https://1library.co/article/estilo-de-llamada-y-retorno-estilo-arquitect%C3%B3nico.zkwmo684>
- Dima Stouhi. (2020, mayo 17). *La importancia de la arquitectura en el diseño de videojuegos* | ArchDaily México. <https://www.archdaily.mx/mx/939403/la-importancia-de-la-arquitectura-en-el-diseno-de-videojuegos>
  - Andrés López, U. T. (2023). *Magic Tiles 3 (Android)*. Uptodown. <https://magic-tiles-3.uptodown.com/android>
  - Hernández P., Andy; Pérez T. Karina; Correa M., Omar. (2017, julio 2). *Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos (Engineering framework for the videogame development process)*. OpenAIRE - Explore. <https://explore.openaire.eu/search/publication?pid=10.5281%2Fzenodo.2617304>
  - PDFs de programación» GUÍA DE CONSTRUCCIÓN DE SOFTWARE EN JAVA CON PATRONES DE DISEÑO. (2018, mayo 18). *PDF de programación—GUÍA DE CONSTRUCCIÓN DE SOFTWARE EN JAVA CON PATRONES DE DISEÑO*. <https://www.lawebdelprogramador.com/pdf/9527-GUIA-DE%20CONSTRUCCION-DE-SOFTWARE-EN-JAVA-CON-PATRONES-DE-DISENO.html>
  - Simone Belli y Cristian López Raventós. (2013, marzo). (PDF) *Breve historia de los videojuegos*. [https://www.researchgate.net/publication/39499381\\_Breve\\_historia\\_de\\_los\\_videojuegos](https://www.researchgate.net/publication/39499381_Breve_historia_de_los_videojuegos)
  - AndroidDev. (2023, agosto 21). *Piano Tiles—Apps en Google Play*. [https://play.google.com/store/apps/details?id=com.sonlam.pianotiles&hl=es\\_US](https://play.google.com/store/apps/details?id=com.sonlam.pianotiles&hl=es_US)
  - Alex Zap Chernyak. (2023). *Pruebas alfa: Tipos, proceso, herramientas y mucho más*. <https://www.zaptest.com/es/pruebas-alfa-que-son-tipos-proceso-vs-pruebas-beta-herramientas-y-mucho-mas>
  - Alex Zap Chernyak. (2023). *Pruebas beta: Tipos, proceso, herramientas y mucho más*. <https://www.zaptest.com/es/pruebas-beta-que-son-tipos-procesos-enfoques-herramientas-vs-pruebas-alfa-mas>
  - lucidchart. (2023). *¿Qué es el lenguaje unificado de modelado (UML)?* | Lucidchart. <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>
  - TheBoyan. (2023). *¿Qué significa «bajo en acoplamiento y alto en cohesión»?* <https://qastack.mx/programming/14000762/what-does-low-in-coupling-and-high-in-cohesion-mean>
  - nigromante, L. (2017, julio 9). *¿Qué son las mecánicas de juego? Una aproximación al concepto*. Todas Gamers. <http://todasgamers.com/2017/07/09/las-mecanicas-juego-una-aproximacion-al-concepto/>
  - visuresolutions. (2023). *Qué son los Requisitos No Funcionales: Ejemplos, Definición, Guía Completa*. Visure Solutions. <https://visuresolutions.com/es/blog/requerimientos-no-funcionales/>
  - Sample Focus. (2023). *Sample Focus | The Easiest Way to Find Free Audio Samples*. <https://samplefocus.com/>
  - Uptodown Content Team, U. T. (2023). *Tap Tap Reborn 2 (Android)*. Uptodown. <https://tap-tap-reborn-2.uptodown.com/android>
  - Uptodown Content Team, U. T. (2023). *Tiles Hop (Android)*. Uptodown. <https://tiles-hop.uptodown.com/android>

- Stephen Rhoton. (2023). *Tipos de videojuegos: Todos los géneros y subgéneros*. Liga de Gamers. <https://www.ligadegamers.com/tipos-de-videojuegos/>
- Alfredo y johayana. (2021, junio 2). *Unity ¿Qué es y para qué sirve? | Tutorial Unity*. <https://www.masterd.es/blog/que-es-unity-3d-tutorial>
- wikiwand. (2022, diciembre 4). *Videojuegos de música: Tipos y características - Clebert*. <https://www.clebert.com/videojuegos-de-musica-tipos-y-caracteristicas>
- In document Desarrollo de un Generador de Datos para el proyecto SIGEP. (2022). *Visual Paradigm—Técnicas y herramientas*. <https://1library.co/article/visual-paradigm-t%C3%A9cnicas-y-herramientas.qmjk127>

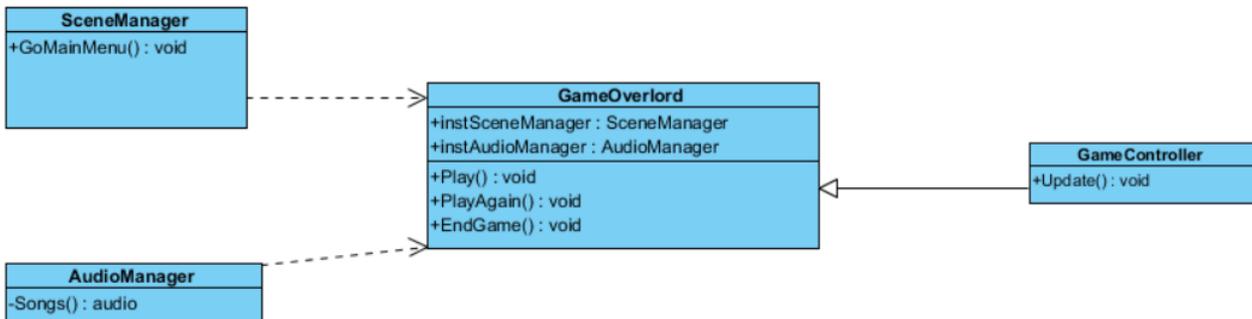
# ANEXOS

## A.1 Diagrama de clases

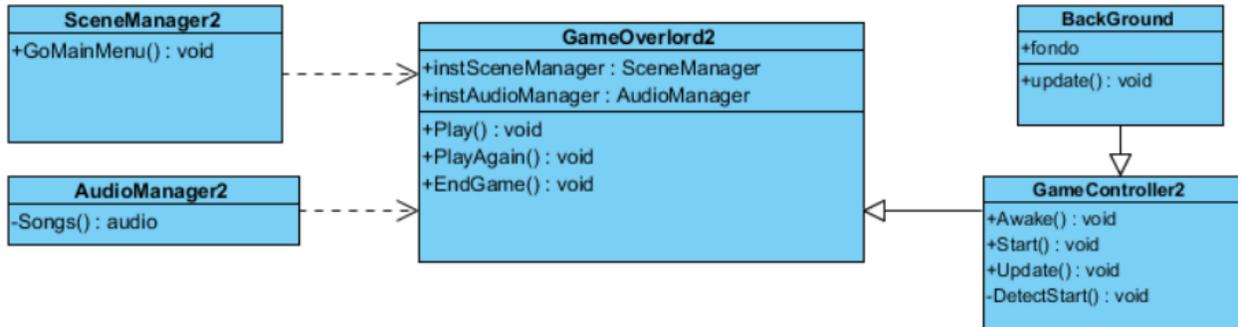
### A 1.1 Diagrama de clases del componente Clave del Son.



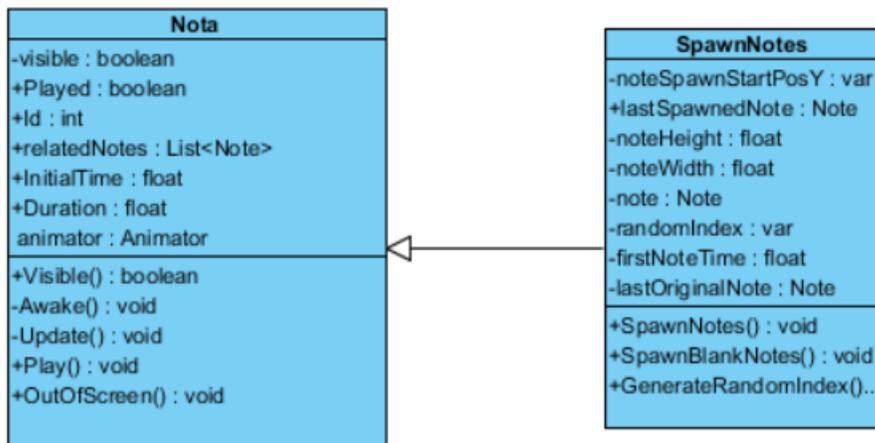
### A 1.2 Diagrama de clases del mecanismo M1 Control del Menú Principal.



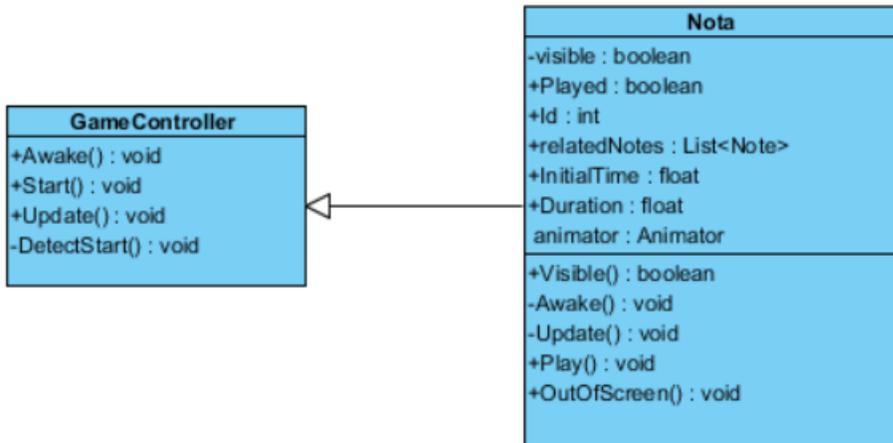
A 1.3 Diagrama de clases del mecanismo M2 Control del Menú Partida.



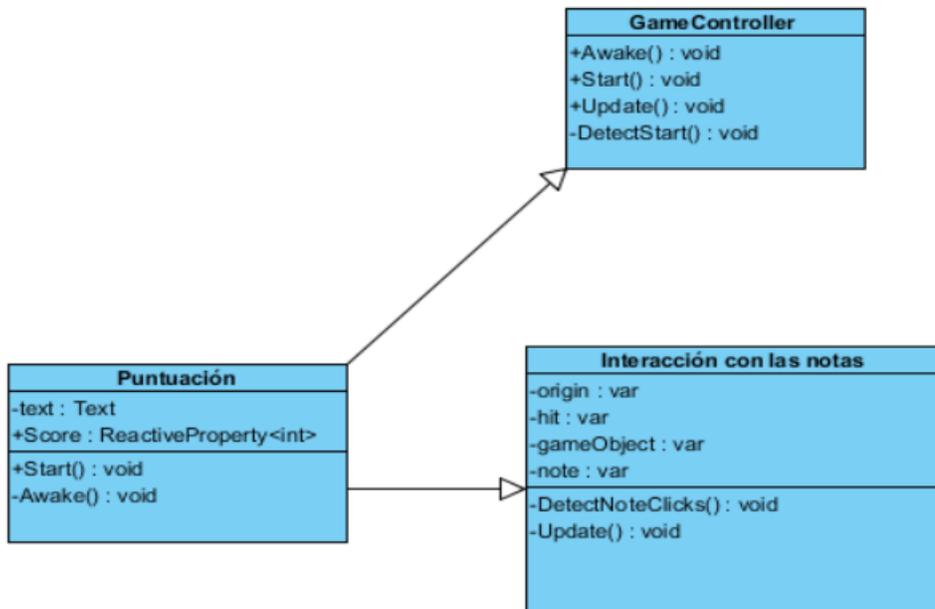
A 1.4 Diagrama de clases del mecanismo M3 Generación de notas.



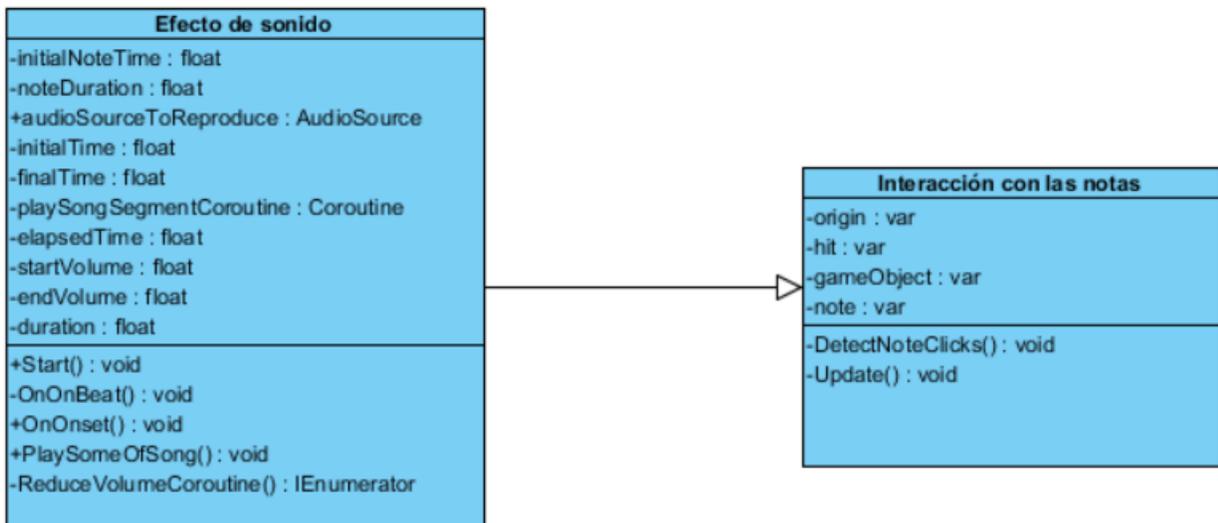
A 1.5 Diagrama de clases del mecanismo M4 Nota.



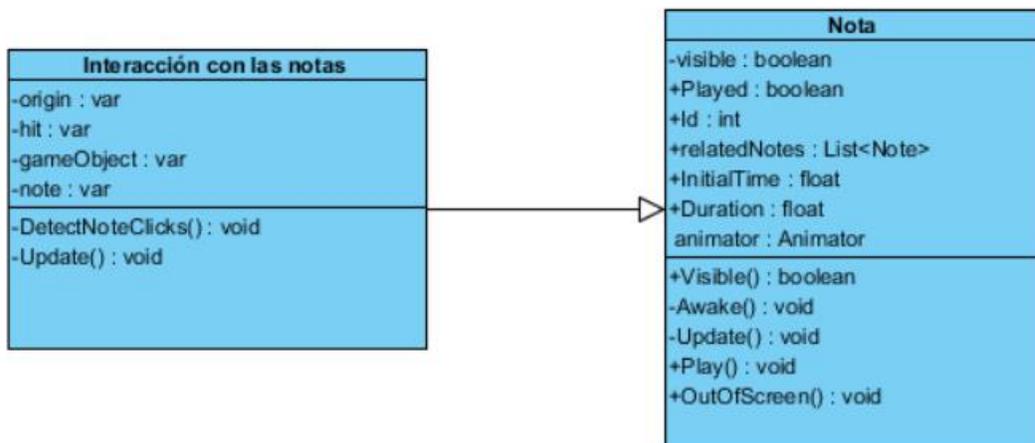
A 1.6 Diagrama de clases del mecanismo M5 Puntuación.



A 1.7 Diagrama de clases del mecanismo M6 Efecto de sonido.

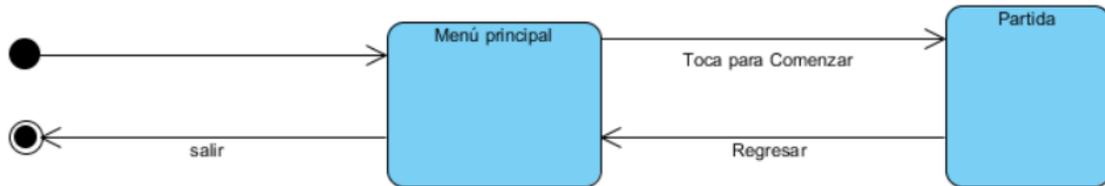


A 1.8 Diagrama de clases del mecanismo M7 Interacción con las notas.

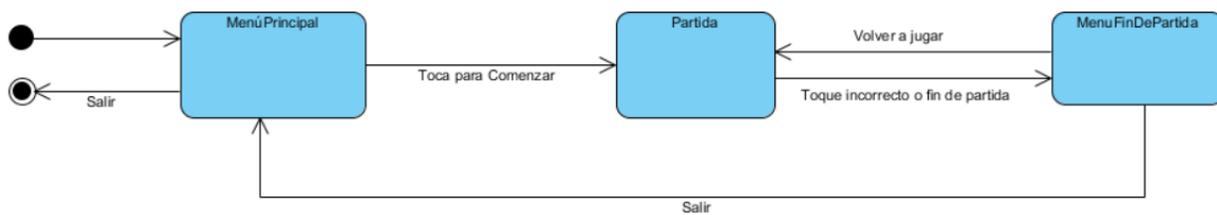


## A.2 Diagrama de estado

### A 2.1 Diagrama de estado del menú principal.

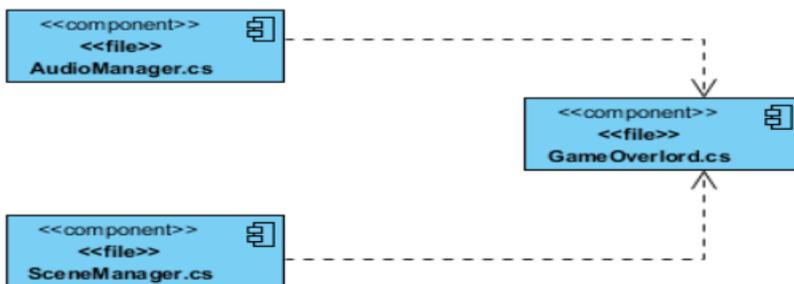


### A 2.2 Diagrama de estado de la partida.

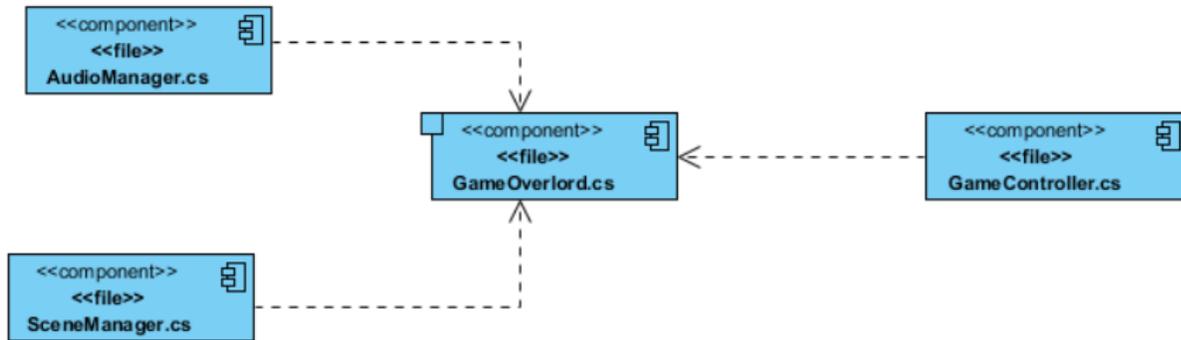


## A.3 Diagramas de componentes

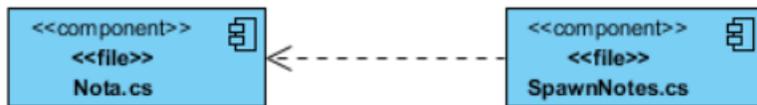
### A 3.1 Diagrama de componentes de mecanismo M1 Control del menú principal.



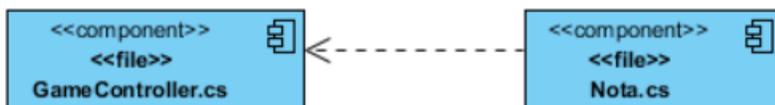
A 3.2 Diagrama de componentes de mecanismo M2 Control de partida.



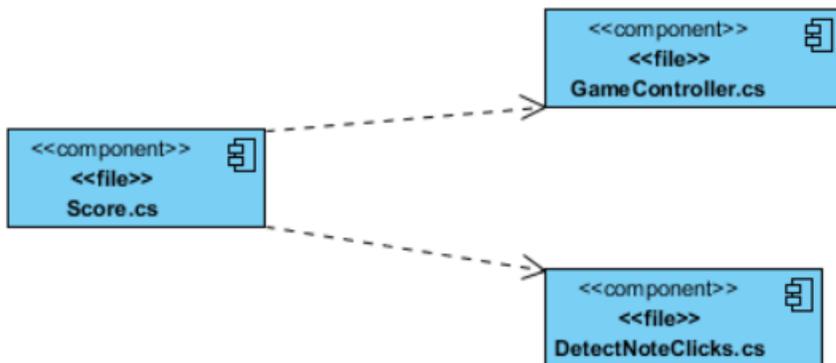
A 3.3 Diagrama de componentes de mecanismo M3 Generación de notas.



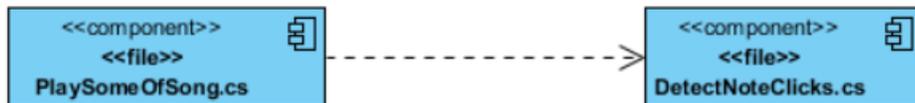
A 3.4 Diagrama de componentes de mecanismo M4 Nota.



A 3.5 Diagrama de componentes de mecanismo M5 Puntuación.



A 3.6 Diagrama de componentes de mecanismo M6 Efecto de sonido.



A 3.7 Diagrama de componentes de mecanismo M7 Interacción con las notas.



#### A 4 No conformidades registradas durante las pruebas Beta

- Iteración 1:
  - NC 1 Error Ortográfico, la palabra comenzar está escrita con s.
    - Nombre del probador: Elier Evian Miranda Abreu.
    - Localización del error: menú principal.
    - Tipo de error: app-Doc/ortografía.
    - Impacto: medio.
    - Estado del defecto: resuelto.
    - Respuesta del equipo: resuelto.
  - NC 2 Error ortográfico, la palabra jugar le falta la r.
    - Nombre del probador: Josué Hernández Martínez.
    - Localización del error: menú de fin de partida
    - Tipo de error: app-Doc/ortografía.
    - Impacto: medio.
    - Estado del defecto: resuelto.
    - Respuesta del equipo: resuelto.
  - NC 3 Cuando se vuelve a jugar la puntuación no se reinicia.
    - Nombre del probador: Leandro Ruiz Domínguez.
    - Localización del error: pantalla de juego.
    - Tipo de error: app-funcionalidad.
    - Impacto: alto.
    - Estado del defecto: resuelto.
    - Respuesta del equipo: resuelto.
  - NC 4 Algunas notas no suman puntuación.
    - Nombre del probador: Johan Chirino Montano.
    - Localización del error: pantalla de juego.
    - Tipo de error: app-funcionalidad.
    - Impacto: alto.
    - Estado del defecto: resuelto.
    - Respuesta del equipo: resuelto.

- NC 5 Las notas no tienen el mismo tamaño.
  - Nombre del probador: David Ricardo Delgado Perez.
  - Localización del error: pantalla de juego.
  - Tipo de error: app-funcionalidad.
  - Impacto: alto.
  - Estado del defecto: resuelto.
  - Respuesta del equipo: resuelto.
- Iteración 2:
  - NC 1 El fondo no se ajusta a diferentes resoluciones.
    - Nombre del probador: Elier Evian Miranda Abreu.
    - Localización del error: menú principal.
    - Tipo de error: app-funcionalidad.
    - Impacto: alto.
    - Estado del defecto: resuelto.
    - Respuesta del equipo: resuelto.
  - NC 2 Permite tocar partes de la pantalla que no sean notas sin perder.
    - Nombre del probador: David Ricardo Delgado Perez.
    - Localización del error: durante la partida.
    - Tipo de error: app-funcionalidad.
    - Impacto: alto.
    - Estado del defecto: resuelto.
    - Respuesta del equipo: resuelto.
  - NC 3 La puntuación del jugador se oculta bajo las notas.
    - Nombre del probador: Josué Hernández Martínez.
    - Localización del error: Durante la partida.
    - Tipo de error: app-funcionalidad.
    - Impacto: alto.
    - Estado del defecto: resuelto.
    - Respuesta del equipo: resuelto.