FACULTAD 4

Desarrollo de los componentes Sabios del Son y Afinación de Instrumentos para el videojuego Isla del Son

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

Autor:

Jose Carlos León Martínez

Tutores:

Ing. Jorge Evelio Valdivia Hernández

M.Sc. PA Yanetsi Millet Lombida

La Habana, noviembre de 2023

Año 65 de la Revolución

DECLARACION DE AUTORÍA

El autor del trabajo de diploma con título "**Desarrollo de los componentes Sabios del Son y Afinación de Instrumentos para el videojuego Isla del 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 29 días del mes de noviembre del año 2023.

Jose Carlos León Martínez

Autor

Ing. Jorge Evelio Valdivia Hernández

Tutor

M.Sc. PA Yanetsi Millet Lombida

Tutor

AGRADECIMIENTOS

Quiero agradecer a toda mi familia por apoyarme a pesar de todas las adversidades.

A mis compañeros con los cuales compartí buenos y malos momentos a partes iguales.

A mis tutores Evelio y Yanetsi por su constante ayuda en el desarrollo de la investigación.

A la profesora Enelis por sus aclaraciones en las etapas tempranas de la realización del documento.

Por último, gracias a mi madre, sin ella nada de esto hubiera sido posible.

RESUMEN

El son es un género musical tradicional de Cuba que ha perdido popularidad en las nuevas generaciones debido a influencias culturales externas y avances tecnológicos. Con el objetivo de preservar y promover este importante fragmento de la cultura cubana, se ha propuesto la creación del videojuego "Isla del Son". Sin embargo, el juego carece de mecanismos para brindar información y permitir a los jugadores familiarizarse con los ritmos y notas musicales característicos del son. Esta investigación tiene como objetivo desarrollar los componentes interactivos necesarios para enriquecer el nivel cultural de los jugadores y promover su conocimiento y familiaridad con los subgéneros del son. Se utilizará el estilo de juego trivia y memoria a corto plazo como juegos denominados juegos serios para lograr este propósito. Para dicho propósito se utilizó como motor gráfico Unity, en el que se ensamblarán todas las partes y componentes necesarios para crear el videojuego. Para mantener una organización adecuada se utilizó como metodología el Marco de desarrollo ingenieril para el proceso de desarrollo de videojuegos y se seleccionó la arquitectura definida para trabajar en videojuegos usando Unity al ser la más adecuada según las necesidades y características de la solución. Se validó la solución a partir de la aplicación de pruebas Alfas y Betas.

PALABRAS CLAVE

Son, Trivia, Unity, Videojuego, Videojuegos de memoria a corto plazo, Videojuegos serios

ABSTRACT

Son is a traditional Cuban musical genre that has lost popularity in new generations due to external cultural influences and technological advances. With the objective of preserving and promoting this important fragment of Cuban culture, the creation of the video game "Isla del Son" has been proposed. However, the game lacks mechanisms to provide information and to allow players to become familiar with the characteristic rhythms and musical notes of the son. This research aims to develop the interactive components necessary to enrich the cultural level of the players and promote their knowledge and familiarity with the son subgenres. Trivia and short-term memory style of play will be used as games called serious games to achieve this purpose. For this purpose, Unity was used as the graphics engine, in which all the parts and components necessary to create the video game will be assembled. To maintain an adequate organization, the Engineering Development Framework for the video game development process was used as a methodology and the architecture defined to work on video games using Unity was selected as it was the most appropriate according to the needs and characteristics of the solution. The solution was validated through the application of Alpha and Beta tests.

KEYWORDS

Serious video games, Short-term memory video games, Son, Trivia, Unity, Video game

ÍNDICE

INTRODUCCIÓN ..... - 1 -

Posibles resultados: ..... - 2 -

Tareas de la investigación: ..... - 2 -

Métodos teóricos ..... - 3 -

Métodos empíricos ..... - 3 -

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA ..... - 5 -

1.1 Videojuego ..... - 5 -

1.2 Juego Serio ..... - 7 -

1.2.1 Videojuego Trivia ..... - 7 -

1.2.2 Videojuegos de habilidad de memoria a corto plazo ..... - 7 -

1.3 Sistemas Homólogos ..... - 8 -

1.3.1 La neurona intranquila ..... - 8 -

1.3.2 Who Wants to Be a Millionaire ..... - 9 -

1.3.3 Touch me ..... - 11 -

1.3.4 Simon ..... - 11 -

1.5 Metodología de desarrollo de software ..... - 12 -

1.6 Herramientas y tecnologías ..... - 14 -

1.6.1 Herramienta de modelado ..... - 14 -

1.6.2 Motor de videojuegos Unity ..... - 14 -

1.6.3 Editor de texto VS Code ..... - 15 -

1.6.4 Lenguaje de programación ..... - 15 -

1.6.5 Lenguaje de modelado ..... - 15 -

Conclusiones parciales ..... - 16 -

CAPÍTULO II: CONCEPTUALIZACIÓN Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN . - 17 -

2.1 Conceptualización de la propuesta de solución ..... - 17 -

2.2 Elementos Formales ..... - 19 -

Objetivo ..... - 20 -

Sabios del Son ..... - 22 -

Afinación de instrumentos ..... - 22 -
Procedimientos ..... - 22 -
Reglas ..... - 22 -
Recursos ..... - 24 -
Conflictos ..... - 25 -
Frontera o Límite ..... - 25 -
Resultado ..... - 25 -
2.3 Especificación de mecanismos ..... - 25 -
2.3.1 Paquetes de mecanismos ..... - 30 -
2.4 Arquitectura de software ..... - 32 -
2.5 Patrones de diseño ..... - 36 -
2.6 Diagrama de Clases ..... - 37 -
2.7 Representación del comportamiento ..... - 39 -
2.7 Descripción de requisitos no funcionales ..... - 41 -
CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA 43
3.1 Estándar de codificación 43
3.1.1 Definición de clases: 43
3.1.2 Declaración de variables: 43
3.1.3 Declaración de métodos: 44
3.2 Integración con el juego macro 44
3.3 Diagramas de componentes 45
3.4 Pruebas 46
CONCLUSIONES DEL CAPÍTULO 55
CONCLUSIONES ..... - 56 -
RECOMENDACIONES ..... - 57 -
REFERENCIAS BIBLIOGRÁFICAS ..... - 58 -

INTRODUCCIÓN

El son es un género musical originario de Cuba que combina parte estructural de la música tradicional española con instrumentos y elementos característicos de la música indígena y afroamericana. El son en su estructura surge en Guantánamo, Baracoa, Manzanillo y Santiago de Cuba a finales del Siglo XIX. En 1892, el tresista de origen haitiano Nené Manfugás lo lleva del monte a los carnavales de Santiago de Cuba. Con el establecimiento en la década de 1920 de la radiodifusión comercial, comenzó el auge y popularización del son, siendo el Septeto Nacional de Ignacio Piñeiro, uno de los principales representantes de esta época. Ignacio Piñeiro es el creador de Échale salsita, probablemente la primera vez que se usó la palabra salsa para denominar a la música caribeña (Fernández, 2001). En septiembre de 2012 fue declarado patrimonio cultural de Cuba.

A pesar de que es uno de los géneros más influyentes de la música latinoamericana, y viva representación del folklore y la cultura cubana, el interés de las nuevas generaciones por dicho género se ha visto en declive en los últimos años. Esto 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 Isla del 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 Isla del Son mecanismos para obtener conocimientos e identificar los ritmos musicales de los subgéneros del son?

Como **objeto de estudio** se define: los juegos serios.

Para dar solución al problema de investigación planteado, se define como **objetivo general**: Desarrollar dos componentes en forma de minijuegos para el videojuego Isla del Son pertenecientes al género agilidad mental que enriquezcan el nivel cultural de los jugadores con respecto a su conocimiento y familiaridad con el son.

Dentro del objeto de estudio se precisa como **campo de acción**: videojuegos de agilidad mental.

Posibles resultados:

Componentes en forma de minijuego para el videojuego Isla del Son que desarrollen de forma interactiva el conocimiento sobre el género musical del son y permita reconocer las diferentes notas musicales de este género.

Tareas de la investigación:

- ✓ Elaboración del marco teórico de la investigación a través del estudio del estado del arte.
- ✓ Caracterización del desarrollo de videojuegos pertenecientes al género agilidad mental.
- ✓ Diseño arquitectónico de la propuesta de solución.
- ✓ Implementación de los minijuegos como componentes propuestos.
- ✓ Validación a través de pruebas a los minijuegos como componentes implementados.

Como métodos de investigación científica se emplearon los siguientes:

Métodos teóricos

Histórico y Lógico: se emplea para la fundamentación y sistematización de los aspectos teóricos observados en el desarrollo de la investigación acerca del son como género musical, trivia y demás elementos vinculados a la temática del trabajo. También se utiliza para la comprensión de los antecedentes y su evolución como base para la investigación acerca del desarrollo de videojuegos educativos con forma de quiz.

Analítico-Sintético: durante el proceso de investigación permite descubrir los elementos esenciales y conceptos en torno al objeto de investigación, que en este caso fueron los juegos serios

Modelación: se emplea para el diseño y mejor comprensión de la aplicación, teniendo en cuenta los elementos teóricos y prácticos del desarrollo de videojuegos de agilidad mental mediante la elaboración de diagramas y modelos.

Métodos empíricos

Consulta de la información en todo tipo de fuente: se utiliza para la elaboración del marco teórico de la investigación acerca de los juegos serios para su uso como alusión para las propuestas realizadas.

Observación: se emplea para poder establecer comparaciones e identificar similitudes con otras aplicaciones existentes que utilicen los videojuegos de agilidad mental como base interactiva para extraer las características principales que puedan ser integradas a la solución propuesta.

El presente documento está compuesto por tres capítulos, en los que se relacionan todo lo referente a la investigación, 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 propuesta de solución. En este capítulo se describen las particularidades técnicas y los artefactos generados para el desarrollo de la solución, como son: 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 definen los conceptos que sustentan el dominio de la investigación, los videojuegos y los juegos serios. Se analiza el juego base Isla del Son para el cual se realiza la propuesta. Se hace énfasis en los géneros trivia y juegos de memoria a corto plazo. También se analizan algunos ejemplos de sistemas homólogos y las características que los definen. Se describen, además, las principales tecnologías y herramientas que pueden ser utilizadas para implementar la solución que se pretende desarrollar, así como la metodología de desarrollo de software a utilizar.

1.1 Videojuego

Son una forma de entretenimiento interactivo que se juega en dispositivos electrónicos, como consolas de videojuegos, computadoras, teléfonos móviles y tabletas. Los videojuegos pueden presentarse en diferentes géneros, como acción, aventura, simulación, estrategia, deportes, entre otros. En los videojuegos, los jugadores asumen el papel de personajes y participan en diferentes desafíos, ya sea en solitario o en modo multijugador. Los videojuegos también pueden incluir elementos de narrativa, música, gráficos y efectos especiales para crear una experiencia inmersiva y atractiva para los jugadores (Tavinor, 2008).

1.1.1 Clasificación y géneros de los videojuegos

Los videojuegos se pueden clasificar en géneros atendiendo a factores como el sistema de juego, el tipo de interactividad con el jugador o sus objetivos. Entre los géneros más comunes se encuentran los de aventura, acción, estrategia, agilidad mental, deportes, rol o simulación (Belli & López Raventós, 2013):

• **Acción**: En este tipo de juegos las acciones básicas son mover el personaje y usar un arma. La mecánica del juego impone generalmente al jugador tener buenos reflejos y precisión. Algunos ejemplos son: DOOM, Half Life, Quake.

• **Deporte**: Los juegos de deporte son aquellos que simulan juegos de deporte real, entre ellos se encuentran: Golf, Tony Hawks, FIFA.

• **Aventura**: Videojuegos orientados al descubrimiento de una trama narrativa con tendencia a la jugabilidad rígida. Se encuentra dentro de este género el juego Colossal Cave Adventure.

• **Rol**: Videojuegos orientados a la reconstrucción de una narrativa con jugabilidad abierta. Se caracteriza por la adopción un rol y la búsqueda de puntos de experiencia, que se obtienen por diversas acciones meritorias en el juego y que sirven para mejorar progresivamente las habilidades del personaje. Presentan una estructura de misiones diseminadas en el juego, entre las que el jugador tiene un cierto margen de libertad en su elección y en el orden de abordarlas. Algunos videojuegos del tipo de rol son: The Elder Scrolls IV: Oblivion, Dark Souls.

• **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.

• **Simulación**: Pretenden emular ante los jugadores vivencias y situaciones lo más realistas posible. La diferencia entre un juego de simulación y otro radica en el nivel de detalle del entorno y el control que puedan ejercer sobre el mismo. De este tipo de videojuego se destacan títulos como Eurotruck.

• **Estrategia**: Requieren que el jugador ponga en práctica sus habilidades de planeamiento y pensamiento para maniobrar, gestionando recursos de diversos tipos como materiales, humanos o militares para conseguir la victoria. Dentro de este género se encuentran: Star Craft, Age of Empires.

Dentro del ámbito de los videojuegos hay un término conocido como juegos serios que hace referencia a juegos que no solo están destinados a entretener si no que poseen fines adicionales como la educación. Esta investigación se centrará en este tipo de juegos en concreto ya que sus características se alinean con el objetivo propuesto con la creación del juego base Isla del Son.

1.2 Juego Serio

Juegos de computadora que están destinados no solo a entretener a los usuarios, sino también a fines adicionales como la educación y la formación. Pueden ser similares a juegos educativos, pero se centran principalmente en una audiencia fuera de la educación primaria o secundaria. Los juegos serios pueden ser de cualquier género y muchos de ellos pueden ser considerado una especie de entretenimiento educativo, pero el objetivo principal de un juego serio no es entretener, aunque el potencial de los juegos para atraer es a menudo un aspecto importante de la elección de utilizar juegos como herramienta de enseñanza. Un juego serio suele ser una simulación, que tiene la apariencia de un juego, pero en realidad es una simulación del mundo real eventos o procesos. El objetivo principal de un juego serio suele ser entrenar o educar usuarios, aunque puede tener otros fines, como marketing o publicidad, mientras les brinda una experiencia agradable (Susi et al., 2007).

Los videojuegos pertenecientes al género de agilidad mental son especialmente compatibles con esta filosofía a la hora de abordar los videojuegos ya que incentivan al jugador a ejercitar su memoria o poner a prueba sus conocimientos. Dos subgéneros que cumplen con estas características son los conocidos como videojuegos de *short-term memory skill* y Trivia.

1.2.1 Videojuego Trivia

Se conoce popularmente como trivia aquel juego generalmente con modalidad de concurso, en el que se enfrentan varios participantes y que consiste en la formulación de preguntas respecto de puntuales temas de interés como son: las artes, las ciencias, la historia, el deporte, entre otros, a las cuales se les agregarán una serie de opciones como respuesta, en tanto, los concursantes deberán escoger entre una de ellas a la cual consideran es la correcta (Monsalve Vidales et al., 2016).

1.2.2 Videojuegos de habilidad de memoria a corto plazo

La **memoria a corto plazo** (MCP), también conocida como "memoria primaria" o "memoria activa", es la capacidad para mantener en mente de forma activa una pequeña cantidad de información, de forma que se encuentre inmediatamente disponible durante un corto periodo de tiempo (RUÈZ-VARGAS, 1994). Los videojuegos que se centran en ejercitar esta parte de

la mente mediante la repetición de patrones son llamados *short-term memory skill games*, estos suelen aparecer principalmente como minijuegos o desafíos en otros juegos. Uno de los más conocidos a lo largo del mundo es el llamado *Simon game* (Guins, 2014).

1.2.3 Isla del Son

En el documento de diseño del videojuego Isla del 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.

1.3 Sistemas Homólogos

Existen una gran variedad de videojuegos que se han desarrollado con el propósito de no solo entretener al jugador, sino además aportarle conocimientos y mejorar su capacidad mental. En el caso específico de los dos géneros que se abordan en esta investigación hay algunos ejemplos relevantes que fueron tomados a consideración.

1.3.1 La neurona intranquila

El videojuego La Neurona se basa en el programa de televisión "El selecto club de la Neurona Intranquila". Es un videojuego de conocimientos y agilidad mental, con las secciones y

preguntas del programa, por lo que va dirigido al público mayor de 12 años. En la pantalla principal se reflejan los dos hemisferios de un cerebro, en el cual aparece una barra que muestra el progreso. Eso permite ver en qué modalidad se es mejor. El videojuego busca estimular la agilidad, el conocimiento y el sentido de competencia que son esencias del programa televisivo (*COSMOX | Videojuego La Neurona 1*, s. f.). En la sección de trivia de dicho juego se le presentan varias respuestas posibles a una pregunta en concreto y se le da un límite de 30 segundos al jugador para responder.

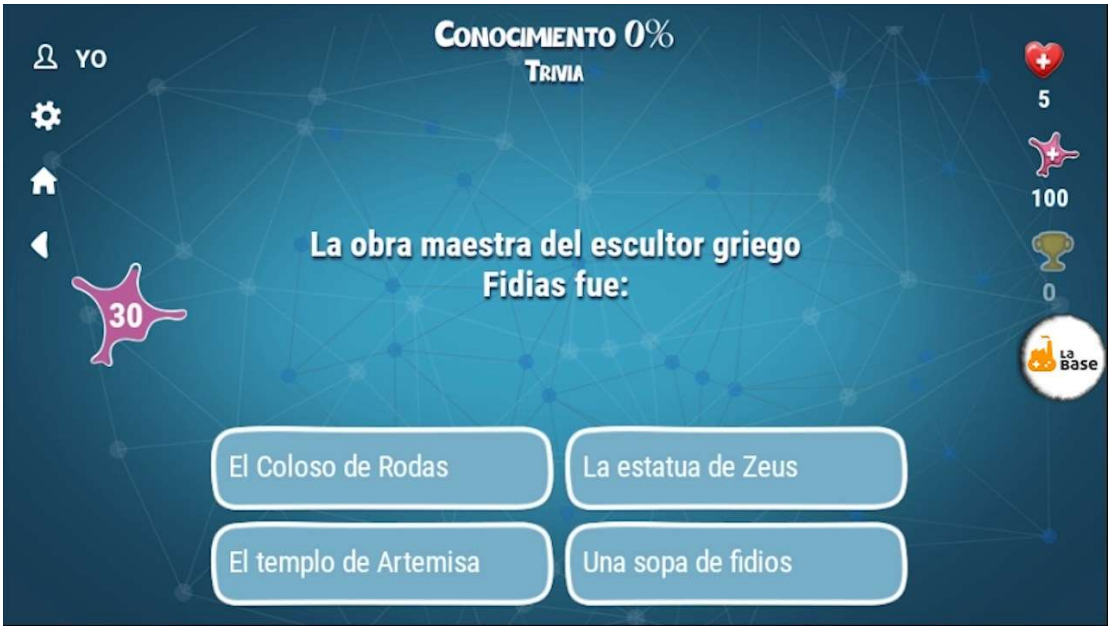


Fig1.La neurona intranquila. (https://www.videojuego.cu/la-neurona-2?page=1).

1.3.2 Who Wants to Be a Millionaire

Who Wants to Be a Millionaire es un videojuego trivia de 1999 desarrollado originalmente por Jellyvision y publicado por Disney Interactive, basado en la franquicia de televisión del mismo nombre. El juego se basó originalmente en la versión estadounidense del programa. Le asigna al jugador la tarea de responder preguntas de prueba en un período de tiempo limitado. Estás va aumentando la dificultad con cada pregunta correctamente respondida y aumenta la cantidad de dinero (moneda del juego) que puede ganar el jugador. (*"Who Wants to Be a Millionaire?" – play the free online game*, s. f.)

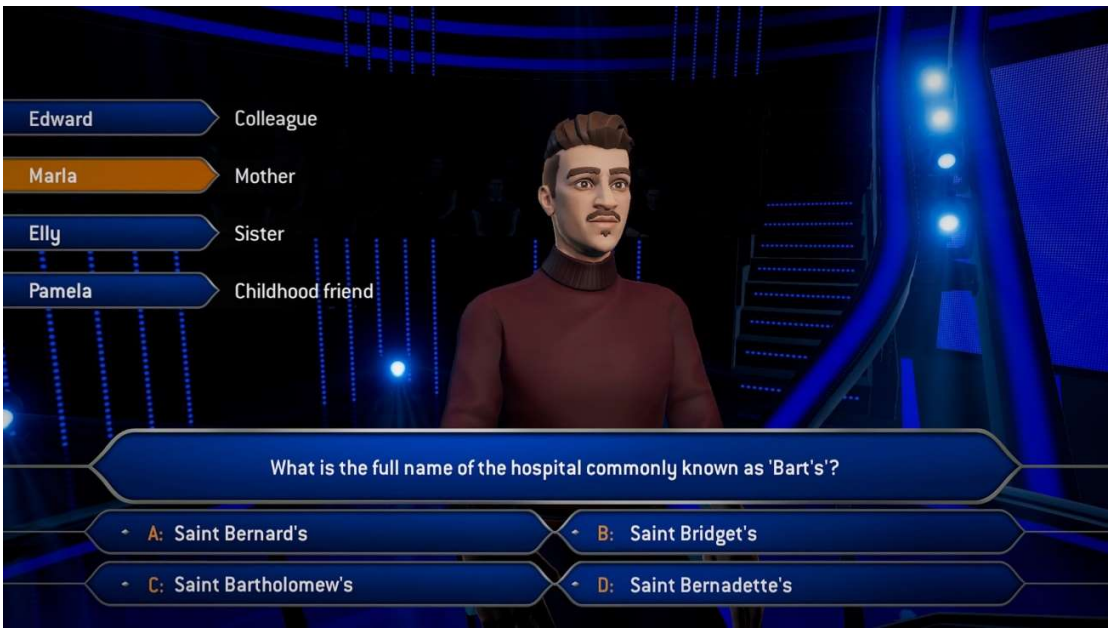


Fig2. Who Wants to Be a Millionaire.
(https://store.steampowered.com/app/1356240/Who_Wants_To_Be_A_Millionaire/
).

En estos ejemplos se puede apreciar que algunas de las características invariables de este tipo de juego son:

- darle al jugador un tiempo límite para responder.
- usar varias alternativas como respuesta que se asemejen mucho a la correcta para añadir cierto nivel de dificultad.
- el uso de potenciadores o comodines para darle la oportunidad al jugador de avanzar en una pregunta que se le haga muy compleja.

En el caso de los sistemas homólogos analizados ninguno posee una temática especifica en la que centrar las preguntas lo que no permite la profundización en un área del conocimiento específico, también carecen de un indicador visual que permita relacionar la información que está preguntando con la apariencia del juego lo que permitiría un mayor impacto en el jugador por lo que en este sentido la propuesta tomará un acercamiento que esté de acorde a los requisitos del juego base. Por otro lado, con respecto a las características generales de estos

sistemas, se llegó a la conclusión que son fundamentales para la jugabilidad de este tipo de juegos, por lo que serán tomadas como referencia en el desarrollo de la propuesta.

1.3.3 Touch me

Touch Me es un juego de arcade lanzado por primera vez por Atari Inc. en 1974 y luego como un juego portátil en 1978. Se puede describir como un juego al estilo *"Simon says"* que consiste en tocar una serie de botones que se iluminan y producen sonidos. El jugador debe observar una secuencia de luces eléctricas parpadeantes y repetir la secuencia en el mismo orden en que ocurrió. Cada vez que esto se complete, el juego producirá otra secuencia con un botón adicional agregado. Este proceso se repite y una ventana de puntuación digital muestra el número total de secuencias de sonido que un jugador repite correctamente. El juego continúa hasta que se alcanza la secuencia máxima de botones o el usuario comete un error (*Touch-Me - Arcade by Atari*, s. f.).

Fig3. Versión portátil de Touch me (https://en.wikipedia.org/wiki/Touch_Me_(arcade_game)).

1.3.4 Simon

Es un juego electrónico de habilidad de memoria a corto plazo inventado por Ralph H. Baer y Howard J. Morrison, que trabajan para la firma de diseño de juguetes Marvin Glass and Associates. En 1977, Ralph Baer vio potencial en el concepto "Simon Says" detrás del juego

Touch Me. Copió el juego de Atari, agregó botones de colores y efectos de sonido musicales, y creó el juego portátil Simon, que se convirtió en un gran éxito (Guins, 2014). El dispositivo crea una serie de tonos y luces y requiere que el usuario repita la secuencia. Si el usuario tiene éxito, la serie se vuelve progresivamente más larga y compleja. Una vez que el usuario falla o se agota el límite de tiempo, el juego termina.

Fig4. Dispositivo para jugar *Simon*(https://es.wikipedia.org/wiki/Simon_(juego)).

Este tipo de mecánicas tiene un gran potencial para la familiarización y memorización de notas musicales por lo que resulta una buena referencia a tener en cuenta en el desarrollo de esta investigación. En el caso de este tipo de juego no hay un videojuego claro que se base solo en estas mecánicas si no que suelen ser utilizadas como complemento a la jugabilidad en forma de minijuegos o puzles como parte de la experiencia completa.

1.4 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 (Hernández et al., 2017). Se compone por cinco etapas que complementan el proceso de desarrollo de los videojuegos: Conceptualización, Diseño, Implementación, Prueba y Mantenimiento. A continuación, se define la estructura de la metodología:

• Etapa 1: Conceptualización

- ❖ Definir el género sobre el cual se desarrollará el videojuego.
- ❖ Describir la mecánica del videojuego.
- ❖ Especificar las metas para la experiencia del jugador.

• Etapa 2: Diseño

- ❖ Describir los elementos formales que definen la estructura del videojuego.
- ❖ Describir los elementos dramáticos que definen el entretenimiento del videojuego.
- ❖ Diseñar las pantallas gráficas elementales que forman la estructura del videojuego.
- ❖ Describir los elementos dinámicos que definen las mecánicas o mecanismos del videojuego.
- ❖ Validar los mecanismos teniendo en cuenta criterios técnicos para su implementación.
- ❖ Modelar el diagrama de paquetes de mecanismos teniendo en cuenta la distribución arquitectónica.
- ❖ Describir la concepción de los mecanismos sobre la distribución arquitectónica diseñada.
- ❖ Modelar el comportamiento de los mecanismos mediante diagramas de transición de estado.
- ❖ Mantener una trazabilidad bidireccional entre cada elemento del videojuego.
- ❖ Describir las características no funcionales del videojuego.

• Etapa 3: Implementación

- ❖ Diseñar los componentes que encapsulan la implementación.
- ❖ Desarrollar las mecánicas especificadas y diseñadas.

• Etapa 4: Prueba o Desarrollar pruebas Alpha.

- ❖ Desarrollar pruebas Beta.
- ❖ Registrar defectos durante las pruebas realizadas.

• Etapa 5: Mantenimiento

- ❖ Realizar análisis Postmortem.
- ❖ Retomar la etapa de Diseño

1.5 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 Isla del Son.

1.5.1 Herramienta de modelado

Visual Paradigm para UML (por sus siglas en inglés, *Unified Modeling Lenguage*) en su versión 5.0, es una herramienta CASE que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software (Paradigm, 2013).

1.5.2 Motor de videojuegos Unity

Para la elaboración de la solución se propone como herramienta, la utilización de Unity (específicamente en su versión 2021.3.4f1). Unity es un motor de videojuego multiplataforma creado por Unity Technologies. Está disponible como plataforma de desarrollo para Microsoft Windows, OS X, Linux. La plataforma de desarrollo tiene soporte de compilación con diferentes tipos de plataformas. Unity tiene dos versiones: Unity Professional (pro) y Unity Personal. Unity puede usarse junto con Blender, 3ds Max, Maya, Softimage, Modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks y Allegorithmic Substance. Los cambios realizados a los objetos creados con estos productos se actualizan automáticamente en todas las instancias de ese objeto durante todo el proyecto sin necesidad de volver a importar manualmente (*Unity Documentation*, s. f.).

El motor gráfico utiliza OpenGL (en Windows, Mac y Linux), Direct3D (solo en Windows) y OpenGL ES (en Android y iOS. Tiene soporte para mapeado de relieve, mapeado de reflejos, mapeado por paralaje, oclusión ambiental en espacio de pantalla, sombras dinámicas

utilizando mapas de sombras, render a textura y efectos de post-procesamiento de pantalla completa. Además, permite el uso de servicios de Android como el uso del GPS. Como es una herramienta de uso recurrente en el centro, se tiene experiencia con la misma, así como soporte, por lo que se decide su uso (*Unity Documentation*, s. f.).

1.5.3 Editor de texto VS Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y Mac, es gratuito y de código abierto. Incluye soporte para depuración, control de Git integrado, resaltado de sintaxis, finalización de código inteligente, fragmentos de código y refactorización de código. También es personalizable, de modo que los usuarios pueden cambiar el tema del editor, los métodos abreviados de teclado y las preferencias [26]. Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta como una aplicación de escritorio. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C++, C#, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity) (*Documentation for Visual Studio Code*, s. f.).

1.5.4 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).

1.5.5 Lenguaje de modelado

El lenguaje unificado de modelado es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group (OMG). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un “plano” del sistema (modelo), incluyendo

aspectos conceptuales entre los que destacan procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados. Es importante remarcar que UML es un “lenguaje de modelado” para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo (Paradigm, 2013).

Conclusiones parciales

Analizadas las necesidades del videojuego Isla del Son se determinó que es necesario el desarrollo de dos minijuegos usando las características inherentes a los juegos serios, y que pertenezcan a los géneros trivia y *short-term memory skill.* Uno que permita al jugador obtener y mejorar sus conocimientos acerca del género musical son y otro que permita al jugador familiarizarse y memorizar los sonidos y acordes típicos de este género. El empleo del Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos permitió enfocar los artefactos necesarios a utilizar, proporcionando así una mayor descripción y organización de los requerimientos del software, reduciendo la curva de aprendizaje y el tiempo de desarrollo.

CAPÍTULO II: CONCEPTUALIZACIÓN Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

En el siguiente capítulo se da a conocer la descripción 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 en las etapas de conceptualización y diseño. Se realiza la definición de los elementos formales y dramáticos, la especificación de los mecanismos, la descripción de los patrones de diseño aplicados, la representación de los diagramas de clases del diseño y la arquitectura utilizada.

2.1 Conceptualización de la propuesta de solución

Sabios del Son

Se propone como solución dos componentes que actúan como minijuegos del videojuego Isla del Son. Uno de los componentes, Sabios del Son, es un trivia en el cual la interfaz muestra un temporizador descendente que detiene la partida al llegar a cero contando como un estado de derrota para el jugador, también se puede apreciar la cantidad de preguntas intentadas y la cantidad de puntos obtenidos por el jugador en la partida, por último, el botón de pausa permite acceder a un menú que da acceso a las opciones de resumir, reiniciar la partida o regresar al menú principal del juego. Dicha interfaz cambia su apariencia para hacer referencias a lugares y objetos típicamente asociados en la cultura a la provincia correspondiente a la casa del son en la que se encuentra el jugador.

Jugabilidad de Sabios del Son

El juego cuenta con una serie de preguntas que varían según la casa del son en la que se encuentre el jugador y un conjunto de cuatro respuestas asociadas a cada pregunta, siendo una de ellas la correcta y las restantes respuestas incorrectas que cambian su contenido según el nivel de dificultad en el que se encuentre el jugador, el jugador también tiene acceso a tres comodines que permiten al jugador usar alguna de sus funciones únicas una vez por partida para obtener algún tipo de pista o ayuda para lograr superar alguna pregunta que le resulte demasiado complicada.

Metas para la experiencia del jugador:

- El jugador obtendrá nuevos conocimientos acerca del son.
- El jugador podrá comprobar sus propios conocimientos acerca del son.
- El jugador querrá mejorar su puntuación en cada partida.
- El jugador estará incentivado a aumentar su posición en el ranking.
- El jugador tendrá un mayor interés en el género musical del son.

Afinación de Instrumentos

El otro componente llamado Afinación de Instrumentos es un juego de *short-term memory skill* con un estilo similar al juego *Simon Says,* en el que se le muestra al jugador una rueda de percusión con cuatro botones de diferentes colores y a cada uno asociado un sonido específico. Un indicador de combos que muestra los objetivos, la cantidad de combos a obtener y las obtenidas. Indicador de multiplicador el cual consiste en que una vez presionados los paneles aparece la cantidad de claves de son obtenidas en la parte superior de la pantalla. El tamaño de los números crece en la medida que la combinación es más larga, escala de tamaño hasta un máximo de X12. Indicador de total de claves obtenidas las cuales representan la puntuación total del jugador en el ranking del minijuego. Botón de pausa que detiene la partida, abre una ventana emergente con los botones reiniciar partida, continuar y salir. También se muestran los tres potenciadores a escoger. Estos se recargan cada 24 horas. Salir abre la pantalla del mapa. Estos son los potenciadores:

Potenciadores:

- Repite una vez más la demostración al usuario.
- Reduce la dificultad de la combinación de notas -4.
- Completa una combinación automáticamente.

Jugabilidad de Afinación de Instrumentos

Al empezar la partida el juego reproduce una combinación de notas. El jugador debe reproducir la combinación exactamente como la escucha presionando en las secciones que corresponden. Gana la partida cuando reproduce todas las combinaciones sin equivocarse.

Pierde la partida si equivoca el orden al presionar y debe empezar del principio. Cada partida tiene mínimo 4 combinaciones por turno, estas incrementan una nota en la medida que el jugador las reproduce sin equivocarse. Puede llegar a un máximo de 12 notas a memorizar. Los niveles escalan en dificultad añadiendo más y más notas a memorizar.

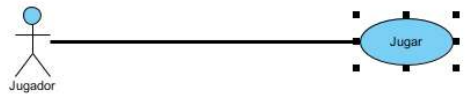
Metas para la experiencia del jugador:

- El jugador se familiarizará con las notas musicales características del son.
- El jugador querrá mejorar su puntuación en cada partida.
- El jugador estará incentivado a aumentar su posición en el ranking.
- El jugador tendrá un mayor interés en el género musical del son.

2.2 Elementos Formales

Los elementos formales definen las características, objetos y comportamientos de los juegos.

Aspecto	Descripción
Invitación a Jugar (ambos juegos)	Botón “Iniciar”
Cantidad de Jugadores (ambos juegos)	Un jugador
Roles	Sabios del Son: Existe una sola modalidad de juego. En esta, el jugador debe responder una serie de preguntas asociadas a la casa del llamo en la que se encuentre. Afinación de instrumentos: Existe una sola modalidad de juego. En esta, el jugador debe seleccionar en el orden correcto los botones con las notas que se le mostraron ante-

	riormente.
Patrón de Interacción (ambos juegos)	

Objetivo

Sabios del Son

Objetivo	Descripción
El objetivo es responder las preguntas del quiz.	El jugador tiene como objetivo responder la mayor cantidad de preguntas correctamente y así obtener una alta puntuación en el ranking.
Tipo	Explícitos.
Categorías	• Agilidad mental • Memoria • Solución

Afinación de instrumentos

Objetivo	Descripción
El objetivo es tocar en orden las notas	El jugador tiene como objetivo tocar en el orden correcto las notas en el círculo de percusión para aumentar su puntuación.
Tipo	Explícitos.

Categorías	• Agilidad mental • Memoria • Solución

	Sabios del Son	**Afinación de instrumentos**
Procedimientos	El jugador puede seleccionar el mini-juego en la casa del llamo en la que se encuentre. El jugador puede responder preguntas seleccionando una respuesta. El jugador puede seleccionar entre 4 opciones para responder dicha pregunta. El jugador puede usar uno de los 3 comodines a su disposición en caso de que necesite ayuda con una pregunta. El jugador puede pausar el juego. El jugador puede salir al menú principal del juego. El jugador puede responder optar por responder hasta un máximo de 8 preguntas.	El jugador puede seleccionar el mini-juego en la casa del llamo en la que se encuentre. El jugador puede seleccionar uno de los botones relacionados con una nota. El jugador puede introducir una secuencia usando los botones de la rueda de percusión El jugador puede usar uno de los 3 potenciadores a su disposición. El jugador puede pausar el juego. El jugador puede salir al menú principal del juego.
Reglas	Al finalizar el tiempo límite se pasará automáticamente a la próxima pregunta. Solo se puede usar cada comodín una	Al jugador se le muestra una rueda con distintas secciones que representan distintos sonidos. Cada sección tiene un color característico.

	vez por ronda. Al responder una pregunta pasará a la siguiente. Se suma un punto al contador de puntos en caso de que haya respondido correctamente. Al responder incorrectamente termina la partida. Al responder correctamente se multiplica por 2 la puntuación actual. El jugador completa el nivel al responder un mínimo de 5 preguntas correctamente.	Al empezar la partida el juego reproduce una combinación de notas. El jugador debe reproducir la combinación exactamente como la escucha presionando en las secciones que corresponden. Gana la partida cuando reproduce todas las combinaciones sin equivocarse. Pierde la partida si equivoca el orden al presionar y debe empezar del principio. Cada partida tiene mínimo 4 combinaciones por turno, estas incrementan una nota en la medida que el jugador las reproduce sin equivocarse. Puede llegar a un máximo de 12 notas a memorizar. Solo se puede escoger un potenciador por intento Los niveles escalan en dificultad añadiendo más y más notas a memorizar.

Recursos	Contador: es el reloj que cuenta regresivamente los 30 segundos de los cuales dispone el jugador para responder la pregunta que le corresponde. Botones de respuesta: botones con opciones de respuesta. Comodines: Apicuba (reduce a la mitad las respuestas a escoger). Cuba Ron (señala la respuesta correcta). EGREM (cambia la pregunta por otra). Pausa: detiene el juego y te da la opción de volver al menú. Puntuación: muestra los puntos obtenidos por el jugador	Indicador de combos: muestra los objetivos, la cantidad de combos a obtener y las obtenidas. Indicador de multiplicador: una vez presionados los paneles aparece la cantidad de claves de son obtenidas en la parte superior de la pantalla. El tamaño de los números crece en la medida que la combinación es más larga. (Escala de tamaño hasta un máximo de X12) Indicador de total de claves obtenidas. Panel de percusión: círculo dividido en 4 partes de distintos colores. Estos espacios que el usuario debe presionar. Botón de pausa: detiene la partida. Abre una ventana emergente. Con los botones reiniciar partida, continuar y salir. Muestra los tres potenciadores a escoger. Estos se recargan cada 24 horas. Solo se puede escoger uno por intento. Salir abre la pantalla del mapa. Estos son los potenciadores: Potenciadores: Repite una vez más

		la demostración al usuario. Reduce la dificultad de la combinación de notas -4. Completa una combinación automáticamente.
Conflictos	El jugador debe responder la mayor cantidad de preguntas sin equivocarse.	El jugador debe presionar las notas en el orden correcto.
Frontera o Límite	Responder todas las preguntas correctamente.	Completar la secuencia de notas correctamente.
Resultado	Alcanzar la mayor puntuación en el ranking.	Alcanzar la mayor puntuación en el ranking.

2.3 Especificación de mecanismos

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

Sabios del Son:

No.	Nombre	Descripción	Organización arquitectónica perteneciente	Referencias cruzadas
1	Mecanismo Menú principal	• Objetos: Botones Jugador • Propiedades: • Comportamientos: Mecanismo de prueba para comprobar la funcio-	Mecanismo alterno	Documento de diseño de Sabios del Son

		nalidad de regresar al menú del juego base • Relaciones: M8		
2	Mecanismo Pregunta	• Objetos: Panel • Propiedades: Panel: Texto con la pregunta • Comportamientos: Se muestra en un panel una serie de preguntas correspondientes a la casa del llamo de la provincia en la que se encuentra el jugador • Relaciones: M3,M4,M5	Mecanismo núcleo	Documento de diseño de Sabios del Son
3	Mecanismo Respuestas	• Objetos: Jugador Botones • Propiedades: Jugador: Seleccionar una respuesta Botones: Respuestas • Comportamientos: Una vez que el jugador se encuentre ya en la partida debe seleccionar una de las cuatro posibles respuestas en concordancia con la pregunta actual, siendo una de estas correcta y el resto incorrectas. • Relaciones: M2, M4, M5	Mecanismo núcleo	Documento de diseño de Sabios del Son
4	Mecanismo Comodín	• Objetos: Botón Jugador • Propiedades: Botones: Apicuba, EGREM, Ron Cuba Jugador: Realizar acciones • Comportamientos: El jugador puede seleccionar uno de los tres comodines disponibles para facilitar la pregunta, Apicuba señala la respuesta correcta, EGREM cambia la pregunta actual por otra disponible, Ron Cuba elimina dos posi-	Mecanismo núcleo	Documento de diseño de Sabios del Son

		bles respuestas. • Relaciones: M2, M3		
5	Mecanismo Tiempo restante	• Objetos: Temporizador • Propiedades: Temporizador: Señala un tiempo • Comportamientos: Un temporizador marcará un tiempo límite establecido e irá decrementando hasta llegar a 0 lo cual cambiará a la siguiente pregunta sin otorgar punto al jugador. • Relaciones: M2, M3	Mecanismo núcleo	Documento de diseño de Sabios del Son
6	Mecanismo Contador de preguntas	• Objetos: Contador • Propiedades: Contador: Marca un número de preguntas • Comportamientos: Este contador se encarga de llevar un registro total de las preguntas totales y restantes que le quedan al jugador para completar el nivel. • Relaciones M2,M3	Mecanismo núcleo	Documento de diseño de Sabios del Son
7	Mecanismo Puntuación	• Objetos: Puntos Jugador • Propiedades: Puntos: cantidad de puntos obtenidos por el jugador Jugador: Obtiene puntos al seleccionar respuestas correctas • Comportamientos: Al seleccionar una respuesta correcta el jugador obtiene una cantidad de puntos que luego serán almacenados en un ranking. • Relaciones: M3	Mecanismo núcleo	Documento de diseño de Sabios del Son
8	Mecanismo Pausa	• Objetos Botones Jugador	Mecanismo alterno	Documento de diseño de Sa-

		• Propiedades Botones: Continuar, Reintentar, Opciones, Salir Jugador: Puede seleccionar las opciones • Comportamientos Durante una partida el jugador puede pausar y acceder a un menú que le permitirá resumir, reintentar el desafío desde el principio o salir al menú principal. • Relaciones M1, M2		bios del Son

Afinación de instrumentos:

No.	Nombre	Descripción	Organización arquitectónica perteneciente	Referencias cruzadas
1	Mecanismo Menú principal	• Objetos: Botones Jugador • Propiedades: • Comportamientos: Mecanismo de prueba para comprobar la funcionalidad de regresar al menú del juego base • Relaciones: M7	Mecanismo alterno	Documento de diseño de Afinación de instrumentos
2	Mecanismo Rueda de percusión	• Objetos: Botón • Propiedades: Botones con diferentes colores y notas asociadas • Comportamientos: Se alumbran y emiten una nota según la secuencia del nivel correspondiente. • Relaciones: M3,M4,M5	Mecanismo núcleo	Documento de diseño de Afinación de instrumentos
3	Mecanismo Multiplicador	• Objetos: Contador • Propiedades:	Mecanismo núcleo	Documento de diseño de Afi-

		• Comportamientos: Aumenta el tamaño en dependencia de que tan largo sea el combo del jugador • Relaciones: M2, M4, M5		nación de instrumentos
4	Mecanismo Potenciador	• Objetos: Botón Jugador • Propiedades: Botones Jugador: Realizar acciones • Comportamientos: El jugador puede seleccionar uno de los tres potenciadores disponibles ayudarle a superar el nivel Potenciadores: Repite una vez más la demostración al usuario. Reduce la dificultad de la combinación de notas -4. Completa una combinación automáticamente. • Relaciones: M2, M3	Mecanismo núcleo	Documento de diseño de Afinación de instrumentos
5	Mecanismo Combo	• Objetos: Contador • Propiedades: Contador: Marca la cantidad de secuencias obtenidas y a obtener • Comportamientos: Este contador se encarga de llevar un registro total de la cantidad de secuencias correctas realizadas por el usuario y las restantes para completar el nivel. • Relaciones M2,M3	Mecanismo núcleo	Documento de diseño de Afinación de instrumentos
6	Mecanismo Puntuación	• Objetos: Puntos Jugador • Propiedades: Puntos: cantidad de puntos obtenidos por el jugador Jugador: Obtiene	Mecanismo núcleo	Documento de diseño de Afinación de instrumentos

		puntos al realizar una secuencia de forma correcta • Comportamientos: Al realizar una secuencia de forma correcta el jugador obtiene una cantidad de puntos que luego serán almacenados en un ranking. • Relaciones: M3		
7	Mecanismo Pausa	• Objetos Botones Jugador • Propiedades Botones: Continuar, Reintentar, Opciones, Salir Jugador: Puede seleccionar las opciones • Comportamientos Durante una partida el jugador puede pausar y acceder a un menú que le permitirá resumir, reintentar el desafío desde el principio o salir al menú principal. • Relaciones M1, M2	Mecanismo alterno	Documento de diseño de Afinación de instrumentos

2.3.1 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 (*Introducing EdrawMax 10*, s. f.). 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.

Mecanismos Núcleos: Representan a los mecanismos centrales del videojuego que constituyen el sentido de enganche del 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 que el jugador avance hacia todos los niveles del juego y en todas las modalidades tanto online como offline.

Mecanismos Alternos: Amplían el espectro de funcionalidades del videojuego usando los mismos mecanismos núcleos. Permiten tener otros elementos en el videojuego que hacen que la experiencia del jugador mejore, ya que poseen una dinámica más activa.

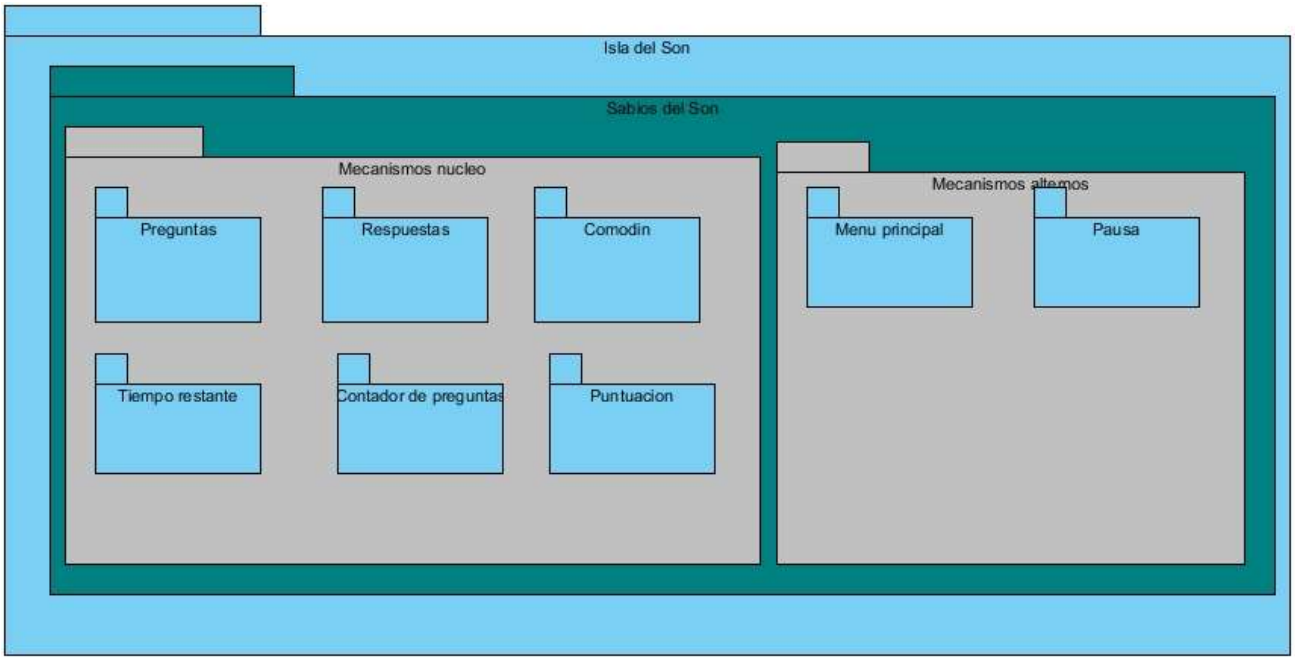


Fig 5. Diagrama de paquetes de los mecanismos de Sabios del Son.

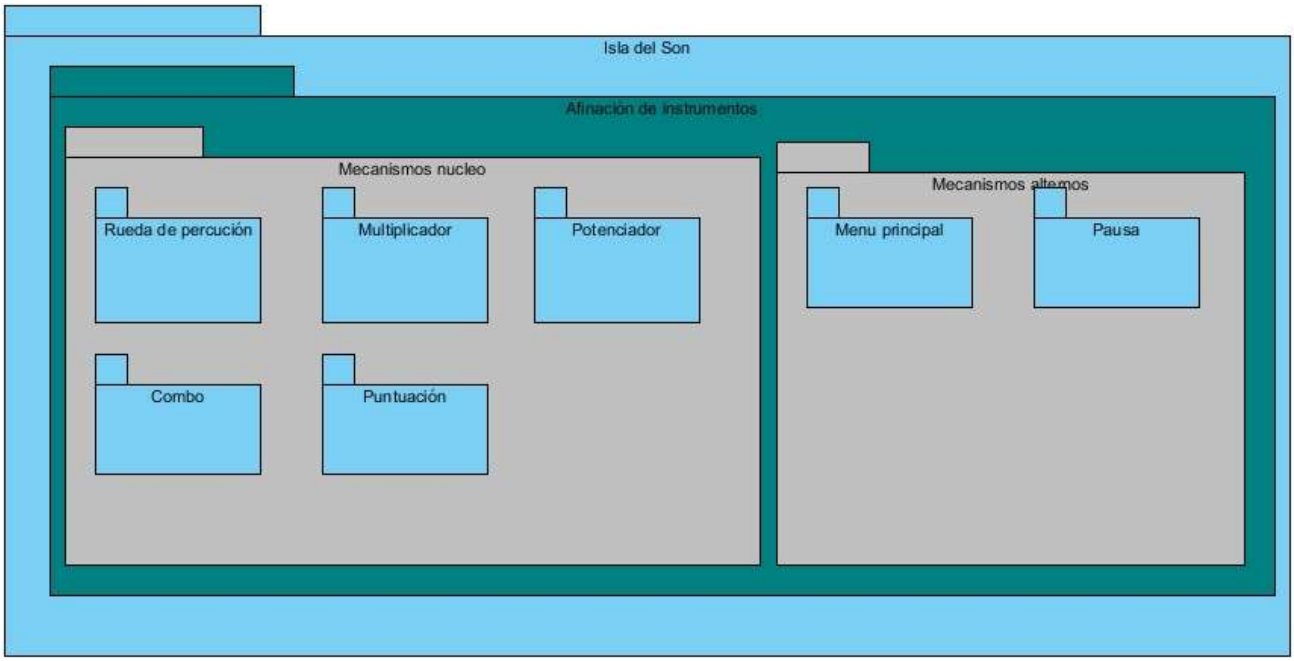


Fig 6. Diagrama de paquetes de los mecanismos de Afinación de instrumentos. Fuente: Elaboración propia.

2.4 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” (Cruz & Andrés, 2017). 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 (Cruz & Andrés, 2017). 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.

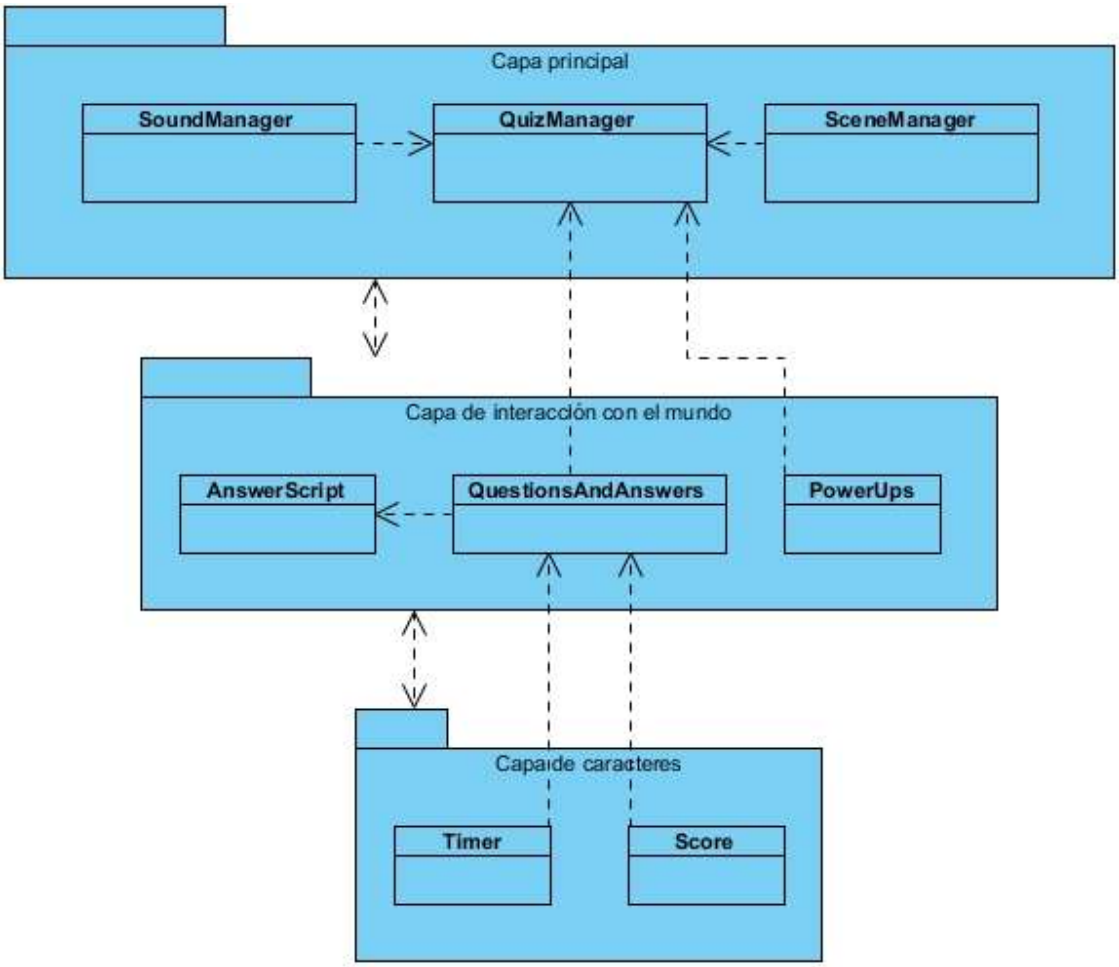


Fig 7. Capas de la arquitectura de Sabios del Son. Fuente: Elaboración propia.

Sabios del Son

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

- *QuizManager*: es el controlador que se encarga de gestionar todo el funcionamiento de las preguntas, las respuestas y los comodines.
- *SceneManager*: es el controlador que se encarga de los cambios de escena.
- *AudioManager*: es el controlador que se encarga de reproducir los diferentes efectos de sonido y la música del juego.

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

- *Answers*: es el controlador que se encarga de gestionar el comportamiento de los botones de respuesta.
- *QuestionsAndAnswers*: es el controlador que gestiona el texto de las preguntas con sus respuestas asociadas.
- *PowerUps*: es el controlador encargado de gestionar el comportamiento de cada comodín y validar sus condiciones.

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

- *Timer*: es el controlador que se encarga de la cuenta regresiva de cada pregunta.
- *Score*: es el controlador que gestiona la cantidad de respuestas correctas del jugador, así como su puntuación total.

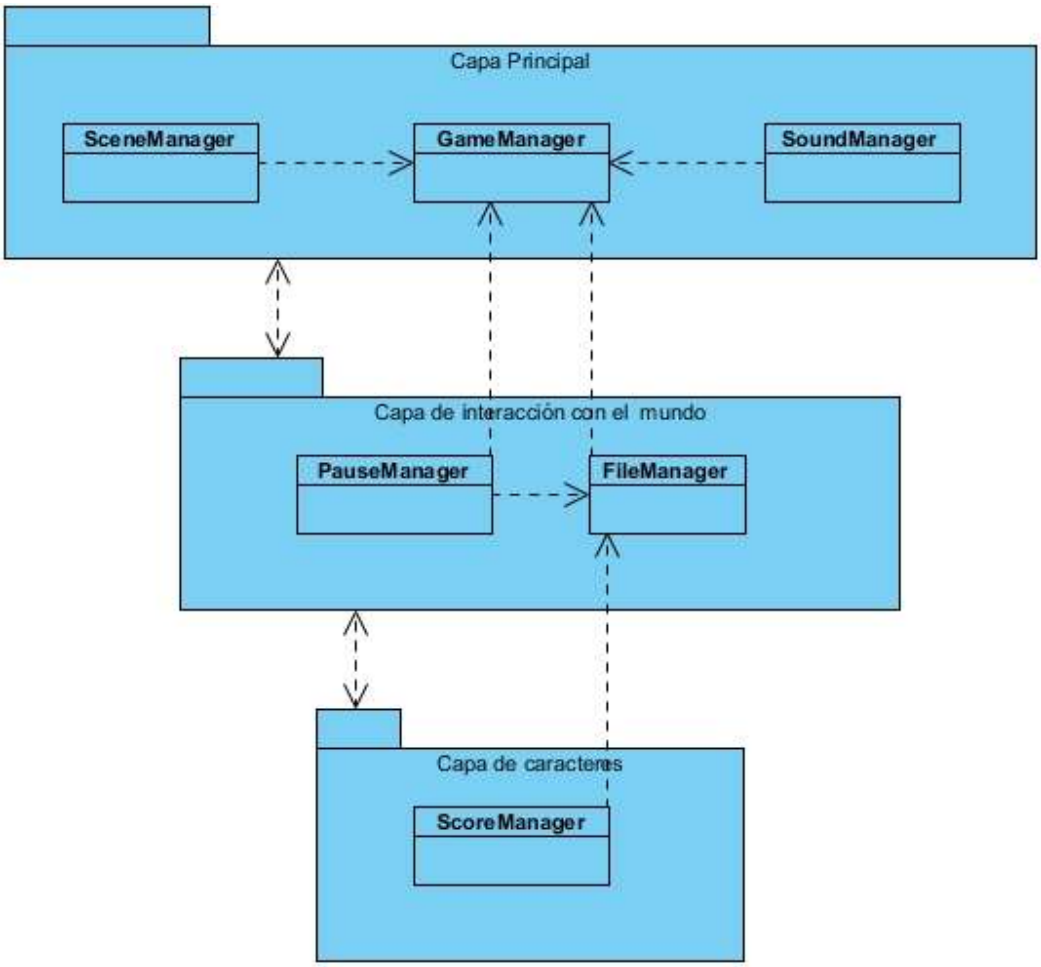


Fig 8. Capas de la arquitectura de Afinación de instrumentos. Fuente: Elaboración propia.

Afinación de instrumentos

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

- *GameManager*: es el controlador que se encarga de gestionar todo el funcionamiento de las mecánicas base del juego.
- *SceneManager*: es el controlador que se encarga de los cambios de escena.
- *AudioManager*: es el controlador que se encarga de reproducir los diferentes efectos de sonido y la música del juego.

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

- *PauseManager:* es el encargado de controlar los cambios de fondo y las funcionalidades de los botones en el menú de pausa.
- *FileManager*: es el controlador que se encarga de leer los datos provenientes del juego principal y guardar los datos necesarios para exportar.

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

- *ScoreManager:* es el controlador encargado de realizar los cálculos relacionados con la puntuación del jugador, así como actualizar como se muestran estos datos en pantalla.

2.5 Patrones de diseño

Los patrones de diseño consisten en la descripción de un problema de diseño dentro de un contexto dado y la definición de una posible solución correcta al mismo (Aldas Mena & Andrade Cadena, 2011). 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).

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.

- 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 *GameMaster* que dieron lugar a la implementación de las clases *ScoreManager* y *FileManager* 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 (Botero Tabares, 2010), se simplifica el mantenimiento, favorece una alta cohesión y facilita la reutilización. Este se puede observar por ejemplo en las clases *AudioManager* y *SceneManager* que poseen una cantidad reducida de relaciones con otras clases para evitar los problemas mencionados anteriormente.

2.6 Diagrama de Clases

En este se muestran las relaciones de las diferentes clases que componen la propuesta de solución, así como sus atributos y métodos.

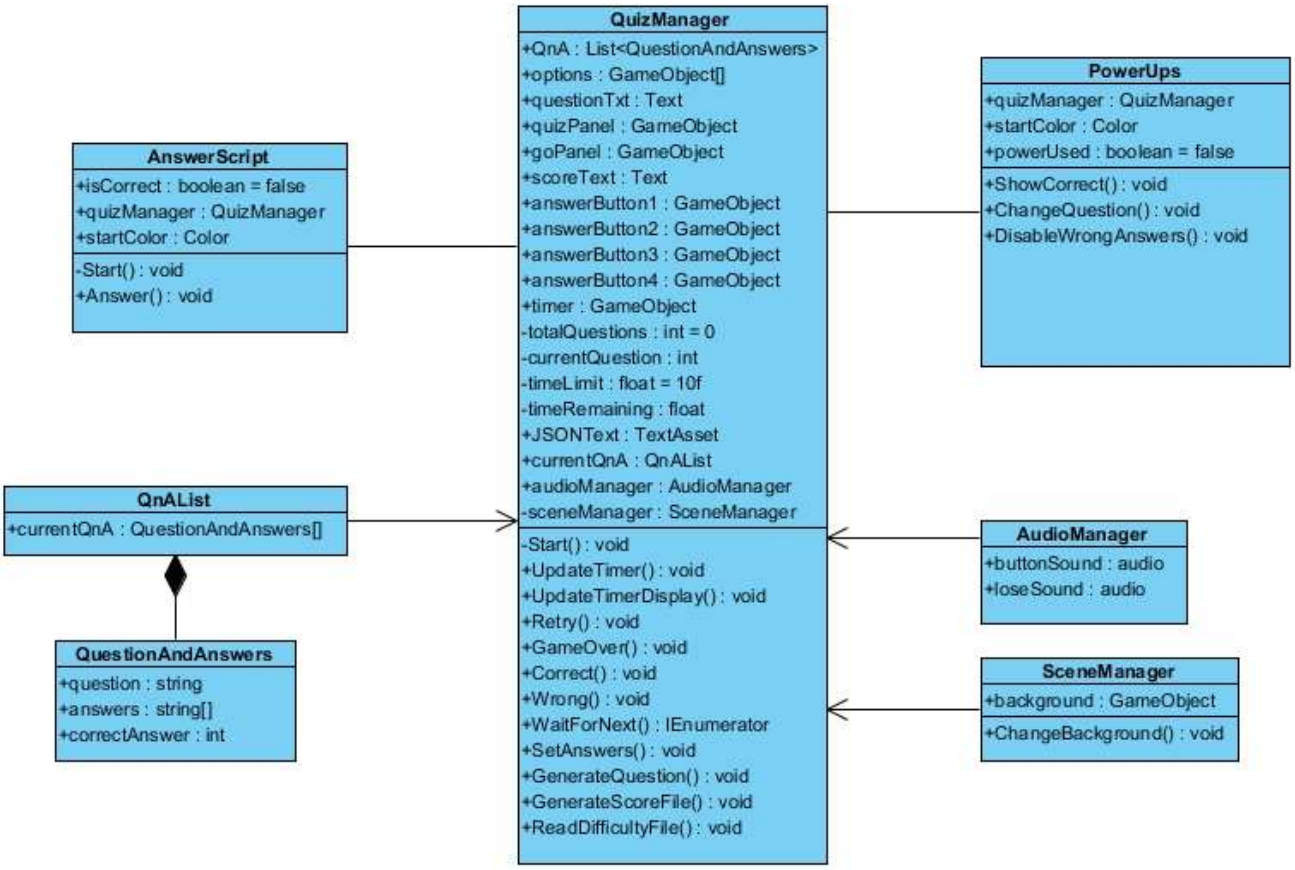


Fig 9. Diagrama de clases (Sabios del son). Fuente: Elaboración propia.

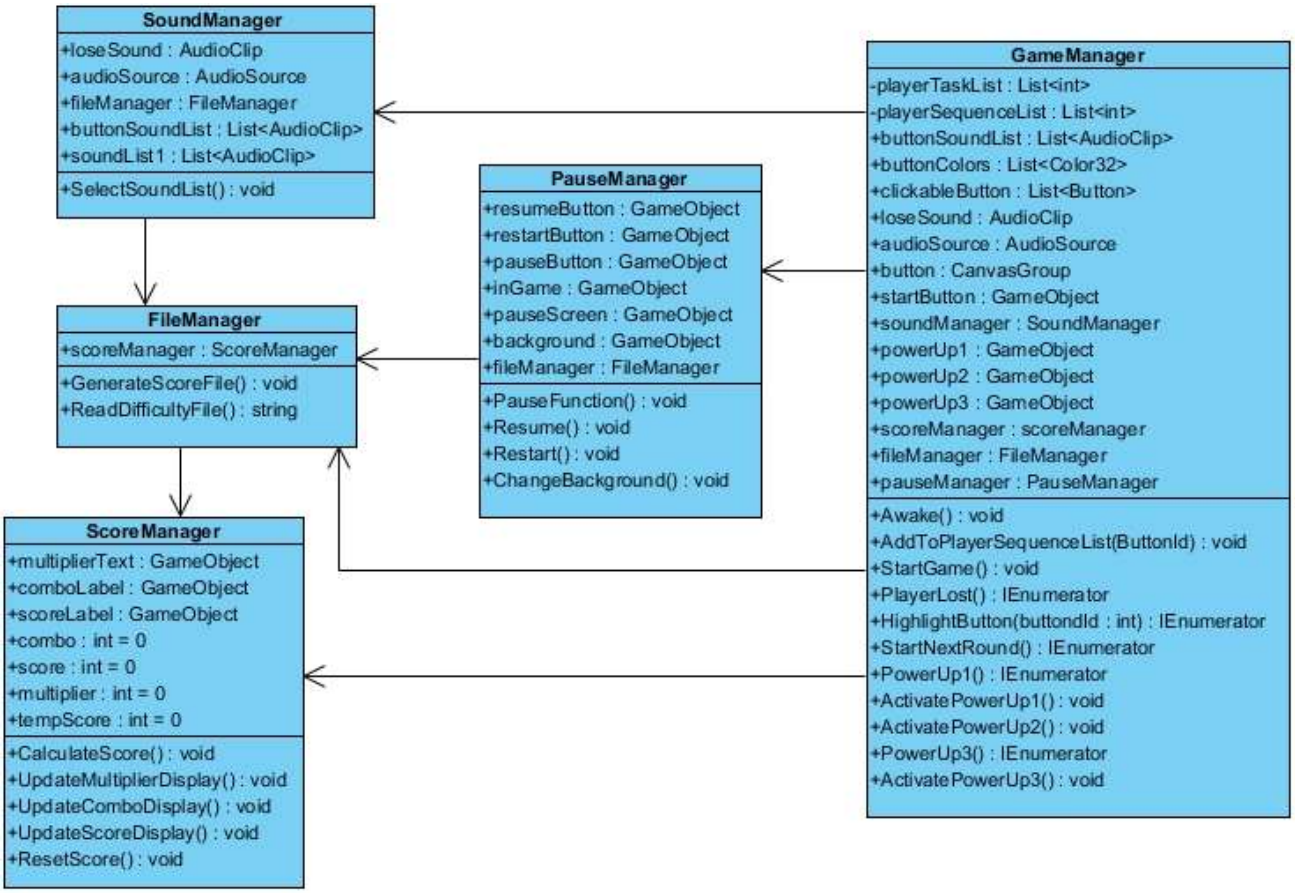


Fig 10. Diagrama de clases (Afinación de instrumentos). Fuente: Elaboración propia.

2.7 Representación del comportamiento

En las siguientes figuras se muestran los diagramas de comportamiento de ambos minijuegos durante la partida y todas las funcionalidades que se manifiestan en esta.

Diagramas de estado:

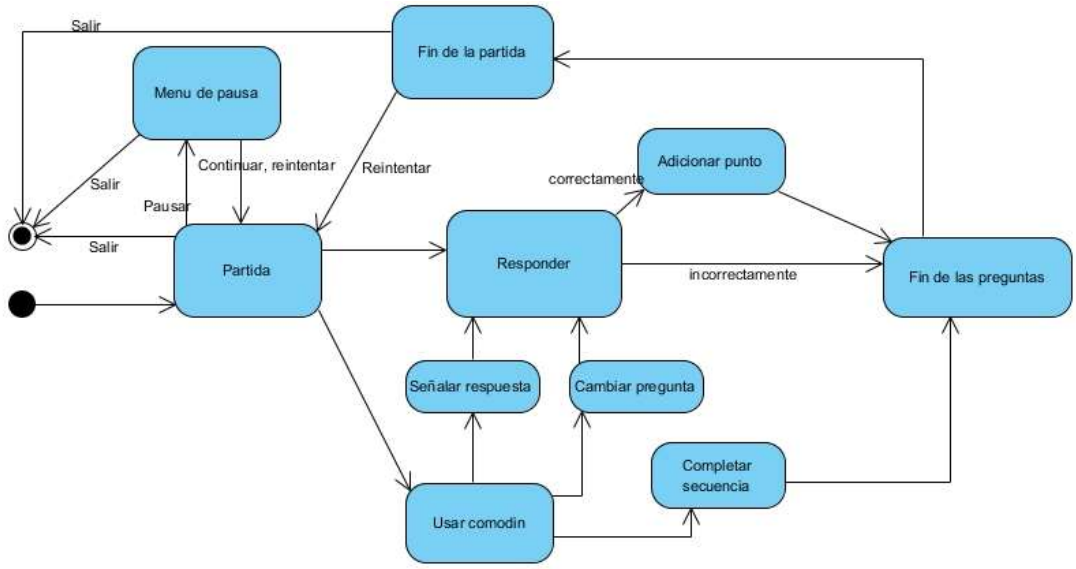


Fig 11. Diagrama de estado durante la partida (Sabios del son). Fuente: Elaboración propia.

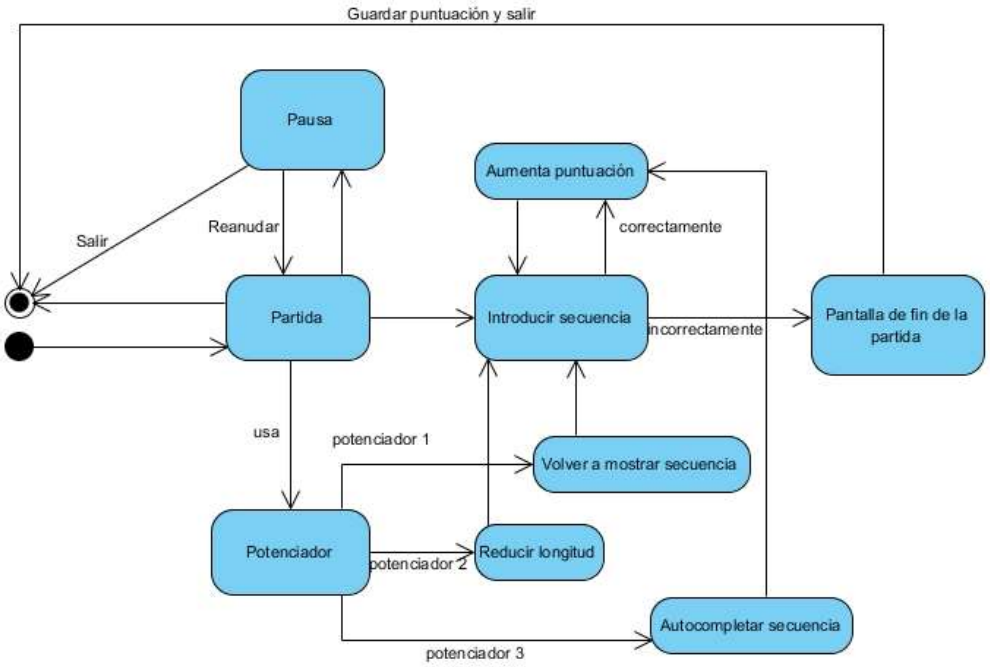


Fig 12. Diagrama de estado durante la partida (Afinación de Instrumentos). Fuente: Elaboración propia.

2.8 Descripción de requisitos no funcionales

Teniendo en cuenta la norma ISO/IEC 25010 se definieron los siguientes requisitos no funcionales:

RnF 1. Usabilidad. La aplicación debe ser lo más navegable posible para que el usuario con poco conocimiento técnico no se abrume en el videojuego y pueda interactuar de forma correcta con los mecanismos.

RnF 2. Compatibilidad. La aplicación debe ser compatible con el juego base Isla del Son y el resto de sus componentes.

RnF 3. Portabilidad. La aplicación será diseñada para adaptarse a dispositivos con sistema Android.

Conclusiones parciales

Con la culminación del presente capítulo, se arribó a la conclusión de que las bases de la solución del videojuego, quedaron bien definidas gracias a los artefactos generados por la metodología empleada, la cual permitió comprender las funcionalidades y el contenido concreto para la implementación del comportamiento de cada elemento que intervienen en los diferentes escenarios, para ello se utilizó una arquitectura que combina los estilos arquitectónicos: basada en capa y basada en componentes. Adicionalmente se describieron los principales requisitos no funcionales y elementos formales que permiten representar las características físicas y una base lógica para el proceso de desarrollo de software.

Capítulo III: Validación de la solución propuesta

En el presente capítulo se define el estándar de codificación, la integración de los minijuegos con el juego macro y se representa la estructura de los videojuego a nivel de componentes. También se explican los resultados obtenidos en las pruebas alfa y beta.

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 usa el estilo de escritura *CamelCase*. Es un estilo de identificador donde las palabras separadas dentro de un identificador están en minúsculas, pero están delimitadas poniendo en mayúscula la primera letra de cada palabra interna (Leino, 2016). Este se usa de dos formas:

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

3.1.1 Definición de clases: las clases comienzan con 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 estilo determinado. Ejemplo de ello se muestra a continuación:

```
    0 references | 1 scene usage | 0 prefab usages
8   public class GameManager : MonoBehaviour
```

3.1.2 Declaración de variables: los identificadores de las variables comenzaran con letra minúscula y si están conformados por palabras compuestas, estas estarán de manera continua, comenzando cada nueva palabra con mayúscula como se muestra a continuación:

```
6 references
27  public int currentQuestion;
4 references
28  public float timeLimit = 10f;
8 references
29  private float timeRemaining;
```

3.1.3 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:

```
0 references
55  public void StartGame()
56  {
57    StartCoroutine(StartNextRound());
58    startButton.SetActive(false);
59  }
```

3.2 Integración con el juego macro

Como parte de los requisitos de la solución es necesario que los minijuegos se integren con el videojuego Isla del 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, la preguntas en caso de Sabios del Son y las notas de la rueda de percusión en el otro minijuego. 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 jugo 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.

3.3 Diagramas de componentes

En estos se muestran los diferentes componentes que forman ambos minijuegos y sus relaciones.

Diagrama de componentes de Sabios del Son:

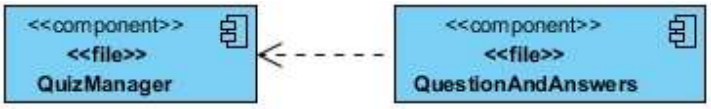


Fig 13. M2 Preguntas. Fuente: Elaboración propia.

Fig 14. M3 Respuestas. Fuente: Elaboración propia.

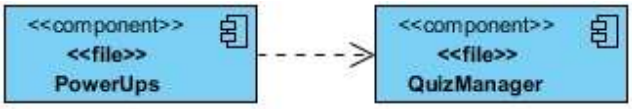


Fig 15. M4 Comodines. Fuente: Elaboración propia.

El resto de mecanismos están controlados por el componente

Diagrama de componentes de Afinación de Instrumentos:

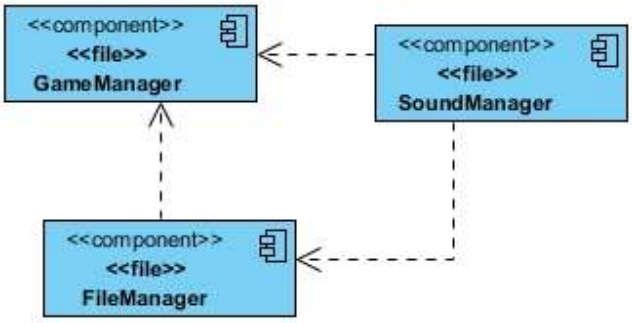


Fig 16. M2 Rueda de percusión. Fuente: Elaboración propia.

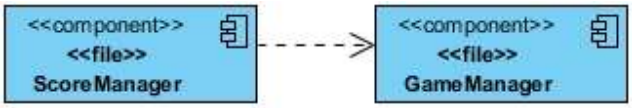


Fig 17. M3 Multiplicador M4 Potenciador M5 Combo M6 Puntuación. Fuente: Elaboración propia.

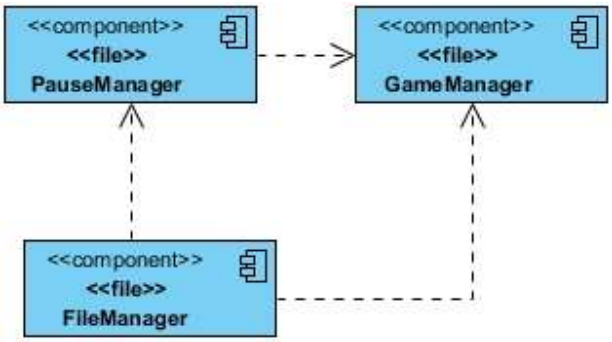


Fig 18. M7 Pausa. Fuente: Elaboración propia.

3.4 Pruebas

En la Ingeniería del software, las pruebas de aceptación se realizan para establecer el grado de confianza en un sistema, partes del mismo o en sus características no funcionales. La confianza en el sistema estará determinada por su grado de adherencia a las necesidades, requerimientos y procesos de negocio solicitados por el usuario o cliente. Es en función a estos que el usuario debe decidir si acepta o no el sistema que le está siendo entregado. Estas pruebas son fundamentales para asegurar el éxito de la implementación final de un proyecto de ingeniería de software, por lo cual deben incluirse obligatoriamente en el plan de pruebas de software (Bath & McKay, 2014).

Prueba Alpha: La prueba Alpha es llevada a cabo frente al desarrollador por un grupo representativo de usuarios finales. El software se prueba en un entorno natural donde el desarrollador se encarga de registrar los errores y problemas de uso. Las pruebas alpha son realizadas en un ambiente controlado (Pressman, 2005).

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 (Pressman, 2005).

A continuación, se describe el registro de defectos encontrados durante las pruebas Alpha realizadas al minijuego **Sabios del Son**:

Iteración	Número	Prioridad	Descripción	Ubicación	Tipo	Respuesta
1	1.1	Media	No cargan todas las preguntas.	Escena de partida	Error de Interfaz	Resuelto
1	1.2	Baja	El botón de pausa no despliega dicho menú.	Escena de partida	Funcionalidad	Resuelto
1	1.3	Alta	Marca el botón de respuesta correcta como incorrecta.	Botones de respuesta	Funcionalidad	Resuelto
2	2.1	Media	Se puede reutilizar los comodines infinitamente	Comodines	Funcionalidad	Resuelto
2	2.2	Alta	El temporizador no cambia la pregunta al llegar a 0	Temporizador	Funcionalidad	Resuelto
2	2.3	Alta	Se puede marcar varias respuestas a la vez en la misma pregunta	Botones de respuesta	Funcionalidad	Resuelto
3	3.1	Baja	No cambian las preguntas según el escenario	Escena de partida	Error de interfaz	Resuelto
3	3.2	Baja	No cambian las respuestas según el escenario	Botones de respuesta	Error de interfaz	Resuelto

A continuación, se describe el registro de defectos encontrados durante las pruebas Alpha realizadas al minijuego **Afinación de Instrumentos**:

Iteración	Número	Prioridad	Descripción	Ubicación	Tipo	Respuesta
1	1.1	Media	Los botones no se iluminan al mostrar la secuencia	Rueda de percusión	Error de Interfaz	Resuelto
1	1.2	Baja	Los sonidos no se reproducen correctamente	Escena de partida	Funcionalidad	Resuelto
1	1.3	Alta	El contador de combo no sube al completar la secuencia	Escena de partida	Error de interfaz	Resuelto
2	2.1	Media	El potenciador 1 no realiza ninguna función al usarlo	Potenciadores	Funcionalidad	Resuelto
2	2.2	Alta	El potenciador 2 puede eliminar la secuencia completamente	Potenciadores	Funcionalidad	Resuelto
2	2.3	Baja	Al reiniciar el juego la puntuación empieza en 5	Escena de partida	Error de Interfaz	Resuelto
3	3.1	Baja	El multiplicador de	Escena de	Error de inter-	Resuelto

			puntos supera el tamaño límite	partida	faz	
3	3.2	Baja	Al usar el potenciador 3 la puntuación se duplica	Botones de respuesta	Funcionalidad	Resuelto
3	3.3	Alta	Al usar un potenciador en el momento en el que se reproducía la secuencia el juego deja de funcionar	Potenciadores	Funcionalidad	Resuelto

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:

- Enrique Ferreiro Ávila (estudiante de 5to año en Ingeniería en Ciencias Informáticas, facultad 4)
- Roberto Vega Rojas (estudiante de 5to año en Ingeniería en Ciencias Informáticas, facultad 4)
- Yilber Serrano (estudiante de 5to año en Ingeniería en Ciencias Informáticas, facultad 4)
- Daniel Chaviano Pérez (estudiante de 4to año en Ingeniería en Ciencias Informáticas, facultad 4)

Se realizaron cuatro iteraciones de las pruebas Beta a continuación se muestra un registro de las no conformidades encontradas por los probadores designados durante el proceso para ambos juegos:

Sabios del Son

Iteración	Número	Prioridad	Descripción	Ubicación	Tipo	Respuesta	Probador
1	1.1	Baja	Error ortográfico	Preguntas	Ortografía	Resuelto	Yilber Serrano
1	1.2	Media	Se muestran respuestas que no concuerdan con la pregunta	Botones de respuesta	Error de Interfaz	Resuelto	Yilber Serrano
1	1.3	Baja	No se resalta la respuesta correcta al seleccionarla después de usar un potenciador	Botones de respuesta	Error de interfaz	Resuelto	Roberto Vega
1	1.4	Media	Al seleccionar una respuesta cuando el contador llega a 0 no se adicionan puntos	Escena de partida	Funcionalidad	Resuelto	Enrique Ferreiro
1	1.5	Media	Al reintentar la puntuación de la partida ante-	Escena de partida	Funcionalidad	Resuelto	Daniel Chaviano

			rior se suma con la nueva				
1	1.6	Media	Al usar el botón de pausa no se detenía el temporizador	Botón de pausa	Funcionalidad	Resuelto	Enrique Ferreiro
2	2.1	Baja	Ortografía	Botones de respuesta	Ortografía	Resuelto	Daniel Chaviano
2	2.2	Media	Las respuestas no están de acorde con la provincia	Botones de respuesta	Error de interfaz	Resuelto	Yilber Serrano
2	2.3	Baja	El temporizador al llegar a 0 vuelve a su valor máximo antes de cambiar la pregunta	Temporizador	Error de interfaz	Resuelto	Roberto Vega
3	3.1	Baja	Al intentar minimizar el juego la dimensión de los botones cambia incorrectamente	Interfaz	Error de interfaz	Resuelto	Roberto Vega

Afinación de Instrumentos:

Iteración	Número	Prioridad	Descripción	Ubicación	Tipo	Respuesta	Probador
1	1.1	Media	Al dar en el botón empezar queda superpuesto al resto de botones	Botón Empezar	Funcionalidad	Resuelto	Roberto Vega
1	1.2	Baja	Error Ortográfico	Menú de Pausa	Ortografía	Resuelto	Yilber Serrano
1	1.3	Alta	Se puede interactuar con los botones mientras se muestra una secuencia	Botones de percusión	Funcionalidad	Resuelto	Roberto Vega
2	2.1	Media	Se puede interactuar con todos los potenciadores durante la misma secuencia	Potenciadores	Funcionalidad	Resuelto	Roberto Vega
2	2.2	Alta	Al intentar cargar otra provincia no cargan los	Escena de partida	Funcionalidad	Resuelto	Enrique Ferreiro

			sonidos				
2	2.3	Baja	Se puede superar el límite del multiplicador al usar el potenciador 3 en la última secuencia	Escena de partida	Funcionalidad	Resuelto	Daniel Chaviano
2	2.4	Alta	Al usar el potenciador 1 varias veces seguidas el juego lanza un error	Potenciador	Funcionalidad	Resuelto	Daniel Chaviano
3	3.1	Media	Al completar la cantidad de secuencias requeridas no se reestablecen los usos de los potenciadores	Potenciador	Funcionalidad	Resuelto	Yilber Serrano
3	3.2	Media	Después de 8 secuencias no se reproducen los	Escena de partida	Funcionalidad	Resuelto	Yilber Serrano

			sonidos				

Antes de comenzar las iteraciones siguientes se dio solución a los problemas encontrados en las anteriores para luego pasar a resolver las no conformidades que se encontraron en las próximas iteraciones. Después del 3er ciclo de cada prueba no se encontraron mas no conformidades. En las siguientes figuras se muestran un total de las no conformidades encontradas durante cada iteración durante las pruebas Alpha y Beta:

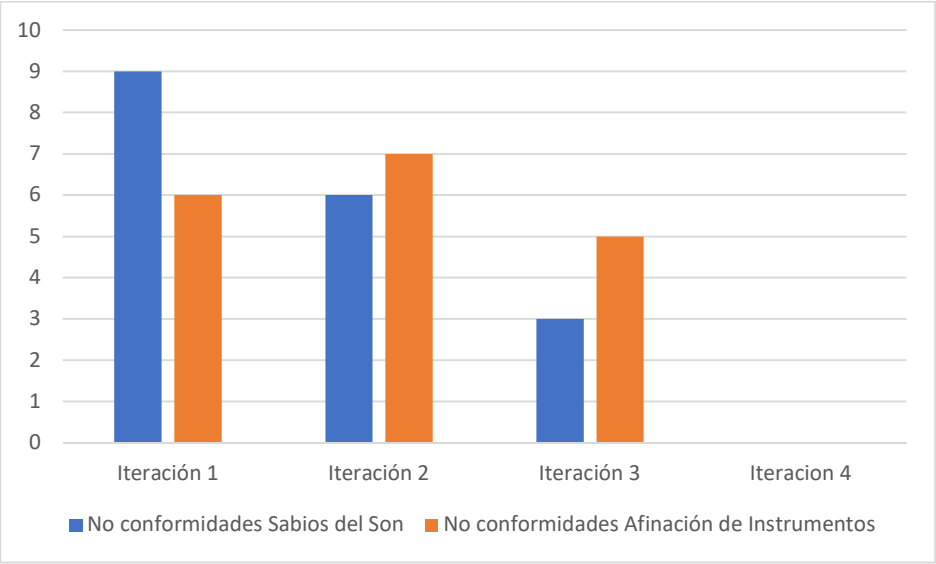


Fig 19. No conformidades durante la validación por iteraciones

Conclusiones del capítulo

Al finalizar el presente capítulo se desarrolló satisfactoriamente la solución propuesta, logrando así la implementación de dos minijuegos para el videojuego Isla del Son. Se definieron los estándares de codificación y se representaron los diagramas de componente lo que permite un mejor entendimiento de la implementación. Se definieron y aplicaron las pruebas alfas, con lo que se elevó la calidad del software a partir de los resultados obtenidos y las acciones que se llevaron a cabo para darle solución a las no conformidades encontradas.

CONCLUSIONES

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

- La elaboración del diseño teórico ratificó la necesidad de proporcionar al videojuego Isla del Son de formas de aumentar el interés y la familiaridad de los jugadores por este género musical.
- La investigación acerca de las tendencias tecnológicas actuales, técnicas y características utilizadas para el desarrollo de videojuegos de agilidad mental, mostró elementos a tener en cuenta en la solución, como fueron el motor de videojuego Unity y el lenguaje de programación C#.
- El empleo de patrones de diseño, el uso de una arquitectura por capas y componentes, y la elaboración de mecanismos de jugabilidad, posibilitaron definir la lógica de la propuesta de solución en el momento de su implementación.
- Se implementaron dos componentes en forma de minijuegos con estilo trivia y memoria a corto plazo para el videojuego Isla del Son.
- Las pruebas realizadas permitieron verificar que los minijuegos y los mecanismos desarrollados cumplen con la solución propuesta, así como el funcionamiento que se espera de la jugabilidad y la experiencia del usuario, validando además la calidad de la solución.

RECOMENDACIONES

Teniendo en cuenta la solución proporcionada se pueden dar varias recomendaciones para investigaciones futuras:

- Aumentar la cantidad de preguntas del componente Sabios del Son.
- Implementar un sistema que modifique la puntuación del jugador según los potenciadores usados.

REFERENCIAS BIBLIOGRÁFICAS

1. Albahari, J. (2022). *C# 10 in a Nutshell*. O'Reilly Media, Inc.
2. Aldas 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.
3. Bath, G., & McKay, J. (2014). *The software test engineer's handbook: A study guide for the ISTQB test analyst and technical test analyst advanced level certificates 2012*. Rocky Nook, Inc.
4. Belli, S., & López Raventós, C. (2013). Breve historia de los videojuegos. *Athenea Digital: Revista de Pensamiento e Investigacion Social*.
5. Botero Tabares, R. de J. (2010). *Patrones grasp y anti-patrones: Un enfoque orientado a objetos desde logica de programacion*.
6. *COSMOX | Videojuego La Neurona 1*. (s. f.). Recuperado 25 de junio de 2023, de https://cosmox.uci.cu/frontend/web/videogame/24
7. Cruz, P., & Andrés, A. (2017). *Arquitectura de software para videojuegos desarrollados sobre el motor de juego Unity 3D* [B.S. thesis]. Universidad de las Ciencias Informáticas. Facultad 4.
8. *Documentation for Visual Studio Code*. (s. f.). Recuperado 22 de noviembre de 2023, de https://code.visualstudio.com/docs
9. Fernández, O. (2001). Antología historica del Son cubano. *Repertorio Americano*, *11*, 48-54.
10. Guins, R. (2014). *Game after: A cultural study of video game afterlife*. MIT Press.
11. Hamilton, D. (2020). *Estándares de codificación de software y pautas de programación*. Parasoft. https://es. parasoft. com/blog/an-ounce-of-prevention-software ….
12. Hernández, A., Pérez, K., & Correa, O. (2017). Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos. *Revista Antioqueña de las Ciencias Computacionales*, *7*(1).
13. *Introducing EdrawMax 10*. (s. f.). Edrawsoft. Recuperado 22 de noviembre de 2023, de https://www.edrawsoft.com/es/article/package-diagram-uml.html
14. 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
15. Monsalve Vidales, C. P., Ospina Sánchez, V. E., & García Ospina, K. J. (2016). *Creación de software tipo juego serio (trivia) para el aprendizaje de las bases de datos*. https://repositorio.itm.edu.co/handle/20.500.12622/4469

16. Paradigm, V. (2013). Visual paradigm for uml. *Visual Paradigm for UML-UML tool for software application development*, *72*.

17. *PDF de programación—GUÍA DE CONSTRUCCIÓN DE SOFTWARE EN JAVA CON PATRONES DE DISEÑO*. (s. f.). Recuperado 22 de noviembre de 2023, de https://www.lawebdelprogramador.com/pdf/9527-GUIA-DECONSTRUCCION-DE-SOFTWARE-EN-JAVA-CON-PATRONES-DE-DISENO.html

18. Pressman, R. S. (2005). *Software engineering: A practitioner's approach*. Palgrave macmillan.

19. RUÈZ-VARGAS, J. (1994). La memoria a corto plazo. *Psicología de la memoria. Madrid: Alianza editorial*, 117-150.

20. Susi, T., Johannesson, M., & Backlund, P. (2007). *Serious games: An overview*.

21. Tavinor, G. (2008). Definition of videogames. *Contemporary Aesthetics (Journal Archive)*, *6*(1), 16.

22. *Touch-Me—Arcade by Atari*. (s. f.). Recuperado 21 de noviembre de 2023, de https://www.arcade-museum.com/game_detail.php?game_id=12694

23. *Unity Documentation*. (s. f.). Unity Documentation. Recuperado 23 de mayo de 2023, de https://docs.unity.com/

24. *"Who Wants to Be a Millionaire?" – play the free online game*. (s. f.). Recuperado 21 de noviembre de 2023, de https://wwbm.com/