

Pontificia Universidad Javeriana Cali.
Facultad de Ingeniería y Ciencias.
Ingeniería de Sistemas y Computación.
Proyecto de Grado.

Prototipo de un sistema de evaluación del desarrollo psicométrico en niños a través de un sistema interactivo basado en gestos corporales.

Autor:

Maria Del Mar Villaquiran Dávila
Fabian Ernesto Ledezma Ledezma

Director:

Dr. Andrés Adolfo Navarro Newball

Junio del 2024



Pontificia Universidad
JAVERIANA
Cali

Abstract

The psychometric assessment of children with different sensory disabilities poses a challenge in the educational field, where the Wechsler Intelligence Scale for Children (WISC) shows limitations by not incorporating experiential learning. Children with various sensory disabilities, from early stages of life, can develop impairments in their thinking, speech, language, behavior, social, and emotional development, ultimately affecting their academic and professional performance. Therefore, this project took the initiative to create a video game with an interactive system that allows children with sensory disabilities to have a more inclusive and experiential experience, thereby improving early detection of issues and promoting their holistic development.

Keywords: Sensory disability, WISC Test, video game, experiential, interactive system.

Resumen

La evaluación psicométrica en niños con diferentes discapacidades sensoriales plantea un desafío en el área educativa, donde el Test de inteligencia para Niños Wechsler (WISC) muestra limitaciones al no incorporar experiencias vivenciales. Los niños con diferentes discapacidades sensoriales, desde etapas tempranas de la vida, pueden desarrollar afectaciones en su pensamiento, habla, lenguaje, conducta, desarrollo social y emocional; lo que eventualmente puede impactar su desempeño escolar y laboral. De modo que por medio de este proyecto se tomó la iniciativa de crear un videojuego de un sistema interactivo, que permita a los niños con discapacidades sensoriales tener una experiencia más inclusiva y vivencial, mejorando así la detección temprana de problemas y promoviendo su desarrollo integral.

Palabras claves: Discapacidad sensorial, Test WISC, videojuego, vivencial, sistema interactivo.

Tabla de contenido

1	Introducción.....	9
1.1	Planteamiento del problema.....	9
1.1.1	Formulación.....	11
1.1.2	Sistematización.....	11
1.2	Objetivos	12
1.2.1	Objetivo General.....	12
1.2.2	Objetivos específicos.....	12
1.3	Justificación	13
1.4	Alcances y Limitaciones	14
2	Marco Conceptual.....	15
2.1	Marco de Referencia	15
2.1.1	Áreas Temáticas.....	15
2.1.2	Marco Teórico.....	15
2.1.3	Trabajos Similares.....	21
3	Metodología.....	25
3.1.1	Tipo de estudio	25
3.1.2	Lista de actividades.....	25
4	Análisis	27
4.1	Actores del sistema.....	27
4.1.1	Jugador.....	27
4.1.2	Profesional Encargado.....	27
4.2	Requisitos	27
4.2.1	Requisitos funcionales	27
4.2.2	Requisitos no funcionales	29
4.3	Diagrama de casos de Uso	30
5	Diseño	32
5.1	Descripción del modelo del sistema.....	32
5.2	Implementación de Elementos del Test WISC en el Sistema.....	32
5.3	Diagrama de Clases	33
5.4	Diagrama de Secuencia.....	34
5.4.1	Escena 1: Menú.....	35
5.4.2	Escena 2: Crear Jugador	36
5.4.3	Escena 3: Seleccionar Jugador.....	37
5.4.4	Escena 4: Seleccionar Dificultad	39
5.4.5	Escena 5: Juego	40
5.4.6	Escena 6: Resultado	41
5.5	Diseño de Interfaz.....	42

5.6	Modelo de Datos	56
5.7	Diseño de Algoritmo	58
6	Implementación.....	59
6.1	Elección del motor de videojuegos.....	59
6.2	Descripción del sistema inicial	61
6.3	Metodología para Configurar la Gravedad en la Caída de Animales	62
6.3	Líneas de código relevantes.....	64
6.3.1	Aparición de animales	64
6.3.2	Controlador de colisiones.....	65
6.4	Sistema de almacenamiento de información jugadores	66
6.5	Sistema de almacenamiento de información de partidas.....	68
7	Pruebas y Validación.....	70
7.1	Descripción de la prueba	70
7.2	Resultados de las pruebas	70
7.3	Información de jugadores y preguntas	72
7.4	Encuesta a profesionales del Instituto para niños Ciegos y Sordos del Valle del Cauca	73
7.5	Validación.....	76
7.5.1	Requisitos funcionales	76
7.5.2	Requisitos no funcionales	81
8	Conclusiones y trabajos futuros	83
8.1	Conclusiones.....	83
8.2	Trabajos futuros.....	84
9	Bibliografía.....	85

1 Introducción

1.1 Planteamiento del problema

Durante las pruebas realizadas con el sistema original basado en Kinect, que consistía en la captura de animales, se evidenció que según la opinión del Profesor Gerardo Restrepo, experto en educación y neurodesarrollo, este sistema tenía posibilidades de ajustarse para apoyar procesos de evaluación [15]. En la actualidad, en el ámbito educativo se han evidenciado las evaluaciones psicométricas en los niños con discapacidad, por lo que representa un desafío muy importante para identificar las necesidades específicas y así poder promover su desarrollo integral.

Entre las herramientas utilizadas para abordar este problema, el Test de Inteligencia para Niños de Wechsler (WISC) [8] destaca como una valiosa herramienta para medir diversos aspectos de la capacidad intelectual en los niños. No obstante, el enfoque planteado genera dudas acerca de la efectividad y del propósito principal de estas evaluaciones, ya que no involucra experiencias vivenciales en los niños. El propósito del test WISC va más allá de ser un simple proceso de evaluación; detrás de esta prueba subyace la intención de capturar, medir y estandarizar datos cruciales, lo que permite a los profesionales interpretar los resultados y comprender los desafíos de los niños. No obstante, el proceso de administración del test WISC se enfrenta a desafíos que van desde la limitada atención de los niños durante las pruebas, es decir, la falta de una experiencia vivencial, hasta la falta de adaptación de la evaluación a las necesidades individuales y a las condiciones particulares de cada niño con discapacidad.[16]

El problema subyacente en todo este contexto, es la necesidad insatisfecha de un sistema de evaluación más sensible y efectivo, que permita la detección temprana de problemas y condiciones en los niños con discapacidad. Los niños en muchas ocasiones experimentan aburrimiento y desinterés durante las pruebas convencionales, lo que compromete la validez de los resultados y limita la utilidad de la evaluación en la toma de decisiones educativas. [14]

En este panorama, se destaca el proyecto colaborativo "Colombia Quebec: Narrativa, Discapacidad y Realidad Virtual", así como la labor del Instituto para Niños Ciegos y Sordos del Valle del Cauca. En relación con el sistema actual de selección de animales con gestos, este proyecto busca una innovación transformadora al tomar el concepto y la estructura de dicho sistema y adaptarlo para convertirlo en una herramienta de evaluación más eficaz. Se plantea utilizar el sistema de selección de animales basado en el Kinect 2 como base, y se trabajará en pulir diversos aspectos de su funcionamiento, como es la captura precisa de gestos, el conteo exacto de los animales visualizados en pantalla y la incorporación de distintos niveles de dificultad.

Estos esfuerzos buscan abordar el problema central de la falta de detección adecuada de problemas en los niños con discapacidad en Colombia. No obstante, el enfoque actual del proyecto, que se basa en aprendizaje a través de experiencias vivenciales, presenta limitaciones al no incorporar herramientas de evaluación más rigurosas y especializadas. El enfoque propuesto tiene el potencial de superar las limitaciones de las evaluaciones tradicionales al brindar a los niños con discapacidad una experiencia más interactiva y vivencial. Esta aproximación podría mejorar la validez y la utilidad de los resultados obtenidos. Además, al usar tecnologías y la realidad virtual, se podría adaptar la evaluación de manera precisa a las necesidades individuales de los niños y a sus condiciones específicas, lo que permitiría superar los desafíos de las pruebas convencionales.

1.1.1 Formulación

¿Cómo desarrollar un sistema de evaluación psicométrico a partir de un sistema interactivo gestual para niños con diversidad sensorial?

1.1.2 Sistematización

- ¿Cuáles elementos de un sistema de interacción gestual pueden aprovecharse para desarrollar un sistema de evaluación psicométrico para niños con diversidad sensorial?
- ¿Cómo diseñar un sistema de evaluación psicométrico basado en interacción gestual para niños con diversidad sensorial?
- ¿Cómo implementar el diseño propuesto?
- ¿Cómo validar el sistema propuesto?

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar un sistema de evaluación psicométrico a partir de un sistema interactivo gestual para niños con diversidad sensorial.

1.2.2 Objetivos específicos

- Estudiar cómo funciona el test WISC para identificar patrones de comportamientos en los niños y acoplarlos a sus necesidades.
- Diseñar un sistema basado en un sistema de selección preexistente, que permita la realización de una experiencia vivencial en los niños.
- Implementar el diseño utilizando tecnologías de realidades aumentada.
- Validar el prototipo implementado.
 - Dado que esto forma parte del proyecto colaborativo Quebec, ya se cuenta con el aval del comité ético del Instituto para niños Ciegos y Sordos, lo que nos permite hacer pruebas con los niños.

1.3 Justificación

Este trabajo de grado forma parte del proyecto colaborativo "Proyecto Colaborativo Colombia-Quebec", cuyo objetivo central es el desarrollo de un sistema de evaluación psicométrico, adaptado a través de una interfaz interactiva gestual diseñada para niños con diversidad sensorial, utilizando la tecnología de realidad virtual [5], específicamente adaptada para abordar los desafíos de distracción que a menudo enfrentan estos niños durante las pruebas psicométricas, como el Test de Inteligencia para Niños de Wechsler (WISC).

En Colombia, a pesar de los avances en la atención a niños con discapacidad, se ha identificado un problema recurrente en la evaluación de su capacidad intelectual. Los niños con discapacidad a menudo presentan dificultades para mantener la atención y la concentración durante las pruebas psicométricas tradicionales, lo que puede llevar a resultados poco precisos o incompletos. Esto plantea interrogantes sobre la eficacia y el propósito de tales evaluaciones en este contexto particular.

Nuestra propuesta busca abordar esta problemática mediante el desarrollo de un entorno de realidad virtual, adaptado y diseñado para capturar de manera más efectiva la atención de los niños con discapacidad y permitir una evaluación más precisa de sus habilidades intelectuales. Colaboraremos con profesionales y expertos en el campo de la neurología y neurociencias para asegurarnos de que esta adaptación sea efectiva y se ajuste a las necesidades individuales y las condiciones particulares de cada niño [15].

Al aprovechar las capacidades de la realidad virtual, aspiramos a mejorar la calidad de las evaluaciones, al mismo tiempo brindar una experiencia más atractiva y menos

estresante para los niños. Este enfoque puede mejorar la precisión de las evaluaciones, promover un mayor interés y participación en el proceso de evaluación, lo que beneficiará el desarrollo integral de los niños con discapacidad y generará un mayor entendimiento sobre sus necesidades específicas [17].

1.4 Alcances y Limitaciones

Este proyecto consistió en un prototipo funcional virtual con cuatro modelos de animales en movimiento, simulando el Test WISC. La plataforma elegida para realizar este prototipo fue el motor de videojuegos Unity, que ofrece tecnologías avanzadas muy útiles para la configuración espacial, el desarrollo de entornos visuales, la integración de elementos multimedia y la compatibilidad de dispositivos. El prototipo estaba pensado para que cada usuario pudiera tener una experiencia vivencial, y así llevar su proceso de manera independiente, haciendo que las pruebas fueran capturadas, medidas y estandarizadas con datos cruciales; esto permitió a los profesionales interpretar los resultados y entender los problemas que los niños enfrentan.

Este proyecto estuvo dirigido a los niños como usuarios objetivo, ya que se pretendía fomentar el desarrollo psicométrico a través de un sistema interactivo basado en gestos corporales. Además, el proyecto estuvo orientado para su uso dentro del Instituto para Niños Ciegos y Sordos, que es un aliado de la Universidad Javeriana y del semillero de Problemas de Computación, donde nació este proyecto. El Instituto jugó un papel importante al proporcionar apoyo y recomendaciones para la realización de este prototipo.

2 Marco Conceptual

2.1 Marco de Referencia

2.1.1 Áreas Temáticas

De acuerdo con el sistema de clasificación computacional ACM, las áreas temáticas que abarca el proyecto se muestran en la **Figura 1**.

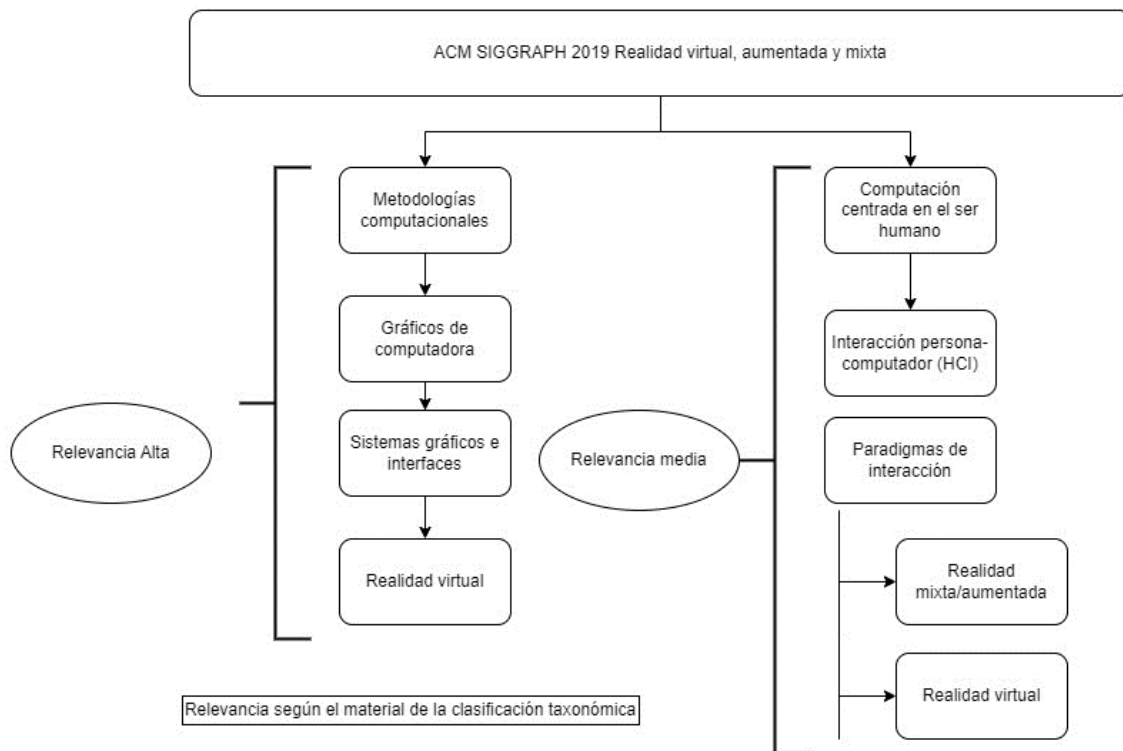


Figura 1: clasificación computacional ACM [18]

2.1.2 Marco Teórico

La realidad virtual ha sido estudiada desde principios del siglo XX. En el año 1935 se encontró la primera referencia específica de lo que hoy se conoce como gafas de Realidad Virtual (RV) [1]. Las primeras gafas de Realidad

virtual fueron creadas en 1960 por Morton Heiling y por eso se le conoce como el padre de la realidad virtual; Las llamó Sensorama y no era un videojuego o algo relacionado a los computadores, esto ofrecía más bien una experiencia cinematográfica inmersa multisensorial en la que se podía experimentar la sensación que produce manejar una moto [2]. La realidad virtual se ha consolidado como disciplina de estudio y ha adquirido relevancia en diversos ámbitos, desde la recreación de entornos ficticios hasta su aplicación en campos tan diversos como la medicina, la educación y la industria. En este contexto, resulta imperativo profundizar en la comprensión de las realidades que la realidad virtual abarca, dado que su impacto trasciende las fronteras de la tecnología para adentrarse en la esencia de la percepción y la experiencia humana. Para lograr el objetivo de este proyecto, es necesario definir ciertos conceptos claves. Entre esos se encuentra la definición de metaverso y las definiciones de los distintos tipos de realidades.

2.1.2.1 Actual reality/virtual continuum

Definición: Es un marco conceptual que describe la relación y la transición entre la realidad física (actual) y la realidad virtual. Esta noción refleja la idea de que las experiencias humanas pueden existir en un espectro que abarca desde la realidad física tangible y objetiva hasta la realidad virtual completamente digital y simulada [22]. Este marco proporciona una manera de conceptualizar cómo las tecnologías de la realidad extendida (XR) afectan la percepción y la participación de las personas en entornos virtuales [4] (Un resumen de estos conceptos se encuentra en la **Figura 2** extraída del texto de Farshid).

En este continuum, se pueden identificar varios niveles o puntos intermedios que representan diferentes grados de inmersión, interacción y digitalización. Estos niveles incluyen:

- **Realidad Física (RF):** Este extremo del continuum se refiere a la realidad tal como la experimentamos en el mundo físico. Las

experiencias y los objetos son tangibles y existen en el entorno físico [24].

- **Realidad Aumentada (RA):** En este punto intermedio, se incorporan elementos virtuales o digitales en la realidad física. La información virtual se superpone o se mezcla con el mundo real a través de dispositivos como gafas de realidad aumentada [23]. Ejemplos de juegos serios en este nivel incluyen aplicaciones educativas que superponen información histórica o científica sobre el mundo real, mejorando la experiencia de aprendizaje.
- **Realidad Mixta (RM):** La realidad mixta representa un nivel de integración más profunda entre la realidad física y la virtual. Los objetos virtuales pueden interactuar con objetos físicos, y viceversa, lo que permite una mayor coexistencia e interacción entre ambos mundos. Un ejemplo de juego serio en este nivel es una aplicación de entrenamiento médico, donde los estudiantes pueden practicar cirugías en un entorno que combina herramientas físicas reales con visualizaciones virtuales detalladas.
- **Realidad Virtual (RV):** En el extremo opuesto del continuum, la realidad virtual es completamente digital y simulada. Los usuarios se sumergen en entornos virtuales y experimentan una realidad que es independiente de la realidad física circundante; frecuentemente a través de dispositivos de realidad virtual como auriculares [5]. Ejemplos de juegos serios en RV incluyen simuladores de vuelo para la formación de pilotos y entornos virtuales para la rehabilitación física, donde los pacientes realizan ejercicios en un mundo simulado.
- **Virtualidad:** Se refiere a la capacidad de existir o interactuar en un entorno digital o conceptual en lugar del mundo físico. Puede abarcar desde experiencias inmersivas en mundos virtuales, hasta la representación de conceptos abstractos en un contexto digital. El término se utiliza en una variedad de campos y contextos para describir la transición desde lo físico a lo digital y lo conceptual [25].

Un ejemplo de juego serio en este contexto es una simulación de gestión empresarial, donde los usuarios pueden aprender sobre administración y toma de decisiones en un entorno completamente virtual.

Esta conceptualización es fundamental en el diseño y desarrollo de tecnologías de realidad extendida (XR) y en la comprensión de cómo estas tecnologías afectan la percepción y la interacción humana. A medida que la tecnología avanza, la ubicación de un usuario en este continuum puede cambiar, lo que tiene implicaciones significativas para diversas aplicaciones, desde el entretenimiento hasta la educación y la simulación.

Realidad	Realidad aumentada	Realidad virtual	Realidad Mixta	Virtualidad Aumentada	Virtualidad
El mundo real que experimentamos con todos nuestros sentidos	El mundo real que experimentamos con todos nuestros sentidos	Una representación completa y digital del mundo real	La introducción de elementos posibles en un mundo verdadero	La introducción de elementos reales en un mundo posible	Un mundo imaginario que en su mayoría sigue reglas del mundo real
Una casa de verdad	Una aplicación de bienes raíces	Proporciona detalles de una casa de verdad	Una imagen 3D de un mueble real	Una vista virtual de una casa real	simulación de muebles diferentes virtual o nuevo, en una casa real
Concepto clave	Concepto clave	Concepto clave	Concepto clave	Concepto clave	Concepto clave
Co-presencia física de personas y objetos	Agregar utilidad a co-presencia física	Habilitar percepción de presencia e inmersión completa	Adaptación de escenarios reales	Participación en escenarios posibles	Visión de un mundo completamente diferente
REAL			POSIBLE		
Realidad actual continua			Realidad virtual continua		

Figura 2: Tipos de Realidades computacionales

2.1.2.2 Evaluaciones psicométricas

Definición: Se refiere a un conjunto de pruebas y herramientas diseñadas para medir y evaluar diferentes aspectos de las capacidades cognitivas, emocionales, comportamentales y otras características de una persona. Estas evaluaciones se utilizan en áreas como la psicología, la educación, la psiquiatría, los recursos humanos y la investigación [7].

2.1.2.3 Test WISC

Definición: El Test de Inteligencia para niños de Wechsler, conocido como WISC-IV, es una evaluación psicométrica ampliamente utilizada para medir

la inteligencia en niños y adolescentes de edades comprendidas entre los seis (6) y los dieciséis (16) años. Este test se basa en modelos neuropsicológicos de procesamiento de información y ha evolucionado para proporcionar una evaluación más completa y actualizada de las capacidades cognitivas de los individuos en este grupo de edad. El WISC-IV se compone de una serie de subpruebas que evalúan habilidades cognitivas como la comprensión verbal, el razonamiento perceptual, la memoria de trabajo y la velocidad de procesamiento. Estas subpruebas se utilizan para calcular índices que reflejan diferentes aspectos de la inteligencia. Su uso es fundamental en la psicología, la educación y la investigación; para comprender las fortalezas y debilidades cognitivas de los niños y adolescentes, y guiar decisiones en contextos educativos y clínicos [8].

- **Cancelación (CA):** La subprueba de Cancelación del WISC-IV es una medida específica diseñada para evaluar la velocidad de procesamiento, la atención selectiva y la coordinación visomotora. En esta tarea se pide a los niños que busquen y marquen símbolos específicos en una página llena de distractores en un tiempo limitado; la subprueba de cancelación proporciona información valiosa sobre la capacidad de un niño para concentrarse y realizar tareas rápidamente y con precisión, mientras filtra información irrelevante. Esta habilidad es crucial para el éxito académico y el manejo de tareas diarias que requieren atención sostenida y selectiva. Evaluar esta habilidad puede ayudar a identificar dificultades en la atención y el procesamiento rápido de información, áreas que son esenciales para el aprendizaje y el desempeño en la escuela [26].

2.1.2.4 Diversidad sensorial

Definición: Se refiere a la variabilidad o diferencia en la percepción y el procesamiento de la información sensorial entre individuos. Cada persona experimenta el mundo a través de sus sentidos (como la vista, el oído, el tacto, el olfato y el gusto), y la diversidad sensorial reconoce que no todos procesan la información sensorial de la misma manera. Esta variabilidad

puede deberse a diferencias en las capacidades sensoriales, las preferencias sensoriales o las diferencias en la percepción sensorial [27].

2.1.2.5 Gestos sensoriales

Definición: Se refieren a las respuestas y expresiones no verbales, las cuales pueden ser utilizadas para comunicarse o manifestar sus experiencias sensoriales. Estos gestos pueden ser una forma importante de expresión y comunicación para niños que pueden tener dificultades con el lenguaje verbal o la comunicación tradicional [9].

2.1.2.6 Herramientas de desarrollo

Definición: Se refiere a un objeto diseñado para facilitar la realización de una tarea/actividad. En este caso, Unity se ha seleccionado como herramienta principal para desarrollar el videojuego, se puede observar en la **Figura 3**. Unity es una plataforma de desarrollo que permite crear y ampliar juegos, aplicaciones y experiencias 3D en tiempo real para las industrias de entretenimiento, automóviles, cine, arquitectura y mucho más.[19]

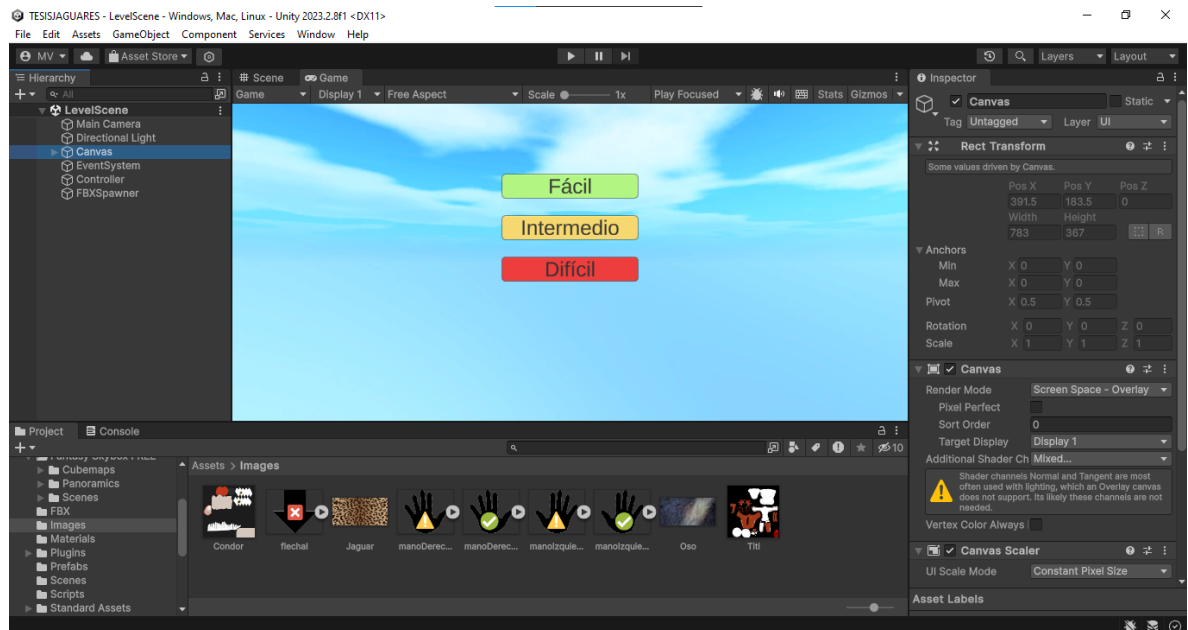


Figura 3: Captura de pantalla donde se ve el editor de Unity con los archivos del proyecto

2.1.3 Trabajos Similares

2.1.3.1 Video Juego Interactivo Mediante SDK Kinect 1.6 para Apoyar la Educación Básica Primaria de Niños Entre 5 a 10 Años de Edad. [1]

Este proyecto aborda la deserción escolar en Colombia mediante una herramienta tecnológica. Consiste en un videojuego interactivo que utiliza Kinect para motivar a los niños a participar en actividades de matemáticas, fomentando el aprendizaje mediante el movimiento de manos. El proyecto es adaptable y busca estimular el interés por la educación en niños colombianos.

Este proyecto se asemeja al nuestro, ya que ambos utilizamos tecnología interactiva para mejorar la experiencia educativa de los niños. Aunque nuestro enfoque es la evaluación psicométrica vivencial en niños con discapacidad, compartimos la visión de hacer que la educación sea más atractiva y efectiva a través de la tecnología. Esto puede tener un impacto positivo en el desarrollo integral de los niños y en su entusiasmo por aprender.

2.1.3.2 El uso del software educativo Cuadernia en el proceso de enseñanza - aprendizaje y en el rendimiento académico de la matemática de los estudiantes del 5to. año de secundaria de la institución educativa nº 5143 escuela de talentos. callao 2015 [11]

Este proyecto se enfoca en evaluar el impacto del uso del software educativo "Cuadernia" en el proceso de enseñanza-aprendizaje y el rendimiento académico de estudiantes de quinto (5to) año de educación secundaria. La investigación utilizó un enfoque cuantitativo y un diseño cuasiexperimental, involucrando a dos grupos de estudiantes, uno que utilizó el software y otro que no lo utilizó.

Este proyecto guarda similitudes con el nuestro, en términos de evaluar el impacto de la tecnología educativa en el proceso de aprendizaje. Aunque se centra en el software "Cuadernia" y el rendimiento académico en

matemáticas en estudiantes de secundaria, comparte con nuestro proyecto la búsqueda de mejorar la experiencia educativa y los resultados académicos mediante el uso de tecnología educativa.

De este proyecto, podemos obtener valiosas lecciones para la fase de pruebas piloto del nuestro. En particular, podemos aprender cómo diseñar grupos de control y experimentales, así como la recopilación y análisis de datos cuantitativos relacionados con el impacto de la tecnología en la educación. De esta manera nos servirán como guía para la planificación y ejecución de las pruebas, una vez que hayamos desarrollado el proyecto.

2.1.3.3 Uso del kinect en los videojuegos para la concientización sobre el cuidado del agua [12]

Este proyecto se enfoca en crear un videojuego para concientizar sobre el cuidado del agua. Utiliza la tecnología Kinect para desarrollar una interfaz háptica y se apoya en el lenguaje de programación C# y el motor de videojuegos Unity.

Este proyecto se asemeja al nuestro en términos de utilizar tecnología avanzada como el Kinect, para crear una experiencia interactiva. Aunque su enfoque es la concientización sobre el cuidado del agua, compartimos la utilización de tecnología interactiva y motores de videojuegos para fines educativos.

De este proyecto en particular podemos observar cómo se integran las herramientas Kinect y Unity para la creación de un proyecto, mismas herramientas que utilizaremos en nuestro prototipo.

2.1.3.4 Desarrollo de una aplicación web para optimizar el proceso del test psicológico basada en la Escala de Inteligencia de Wechsler para niños - V (WISC V) en el área de psicología de un hospital [13]

En este proyecto desarrollan una herramienta web para agilizar y mejorar la evaluación del WISC. El proyecto adopta una metodología adecuada para el desarrollo e implementación, generando un producto multiplataforma,

eficiente y escalable. Los especialistas pueden utilizar esta herramienta para facilitar la evaluación del test psicológico WISC.

Este proyecto guarda similitudes con el nuestro en términos de utilizar tecnología para mejorar los procesos de evaluación. Ambos proyectos buscan automatizar y optimizar la aplicación de pruebas psicométricas. Aunque se enfoca en el WISC-V en un contexto hospitalario, comparte con el nuestro la idea de utilizar tecnología para hacer que las evaluaciones sean más eficientes y precisas.

En la comparación de los proyectos mencionados con el nuestro, es evidente que todos comparten el objetivo de mejorar procesos educativos y evaluativos a través de la tecnología. Sin embargo, existen diferencias clave en los enfoques y las aplicaciones específicas de cada uno. Todos utilizan tecnología interactiva para mejorar la experiencia de los usuarios. Nuestro proyecto y el de Tangarife & Nieto Acevedo [11] utilizan Kinect para fomentar la participación de los niños a través del movimiento de manos. La diferencia radica en que nuestro enfoque está en la evaluación psicométrica, mientras que el suyo se centra en actividades educativas de matemáticas.

Al igual que Liñán [11], nuestro proyecto busca evaluar el impacto de la tecnología en el proceso de enseñanza-aprendizaje. No obstante, Liñán se enfoca en el rendimiento académico en matemáticas en estudiantes de secundaria, mientras que el nuestro se orienta hacia el desarrollo psicométrico de niños con discapacidades sensoriales. El proyecto de Conde [12] tiene como objetivo la concientización sobre el cuidado del agua utilizando Kinect y Unity, tecnologías que también usamos. La diferencia principal es el enfoque temático: nosotros buscamos evaluar habilidades psicométricas, mientras que Conde se centra en la educación ambiental.

Nuestro proyecto y el de Capcha [13] comparten la intención de optimizar procesos evaluativos. La ventaja de nuestro sistema, es su enfoque en una experiencia vivencial y adaptada a niños con discapacidades sensoriales, lo cual no es el foco principal del proyecto de Capcha, que se orienta hacia la escalabilidad y la eficiencia en un entorno hospitalario.

En cuanto al diseño experimental y metodología para evaluar el impacto de la tecnología, los proyectos de Liñán [11] y Tangarife & Nieto Acevedo [12] nos dan valiosas lecciones. Estos elementos son cruciales para la fase de pruebas piloto de nuestro proyecto, asegurando una evaluación robusta y cuantitativa de los resultados.

Es importante resaltar que aunque nuestro sistema está inspirado en el Test WISC IV y pretende evaluar la psicometría de los niños, no es el Test WISC. Nuestro prototipo se centra en proporcionar una experiencia interactiva y vivencial que facilite la detección temprana de problemas psicométricos y promueva el desarrollo integral de los niños con discapacidades sensoriales. Aunque compartimos elementos tecnológicos y objetivos con otros proyectos, nuestro enfoque único en la evaluación psicométrica vivencial y la adaptación para niños con discapacidades sensoriales, destaca la relevancia y la innovación de nuestro trabajo dentro del campo de la tecnología educativa y la psicometría.

3 Metodología

3.1.1 Tipo de estudio

El proyecto es un estudio exploratorio que pretende ser una fase preliminar (prototipo) para crear una herramienta peculiar que permita hacer evaluaciones psicométricas a partir de un sistema interactivo gestual, y pueda usarse tanto para detectar los problemas en los niños, como en un futuro en estudios más profundos del tema.

Para el desarrollo del prototipo se seguirá la metodología scrum en donde a través de Sprints se permita ir generando valor al proyecto, esto con la finalidad de lograr los objetivos propuestos, partiendo desde la investigación hasta la elaboración del prototipo. Como parte integral de esta investigación exploratoria, se utilizarán cuestionarios y observación directa; estas técnicas de recolección de información se revelan como instrumentos fundamentales para profundizar en el entendimiento del problema abordado y respaldar el desarrollo del sistema interactivo gestual.

3.1.2 Lista de actividades

3.1.2.1 Objetivo 1: Estudiar cómo funciona el test WISC para identificar patrones de comportamientos en los niños y acoplarlos a sus necesidades.

Para este objetivo se realizarán las siguientes actividades:

- Tomar información de páginas o bases de datos oficiales de centros de estudio sobre el funcionamiento e impacto del test WISC.
- Acoplar dicha información a las necesidades específicas e individuales.

3.1.2.2 Objetivo 2: Diseñar un sistema basado en un sistema de selección preexistente que permita la realización de una experiencia vivencial en los niños.

Para este objetivo se realizarán las siguientes actividades:

- Seleccionar una estrategia viable para la realización de un sistema de evaluación gestual acorde a los niños con las condiciones mencionadas.
- Diseñar la arquitectura del prototipo para el entorno virtual.

3.1.2.3 Objetivo 3: Implementar el diseño utilizando tecnologías de realidad aumentada.

Para este objetivo se realizarán las siguientes actividades:

- Seleccionar tecnologías de realidades aumentadas que se puedan usar para implementar el sistema diseñado.
- Implementar la estructura del prototipo en la tecnología de realidad aumentada seleccionada.
- Programar e implementar el sistema de evaluación en el prototipo.
- Ajustar las estrategias de evaluación con respecto al test WISC en el prototipo.

3.1.2.4 Objetivo 4: Validar el prototipo implementado.

Para este objetivo se realizarán las siguientes actividades:

- Hacer las respectivas pruebas para encontrar los errores y problemas que existan en el prototipo.
- Hacer las respectivas pruebas para validar la accesibilidad del prototipo en los niños con condiciones visuales y auditivas visuales.
- Hacer las respectivas pruebas para evaluar el objetivo principal si cumple con el ámbito educativo para las personas que usen el prototipo.
- Hacer las respectivas pruebas para evaluar el funcionamiento del sistema con respecto al cumplimiento de la experiencia vivencial.

4 Análisis

4.1 Actores del sistema

Dentro de las personas del Instituto para Niños Ciegos y Sordos del Valle del Cauca, se identificaron dos usuarios directamente con el sistema que son el profesional encargado y el jugador/niño.

4.1.1 Jugador

El JUGADOR/NIÑO es definido como cualquier niño que esté incorporado en el Instituto para Niños Ciegos y Sordos del Valle del Cauca y que use el videojuego.

4.1.2 Profesional Encargado

El PROFESIONAL ENCARGADO se define como el encargado de la educación y cuidado de los niños en el Instituto. El actor debe guiar al usuario JUGADOR en cómo realizar el juego de la manera adecuada, así que el sistema debe permitirle crear nuevos jugadores, seleccionar jugadores ya creados, ver resultados; entre otras funciones. Una de las funciones importantes es seleccionar la dificultad del sistema evaluativo de la captura de animales que se enfrentará el jugador según el criterio médico (Fácil, Intermedio, Difícil). Las velocidades del sistema en cada nivel de dificultad se acordaron con el director del Instituto y el director del trabajo de grado.

4.2 Requisitos

4.2.1 Requisitos funcionales

A continuación se muestra la lista completa de los requisitos funcionales del proyecto:

- RF-01 Interfaz de usuario:
 - Descripción: El sistema debe permitir la interacción con una interfaz entre usuario y sistema.
- RF-02 Menú inicial:

- Descripción: Al iniciar el juego, el sistema debe mostrar un menú para darle acceso a la aplicación.
- RF-03 Puntaje:
 - Descripción: El sistema debe mostrar mediante la interfaz un puntaje del sistema de evaluación psicométrico completado por el usuario jugador.
- RF-04 Seleccionar Jugador:
 - Descripción: El sistema debe permitir al usuario profesional encargado seleccionar el jugador por medio de un ID.
- RF-05 Contraste de elementos y videojuego:
 - Descripción: Los elementos visuales del sistema deben tener alto contraste para facilitar la visualización.
- RF-06 Dificultades del juego:
 - Descripción: Sin importar el modo escogido, el usuario profesional encargado debe poder seleccionar la dificultad del juego antes de empezar.
- RF-07 Creación nuevo jugador:
 - Descripción: El sistema debe contar con una interfaz que permita al usuario profesional encargado ingresar la información de un nuevo jugador.
- RF-08 Resumen de partida:
 - Descripción: Al finalizar cada modo del juego(niveles) el sistema debe mostrar un resumen del puntaje obtenido por el usuario jugador.
- RF-09 Visualización de resultados:
 - Descripción: El sistema debe permitir al usuario profesional encargado visualizar los resultados de las partidas realizadas por cada jugador.
- RF-10 Captura de animales:
 - Descripción: El sistema debe generar el sonido respectivo del animal que el usuario jugador capturó.
- RF-11 Captura por el lado izquierdo/derecho:

- Descripción: El sistema debe registrar y medir si la captura de los animales se realiza por el lado izquierdo o derecho, permitiendo así una evaluación detallada de la coordinación visomotora del usuario.
- RF-12 Medición de capturas correctas e incorrectas:
 - Descripción: El sistema debe registrar y medir el número de capturas correctas e incorrectas realizadas por el usuario, proporcionando datos precisos sobre su desempeño durante el juego.

4.2.2 Requisitos no funcionales

A continuación se muestra la lista completa de los requisitos no funcionales del proyecto:

- RNF-01 Usabilidad:
 - Descripción: La interfaz debe ser intuitiva y fácil de navegar, con instrucciones claras y accesibles para los jugadores de diferentes edades.
- RNF-02 Compatibilidad:
 - Descripción: El videojuego debe funcionar correctamente en los principales sistemas operativos que soporten Kinect, como Windows 10 o versiones superiores, y no debe requerir software adicional que no esté ampliamente disponible.
- RNF-03 Mantenibilidad:
 - Descripción: El código y los componentes del sistema deben estar bien documentados para facilitar el mantenimiento y las futuras actualizaciones
- RNF-04 Escalabilidad:
 - Descripción: El sistema debe ser capaz de manejar una mayor complejidad en los niveles de juego sin una degradación significativa del rendimiento.

4.3 Diagrama de casos de Uso

Los diagramas de casos de uso son para representar los actores externos que interactúan con el sistema de información y a través de que funcionalidades (casos de uso o requisitos funcionales) se relacionan. Dicho de otra manera, muestra de manera visual las distintas funciones que puede realizar un usuario (más bien un tipo de usuario) de un sistema de información.[20]

En la **Figura 4** se observa el diagrama resultante que describe la interacción de los actores con el sistema. Además, se puede observar una secuencia en las acciones que se toman, que de igual forma se pueden observar en la interfaz gráfica; el actor que se encarga de darle más manejo a cosas técnicas (seleccionar jugador, seleccionar dificultad, crear nuevo jugador, etc) es el Profesional Encargado, es decir, dicho actor posee la mayor parte de la interacción directa con el sistema. Por otro lado, el actor Niño/Jugador realiza una acción directa con el sistema que es jugar.

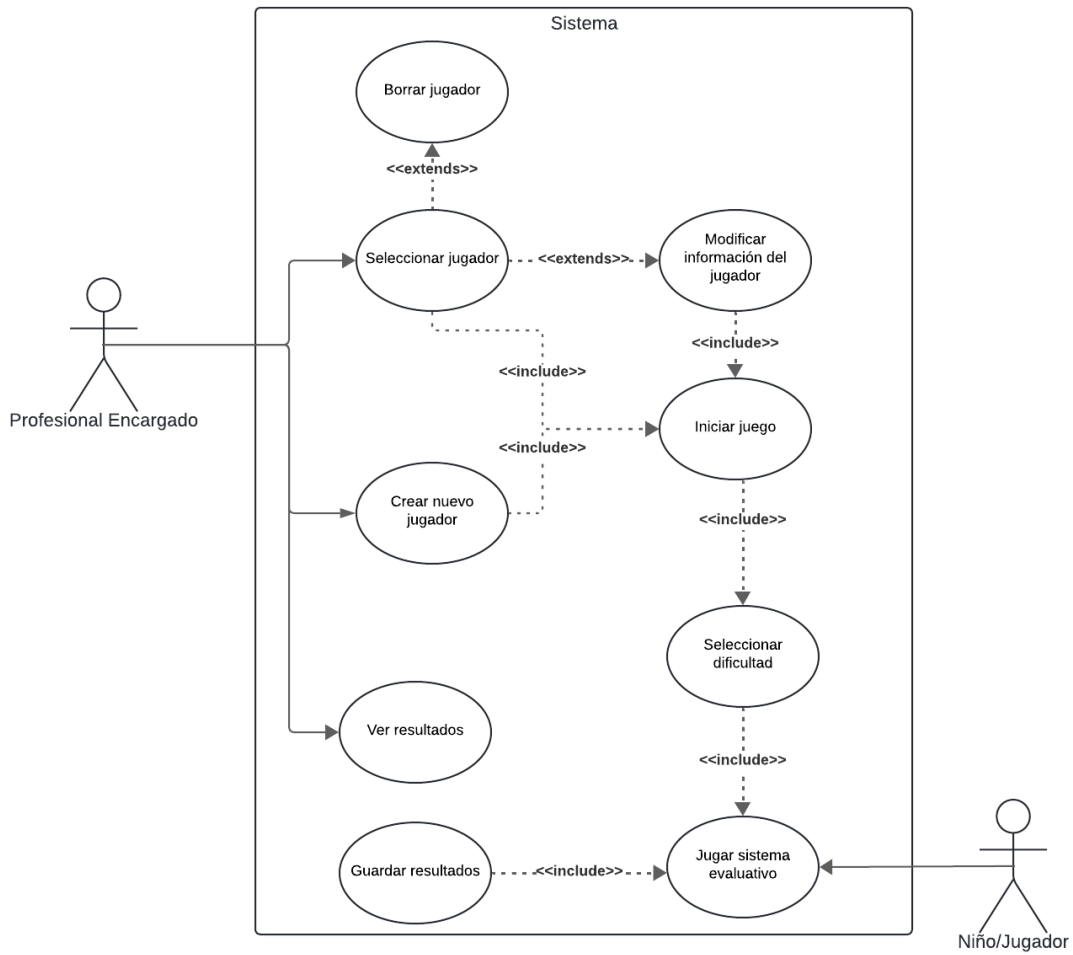


Figura 4: Diagrama de casos de uso que representa los actores que interactúan con el sistema.

5 Diseño

5.1 Descripción del modelo del sistema

En esta sección se especificará el diseño del proyecto, detallando cómo está constituido y estructurado mediante diversos diagramas y modelos. A través de diagramas de clases, diagramas de secuencias, diseño de interfaces, modelo de datos y diseño de algoritmos; se podrá visualizar y entender desde otra perspectiva el funcionamiento y la estructura del sistema.

5.2 Implementación de Elementos del Test WISC en el Sistema

En el desarrollo de este videojuego, se incorporará una serie de elementos inspirados en las subpruebas del Test de Inteligencia Wechsler para Niños (WISC), específicamente en la subprueba de Cancelación (2.1.2.3).

Descripción de la Implementación

1. **Entorno Interactivo y Estímulos Visuales:** El videojuego presentará a los usuarios un entorno en el que diversos animales caerán desde la parte superior de la pantalla. Estos animales actuarán como estímulos visuales, similares a los símbolos en la subprueba de Cancelación. A lo largo de cada sesión de juego, el jugador deberá identificar y capturar estos animales usando esferas virtuales controladas por los movimientos de sus manos, detectados a través del sensor Kinect.
2. **Procesamiento Visual y Atención Selectiva:** Al igual que en la subprueba de Cancelación, donde el niño deberá seleccionar y marcar ciertos símbolos específicos entre muchos otros distractores, en este videojuego el jugador deberá enfocarse en capturar ciertos animales mientras ignora otros que no son objetivos o que podrían actuar como distractores. Este proceso requerirá que el jugador mantenga un alto nivel de atención selectiva y procesamiento visual rápido, capacidades que también se evaluarán en la subprueba de Cancelación.

3. **Evaluación de la Velocidad de Procesamiento:** La velocidad con la que el jugador podrá identificar y capturar los animales será fundamental en la evaluación de su desempeño, similar a cómo se medirá en la subprueba de Cancelación del WISC. En el videojuego, se registrará la cantidad de animales capturados dentro de un tiempo determinado, y se evaluará cuántos de estos fueron capturados con la mano derecha y cuántos con la mano izquierda. Este enfoque permitirá medir no solo la rapidez, sino también la coordinación y la eficiencia del jugador.
4. **Coordinación Mano-Ojo y Respuesta Motora:** La utilización del Kinect en la detección de los movimientos de las manos agregará una dimensión adicional a la evaluación, integrando la coordinación mano-ojo y la respuesta motora en la tarea. Esta característica del videojuego permitirá analizar cómo el jugador combinará la percepción visual con la acción física para lograr los objetivos del juego, una habilidad esencial que también será implícitamente evaluada en la subprueba de Cancelación.

5.3 Diagrama de Clases

El diagrama de clases en la **Figura 5** se puede observar las principales clases del proyecto a desarrollar. "FBXSpawner" es la clase central que gestiona la aparición y desaparición de los animales, recibiendo información de "SphereJoint" y comunicándose con "FBXController" para controlar cada animal individualmente. "TextController" se encarga de registrar y mostrar los datos de los animales capturados, mientras que "TimeController" controla el temporizador del juego y coordina con "MenuController" para cambiar de escena al finalizar el tiempo. Adicionalmente, "PauseMenu" gestiona el menú de pausa y permite cambiar de escena durante el juego. En cuanto a la gestión de jugadores, "LoadEditPlayerInformation" maneja la selección y edición de jugadores, enviando los datos a "SaveUpdateDataPlayers" para su almacenamiento. "SaveDataGame"

se ocupa de guardar la información de las partidas, asegurando que todos los datos relevantes se mantengan actualizados y accesibles.

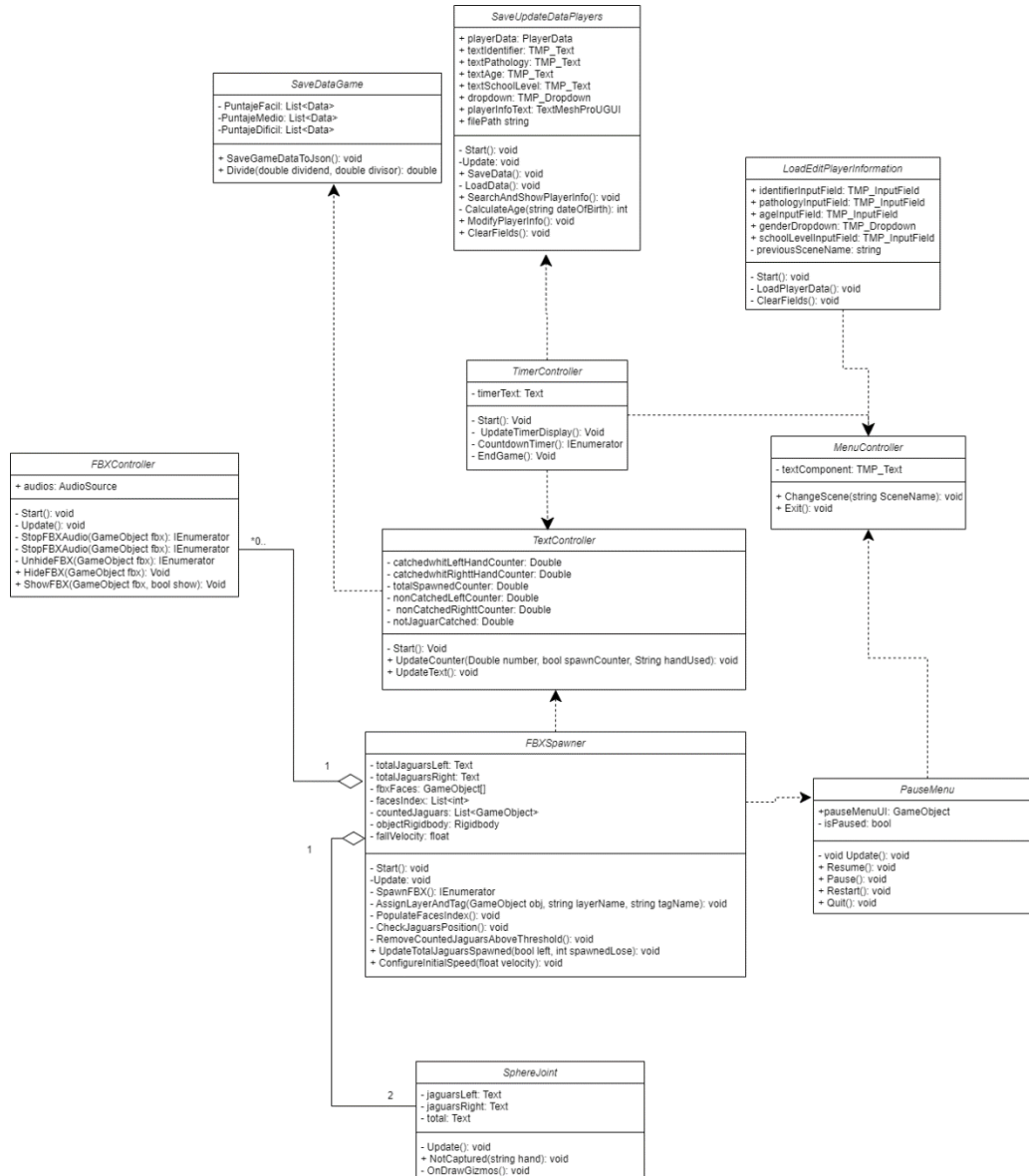


Figura 5: Diagrama de Clases: Muestra las principales clases del proyecto.

5.4 Diagrama de Secuencia

Los diagramas de secuencias describen cómo un grupo de objetos funcionan en conjunto. En los diagramas representados en las **Figuras 6, 7, 8, 9, 10, 11**.

5.4.1 Escena 1: Menú

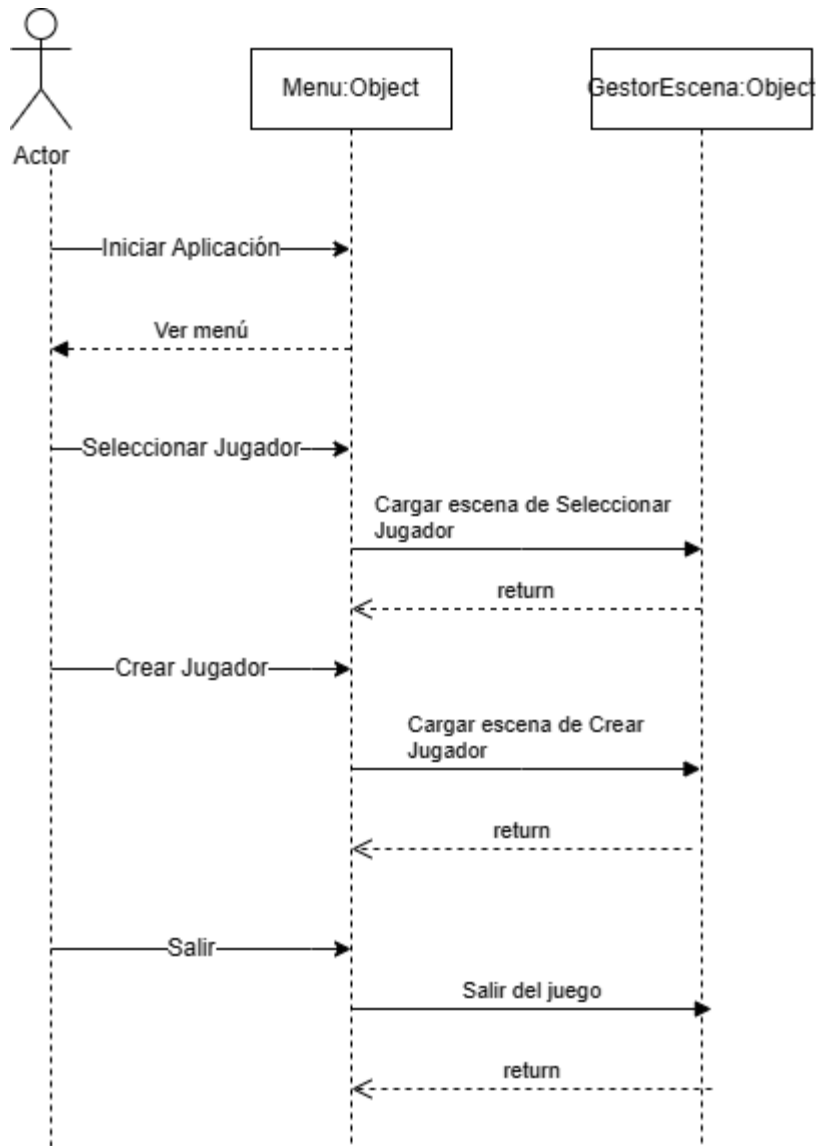


Figura 6: Diagrama de secuencia 1: Muestra la secuencia de ejecución para la escena menú.

En el diagrama presentado en la **Figura 6** se muestra la interacción de los objetos en la escena menú. Este diagrama muestra la secuencia en la que el actor, en este caso, el profesional encargado puede escoger 3 opciones: seleccionar jugador, crear jugador y salir.

5.4.2 Escena 2: Crear Jugador

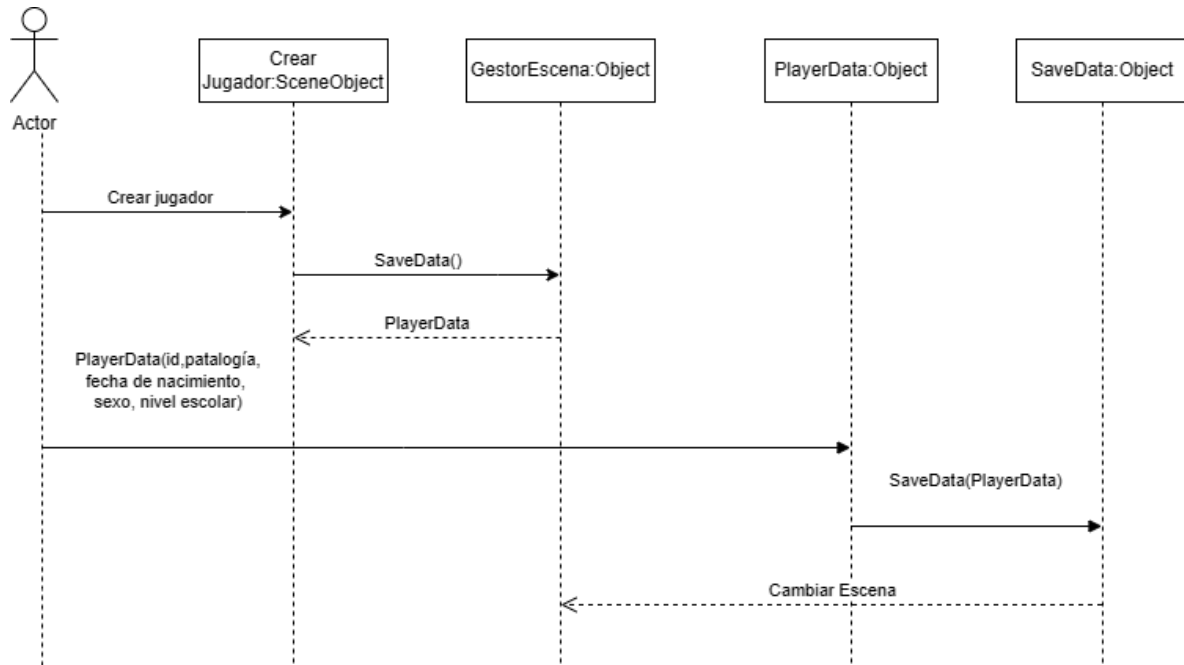


Figura 7: Diagrama de secuencia 2: Muestra la secuencia de ejecución para la escena 2, donde se llena el formulario de información del jugador y se guarda.

En el diagrama presentado en la **Figura 7** se muestra la interacción de objetos y escenas en la escena de crear jugador. Este diagrama muestra la secuencia en la que el actor, en este caso, el profesional encargado debe llenar el formulario con la información del jugador, el cual consta de 5 campos: Identificador, patología, fecha de nacimiento, sexo y nivel escolar. Esta información se crea en PlayerData donde se almacena toda la información y posteriormente en la clase SaveUpdateData y en la función SaveData() se guarda dicha información en la base de datos.

5.4.3 Escena 3: Seleccionar Jugador

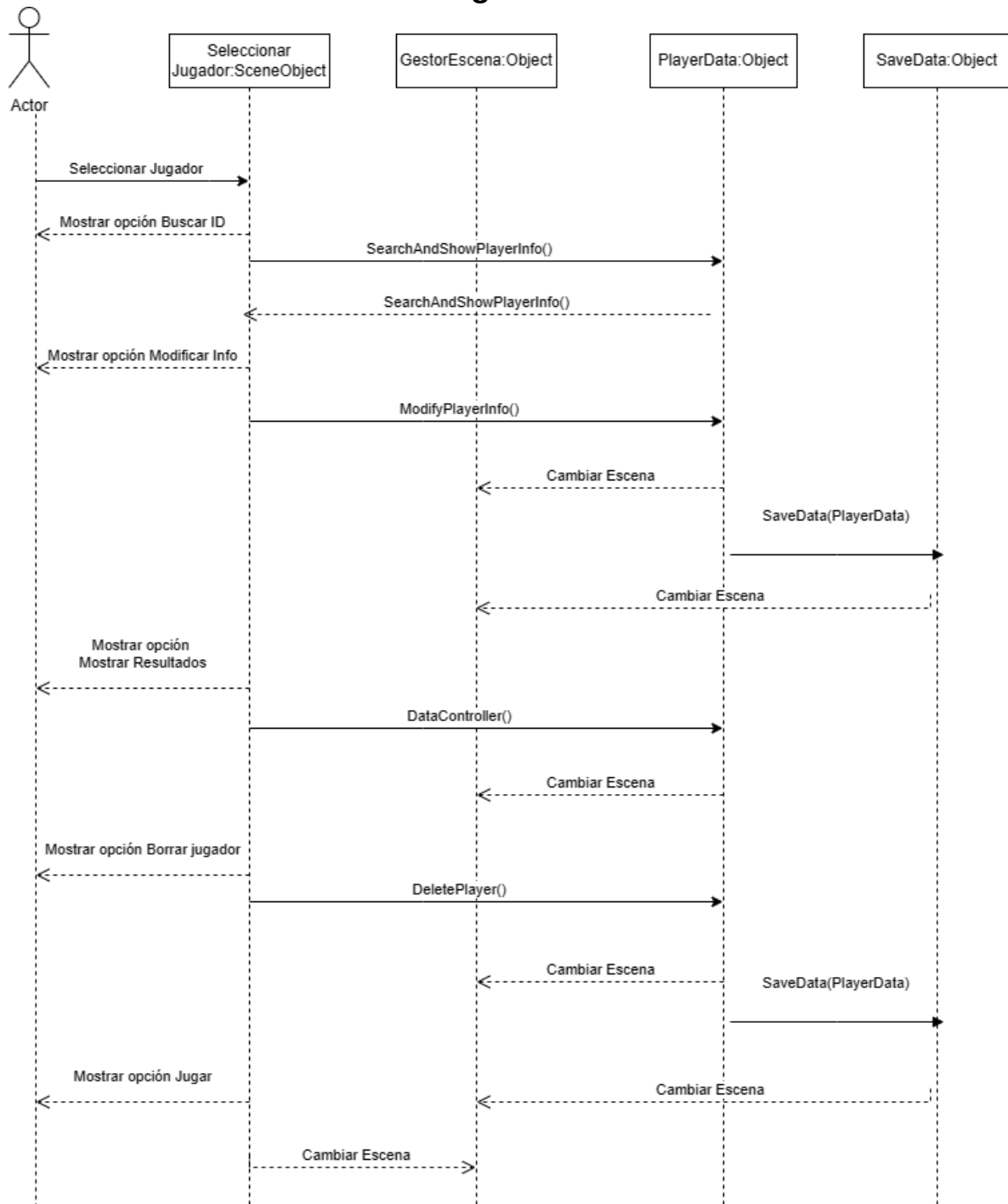


Figura 8: Diagrama de secuencia 3: Muestra la secuencia de ejecución para la escena 3, donde selecciona el jugador.

En el diagrama presentado en la **Figura 8** se muestra la interacción de objetos y escenas en la escena de seleccionar jugador. Este diagrama muestra la secuencia

en la que el actor, en este caso, el profesional encargado puede buscar la información del jugador por medio del ID, esto se hace en el método `SearchAndShowPlayerInfo()`, el cual busca el ID y muestra la respectiva información del jugador; en caso de no estar ese ID en la base de datos se muestra en la escena que no se encontró ninguna información con esa ID. El profesional encargado puede modificar la información del jugador por medio del ID, esto se hace en el método `ModifyPlayerInfo()`, el cual ya estando el ID y desplegada la información del jugador, lo llevará a la escena de `NewPlayerScene` con la información de dicho jugador; una vez modificado, podrá guardar la información modificada y esta quedará registrada en la base de datos. El profesional encargado puede borrar la información del jugador por medio del ID, esto se hace por medio de método `DeletePlayer()`, el cual ya estando el ID y desplegada la información del jugador se podrá borrar toda la información de la base de datos. El profesional encargado puede visualizar los resultados de las partidas de manera gráfica del ID seleccionado por medio `DataController` y en la escena `GraphResults`. Por último, el profesional encargado puede seleccionar la opción de jugar, el cual lo llevará a la escena `LevelScene`, para poder seleccionar el nivel respectivo.

5.4.4 Escena 4: Seleccionar Dificultad

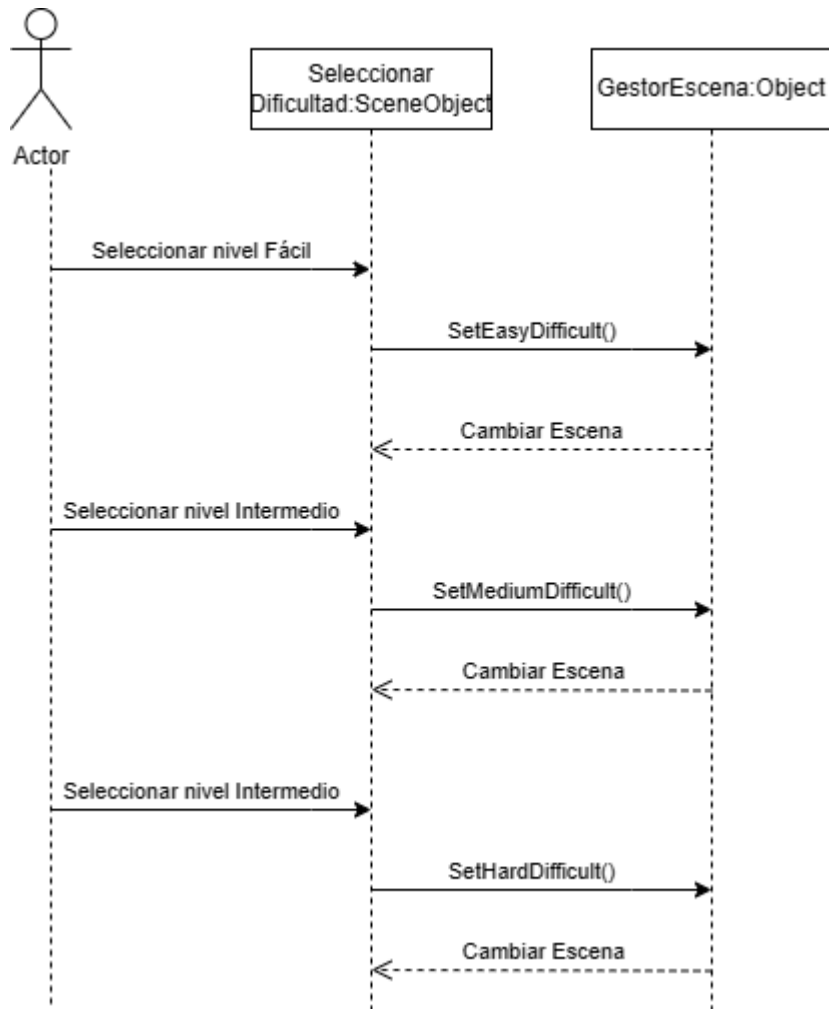


Figura 9: Diagrama de secuencia 4: Muestra la secuencia de ejecución para la escena 4, donde selecciona el jugador.

En el diagrama presentado en la **Figura 9** se muestra la interacción de objetos y escenas en la escena de seleccionar dificultad. Este diagrama muestra la secuencia en la que el actor, en este caso, el profesional encargado puede seleccionar la dificultad que va a afrontar el jugador, el cual se puede: Fácil, Intermedio, Difícil. Esto se hace por medio de los métodos: SetEasyDifficult(), SetMediumDifficult() y SetHardDifficult(). Una vez el profesional encargado seleccione el nivel, lo llevará a la escena del juego, por medio de ChangeScene().

5.4.5 Escena 5: Juego

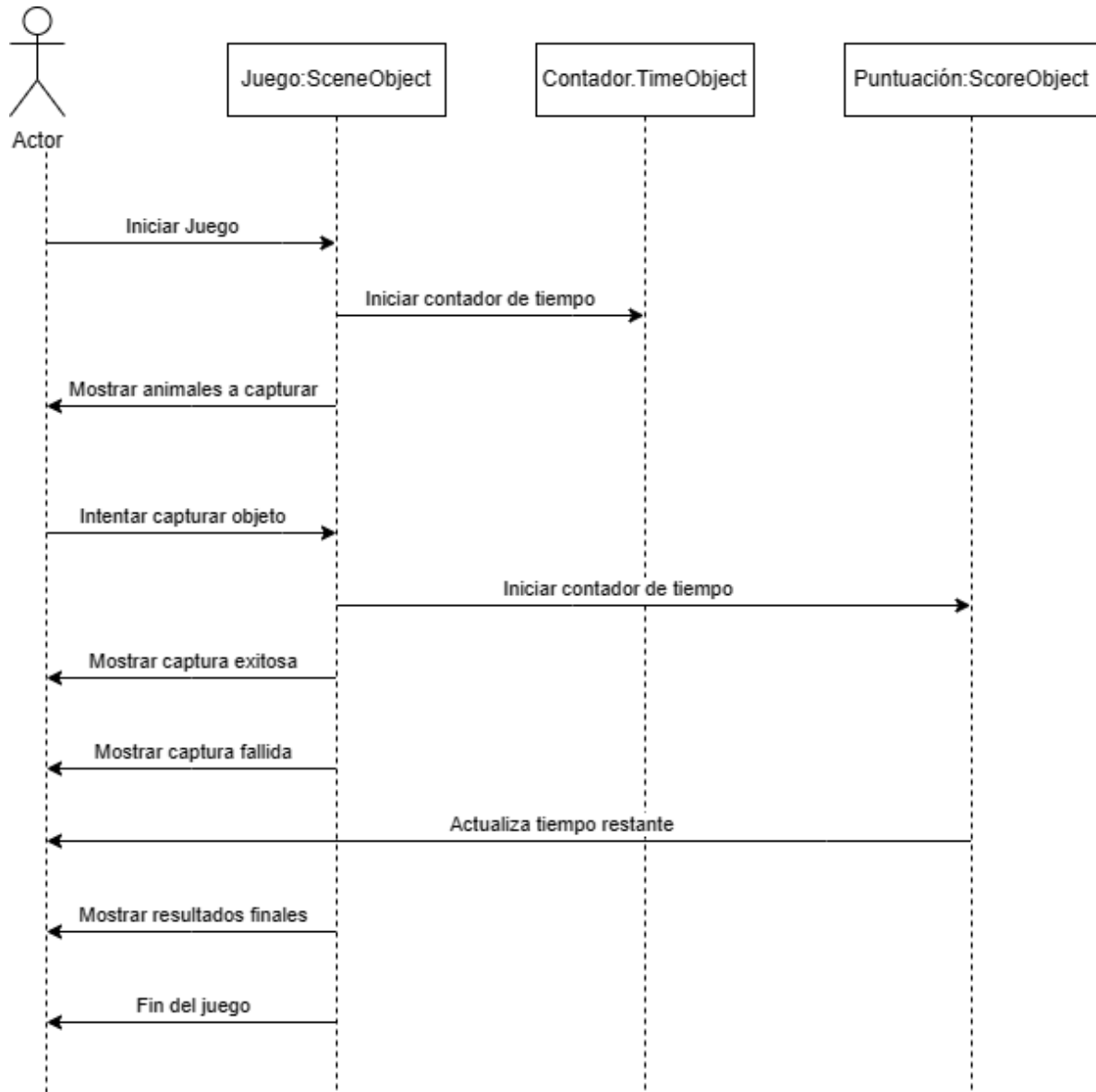


Figura 10: Diagrama de secuencia 5: Muestra la secuencia de ejecución para la escena 5, donde comienza el juego.

En el diagrama presentado en la **Figura 10** se muestra la interacción de objetos y escenas en una sesión de juego donde el objetivo es capturar animales. Este diagrama ilustra la secuencia de eventos y la comunicación entre diferentes componentes del sistema de juego, los cuales incluyen al actor, el objeto Juego:SceneObject, el Contador:TimeObject y el objeto Puntuación:ScoreObject. El

actor comienza el juego, lo que lleva al objeto Juego:SceneObject a iniciar el contador de tiempo al enviar una señal al objeto Timer que se maneja por medio de la clase TimerController(). Luego, los objetos muestran los animales que deben ser capturados. El actor intenta capturar un objeto, y el objeto Juego:SceneObject que está en la clase FBXSpawner(), el cual se encarga de spawnear los animales. Si la captura es exitosa, el objeto Juego:SceneObject muestra un contador, indicando que la captura fue exitosa, mientras que si la captura falla, muestra un contador indicando que la captura fue fallida. El objeto Timer actualiza el tiempo restante y envía esta información de vuelta al objeto Juego:SceneObject. Una vez terminado el tiempo, el objeto Juego:SceneObject muestra los resultados finales de la sesión de juego. Finalmente, el juego termina y el actor es notificado del final del juego. Este diagrama de secuencia muestra cómo se coordinan los diferentes componentes del sistema para llevar a cabo las acciones del juego, actualizando el tiempo y mostrando resultados según las interacciones del actor.

5.4.6 Escena 6: Resultado

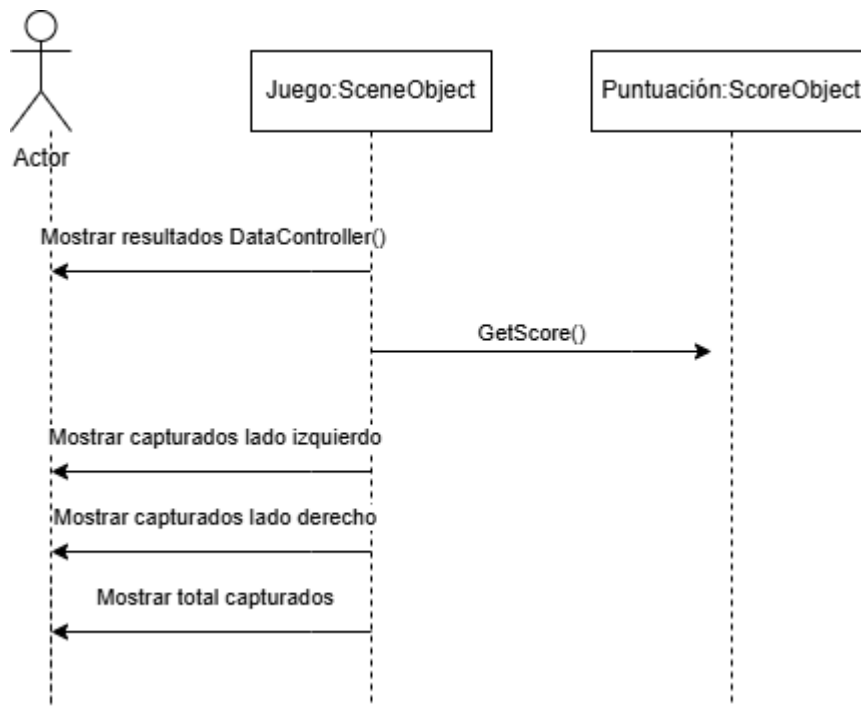


Figura 11: Diagrama de secuencia 4: Muestra la secuencia de ejecución para la escena 6, donde se pueden ver los resultados del juego.

En el diagrama presentado en la **Figura 11**, se muestra la interacción de objetos y escenas en una sesión de juego para mostrar los resultados de capturas. Este diagrama ilustra la secuencia de eventos y la comunicación entre diferentes componentes del sistema de juego, los cuales incluyen al actor (el jugador), el objeto Juego:SceneObject y el objeto Puntuación:ScoreObject. El jugador comienza solicitando mostrar los resultados a través del método DataController(). El objeto Juego:SceneObject, en respuesta, realiza una llamada al método GetScore() del objeto Puntuación:ScoreObject para obtener la puntuación actual. Una vez obtenida la puntuación, el objeto Juego:SceneObject procede a mostrar los animales capturados en el lado izquierdo, seguido de los animales capturados en el lado derecho, y finalmente muestra el total de animales capturados. Este diagrama de secuencia muestra cómo se coordinan los diferentes componentes del sistema para presentar los resultados de las capturas al actor, proporcionando una vista detallada de las capturas realizadas en cada lado y el total general.

5.5 Diseño de Interfaz

En este proyecto se realizaron las historias de usuarios utilizando la metodología Scrum. Comenzando por identificar los stakeholders y las necesidades que eran requeridas. Durante cada Sprint se fueron seleccionando historias, haciendo posible la división de subtareas y la implementación. Al finalizar cada Sprint se mostraban las funcionalidades desarrolladas, el cual recibieron un feedback por parte del director de tesis, Andrés Adolfo Navarro Newball y por parte del Instituto para Niños Ciegos y Sordos. Este proceso se repitió en ciclos iterativos para poder garantizar una entrega continua y adaptarse a las necesidades del proyecto.

El sistema general cuenta con siete interfaces donde puede interactuar el profesional encargado o el jugador. La primera Interfaz es el menú inicial, es la primera pantalla que se ve una vez el profesional encargado inicia la aplicación. Para cada escena se implementaron historias de usuario para seguir la metodología Scrum.

Para esta escena menú llevamos a cabo la realización de historias de usuario.

- Seleccionar jugador

Como profesional encargado quiero poder seleccionar un jugador para acceder a su información y ver sus resultados de partidas.

Criterios de aceptación:

- Debe haber una opción en el menú para seleccionar jugador.
- Debe ser posible ver y modificar la información del jugador buscado.
- Debe ser posible ver los resultados de manera gráfica de las partidas del jugador buscado.

- Crear jugador

Como profesional encargado quiero poder crear un nuevo jugador para registrar su información en el sistema.

Criterios de aceptación:

- Debe haber una opción en el menú para crear jugador.
- Al seleccionar esta opción, se debe mostrar un formulario para ingresar la información del nuevo jugador.
- El formulario debe incluir campos para el identificador, patología, fecha de nacimiento, sexo y nivel escolar del jugador.
- Después de ingresar la información, debe quedar registrado el jugador en la base de datos.

- Salir

Como profesional encargado quiero poder salir del juego cuando sea necesario

Criterios de aceptación:

- Debe haber una opción en el menú para salir del juego.

- Al seleccionar esta opción, se debe cerrar la aplicación.

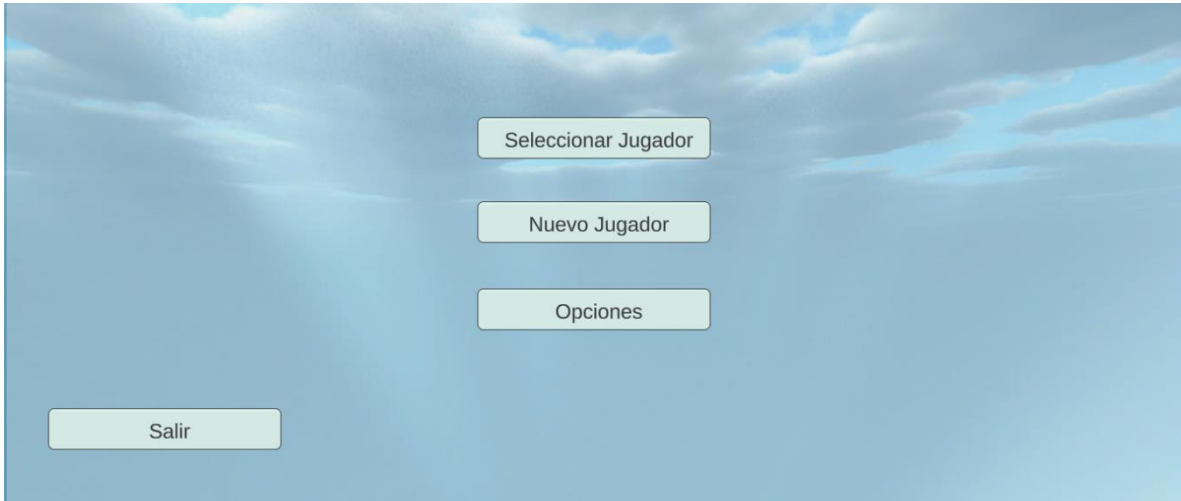


Figura 12: Menú Inicial del sistema

Como resultado al momento de diseñar en el editor de Unity el menú presentado en la **Figura 12**

Para esta escena crear jugador llevamos a cabo la realización de historias de usuario.

- Ingresar Información del jugador

Como profesional encargado, quiero poder ingresar la información de un nuevo jugador en el sistema.

Criterios de Aceptación:

- Debe proporcionar un formulario con los campos: identificador, patología, fecha de nacimiento, sexo y nivel escolar.
 - Debe ser posible ingresar valores válidos en cada uno de los campos del formulario.
 - El formulario debe ser intuitivo y fácil de completar.
- Guardar información del jugador en la base de datos.
Como profesional encargado quiero que la información ingresada del nuevo jugador se guarde en la base de datos del sistema.

Criterios de aceptación:

- Después de completar el formulario y hacer click en el botón de guardar, la información del jugador debe ser almacenada en la base de datos del sistema.

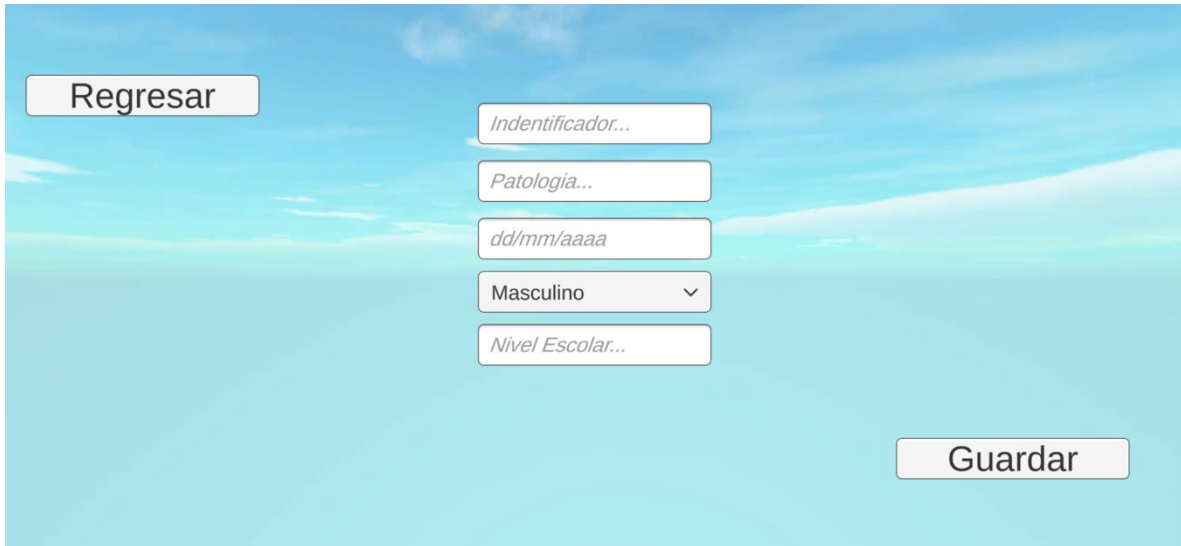


Figura 13: Formulario para nuevo jugador.

Como resultado al momento de diseñar en el editor de Unity el menú presentado en la **Figura 13**

Para esta escena seleccionar jugador llevamos a cabo la realización de historias de usuario.

- Buscar Información del jugador por Identificador.
Como profesional encargado, quiero poder buscar la información de un jugador utilizando su identificador.
Criterios de Aceptación:
 - Debe proporcionar un campo de búsqueda donde se pueda ingresar el identificador del jugador.
 - Al realizar la búsqueda, el sistema debe encontrar y mostrar la información del jugador correspondiente.
 - En caso de no encontrar al jugador con el ID ingresado, debe mostrar un mensaje que no lo encontró.

- Modificar información del jugador

Como profesional encargado, quiero poder modificar la información de un jugador existente en el sistema.

Criterios de aceptación:

- Después de buscar y encontrar la información del jugador por su ID, se debe proporcionar una opción para modificar dicha información.
- Al seleccionar esta opción, el sistema debe cargar la información del jugador en un formulario editable.
- Después de realizar las modificaciones necesarias, el usuario debe poder guardar los cambios y el sistema debe actualizar la información del jugador en la base de datos.

- Borrar jugador.

Como profesional encargado, quiero poder borrar la información de un jugador existente en el sistema.

Criterios de aceptación:

- Después de buscar y mostrar la información del jugador por su identificador, debe proporcionar una opción para borrar el jugador.
- Al borrar el jugador, el sistema debe borrarlo de la base de datos.

- Visualizar resultados de las partidas.

Como profesional encargado, quiero poder visualizar los resultados de las partidas de un jugador de manera gráfica.

Criterios de aceptación:

- Después de buscar y mostrar la información del jugador por su identificador, debe proporcionar una opción para visualizar los resultados de las partidas.
- Al seleccionar esta opción, el sistema debe mostrar gráficamente los resultados de las partidas del jugador, proporcionando información relevante para el análisis.

- Jugar.

Como profesional encargado, quiero poder seleccionar un jugador para comenzar a jugar.

Criterios de aceptación:

- Después de buscar y mostrar la información del jugador por su identificador, debe proporcionar una opción para jugar.
- Al seleccionar esta opción, el sistema debe llevar al profesional encargado a la escena de selección de nivel, donde pueda elegir el nivel en el que el jugador jugará.



Figura 14: Menú de seleccionar jugador.

Como resultado al momento de diseñar en el editor de Unity el menú de seleccionar jugador presentado en la **Figura 14**

Para esta escena seleccionar dificultad llevamos a cabo la realización de historias de usuario.

- Seleccionar dificultad fácil

Como profesional encargado, quiero poder seleccionar el nivel de dificultad fácil para el jugador.

Criterios de aceptación:

- Debe haber una opción en la interfaz para seleccionar la dificultad fácil.

- Al seleccionar esta opción, el sistema debe configurar el juego para la dificultad fácil..
- Después de seleccionar la dificultad, el sistema debe llevar al jugador a la escena del juego.
- Seleccionar dificultad intermedia

Como profesional encargado, quiero poder seleccionar el nivel de dificultad intermedia para el jugador.

Criterios de aceptación:

- Debe haber una opción en la interfaz para seleccionar la dificultad intermedia.
- Al seleccionar esta opción, el sistema debe configurar el juego para la dificultad intermedia.
- Después de seleccionar la dificultad, el sistema debe llevar al jugador a la escena del juego.
- Seleccionar dificultad difícil

Como profesional encargado, quiero poder seleccionar el nivel de dificultad difícil para el jugador.

Criterios de aceptación:

- Debe haber una opción en la interfaz para seleccionar la dificultad difícil.
- Al seleccionar esta opción, el sistema debe configurar el juego para la dificultad difícil.
- Después de seleccionar la dificultad, el sistema debe llevar al jugador a la escena del juego.



Figura 15: Seleccionar dificultad del jugador

Como resultado al momento de diseñar en el editor de Unity el menú de seleccionar dificultad presentado en la **Figura 15**

Para esta escena del juego llevamos a cabo la realización de historias de usuario.

- Iniciar juego y capturar animales

Como jugador, quiero poder iniciar el juego para capturar animales con mis manos usando realidad virtual con Kinect.

Criterios de Aceptación:

- Debe haber una opción para iniciar el juego, que active el contador de tiempo.
- Los animales (cóndor, tití, oso y jaguar) deben aparecer en la escena para ser capturados.
- La detección de manos debe ser precisa para permitir la captura de animales con las manos.
- Captura exitosa de animal con la mano derecha

Como jugador, quiero que el sistema registre una captura exitosa cuando capture un animal correctamente con mi mano derecha.

Criterios de aceptación:

- Al capturar un animal con la mano derecha en el lado correcto, el sistema debe contar la captura como exitosa.
- El sistema debe actualizar el contador de capturas exitosas y mostrar una indicación visual de éxito.
- Captura fallida de animal con la mano derecha

Como jugador, quiero que el sistema registre una captura fallida cuando intente capturar un animal con mi mano derecha en el lado incorrecto o deje caer al animal.

Criterios de aceptación:

- Al capturar un animal con la mano derecha en el lado incorrecto o dejarlo caer, el sistema debe contar la captura como fallida.
- El sistema debe actualizar el contador de capturas fallidas y mostrar una indicación visual de fallo.
- Captura exitosa de animal con la mano izquierda

Como jugador, quiero que el sistema registre una captura exitosa cuando capture un animal correctamente con mi mano izquierda.

Criterios de aceptación:

- Al capturar un animal con la mano izquierda en el lado correcto, el sistema debe contar la captura como exitosa.
- El sistema debe actualizar el contador de capturas exitosas y mostrar una indicación visual de éxito.
- Captura fallida de animal con la mano izquierda

Como jugador, quiero que el sistema registre una captura fallida cuando intente capturar un animal con mi mano izquierda en el lado incorrecto o deje caer al animal.

Criterios de aceptación:

- Al capturar un animal con la mano izquierda en el lado incorrecto o dejarlo caer, el sistema debe contar la captura como fallida.
- El sistema debe actualizar el contador de capturas fallidas y mostrar una indicación visual de fallo.
- Mostrar resultados finales del juego

Como jugador, quiero ver los resultados finales de mi sesión de juego al finalizar el tiempo.

Criterios de aceptación:

- El sistema debe mostrar un resumen de las capturas exitosas y fallidas.
- Debe haber una indicación clara del final del juego.
- Notificación de fin del juego

Como jugador, quiero ser notificado cuando el juego haya terminado.

Criterios de aceptación:

- Al finalizar el tiempo de juego, el sistema debe mostrar una notificación clara de que el juego ha terminado.
- La notificación debe incluir los resultados finales y una opción para reiniciar o salir del juego.

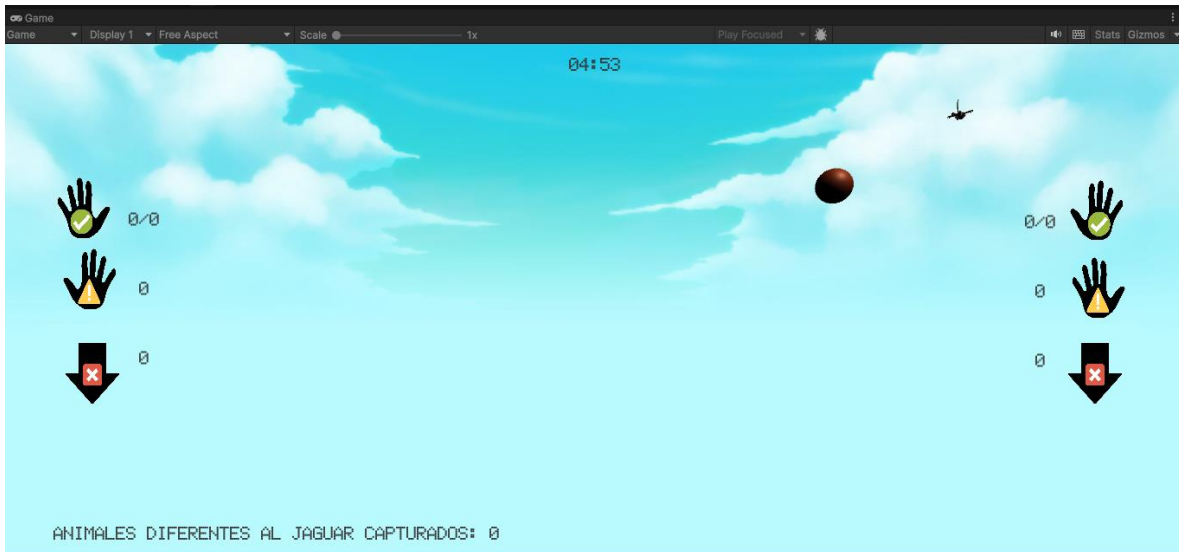


Figura 16: Juego

Como resultado al momento de diseñar en el editor de Unity el juego presentado en la **Figura 16**

Para esta escena de ver los resultados del juego llevamos a cabo la realización de historias de usuario.

- Solicitar mostrar resultados del juego

Como jugador, quiero poder solicitar la visualización de los resultados de mi sesión de juego.

Criterios de aceptación:

- Debe haber una opción para solicitar la visualización de los resultados una vez finalizado el juego.
- Al seleccionar esta opción, el sistema debe mostrar los resultados.
- Obtener puntuación actual

Como jugador, quiero que el sistema obtenga la puntuación actual para mostrar los resultados precisos de mi sesión de juego.

Criterios de aceptación:

- La puntuación obtenida debe incluir detalles de las capturas realizadas con ambas manos y el total de capturas.
- Mostrar capturas del lado izquierdo

Como jugador, quiero ver un desglose de los animales capturados con la mano izquierda durante mi sesión de juego.

Criterios de aceptación:

- El sistema debe mostrar una lista de los animales capturados con la mano izquierda.
- La lista debe incluir el número de capturas exitosas y fallidas para cada tipo de animal.
- Mostrar capturas del lado derecho

Como jugador, quiero ver un desglose de los animales capturados con la mano derecha durante mi sesión de juego.

Criterios de aceptación:

- El sistema debe mostrar una lista de los animales capturados con la mano derecha.
- La lista debe incluir el número de capturas exitosas y fallidas para cada tipo de animal.
- Mostrar total de capturas

Como jugador, quiero ver el total de animales capturados durante mi sesión de juego.

Criterios de aceptación:

- El sistema debe mostrar el total de animales capturados, sumando las capturas exitosas y fallidas de ambas manos.
- Debe haber una presentación clara y concisa del total general de capturas.

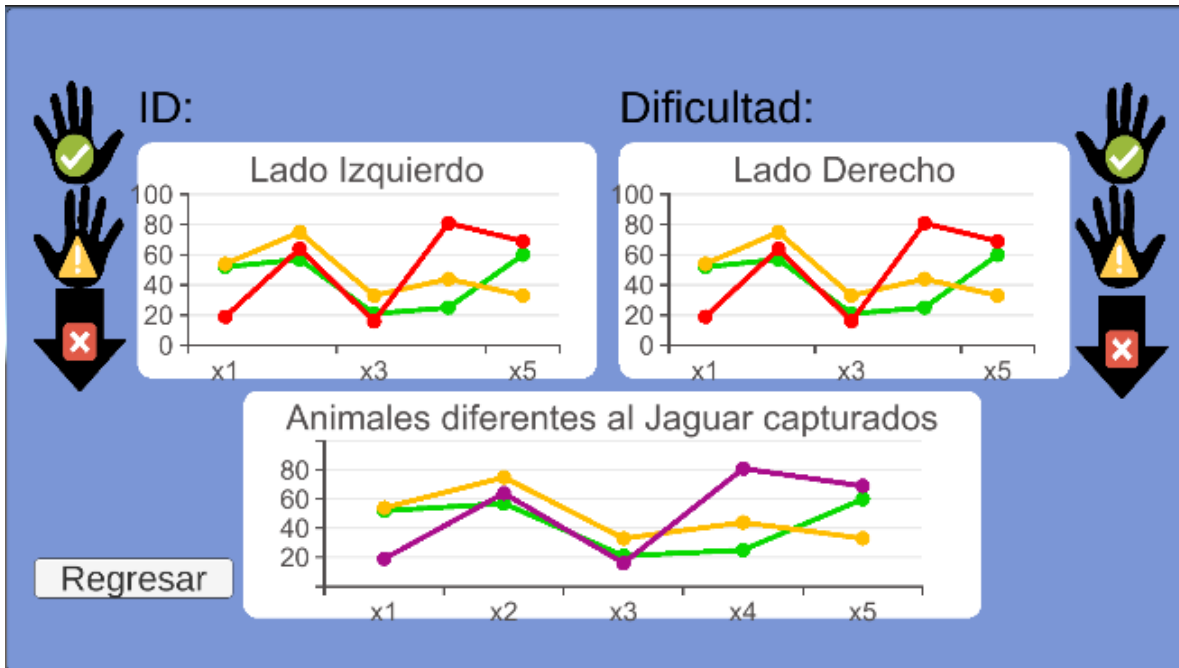


Figura 17: Mostrar resultados

Como resultado al momento de diseñar en el editor de Unity el mostrador de resultados presentado en la **Figura 17**

Para esta escena de ver configurar las velocidades según el nivel, llevamos a cabo la realización de historias de usuario.

- Configurar la velocidad según el nivel

Como usuario profesional encargado, quiero poder ajustar la velocidad a la que los animales se mueven a través de la pantalla en diferentes niveles de dificultad para adaptar el juego a las necesidades específicas de cada jugador.

Criterios de aceptación:

- Debe haber una opción en el menú de configuración para ajustar la velocidad de los animales en los niveles fácil, medio y difícil.
- Al ajustar los controles deslizantes, los valores de velocidad deben actualizarse y mostrarse en segundos.
- El sistema debe permitir guardar los ajustes de velocidad personalizados para cada nivel de dificultad.

- Guardar la configuración de velocidad

Como usuario profesional encargado, quiero poder guardar las configuraciones de velocidad ajustadas para asegurar que los cambios se apliquen en las futuras sesiones de juego.

Criterios de aceptación:

- Debe haber un botón "Guardar" que permita almacenar las configuraciones de velocidad ajustadas.
- Al presionar el botón "Guardar", el sistema debe confirmar que los ajustes han sido guardados correctamente.
- Las configuraciones guardadas deben ser persistentes y aplicarse en futuras ejecuciones del juego.
- Navegar de regreso al menú principal

Como usuario profesional encargado, quiero poder regresar al menú principal desde la pantalla de configuración para navegar fácilmente a otras secciones del juego.

Criterios de aceptación:

- Debe haber un botón "Regresar" que permita volver al menú principal.
- Al presionar el botón "Regresar", el sistema debe cerrar la pantalla de configuración y mostrar el menú principal.

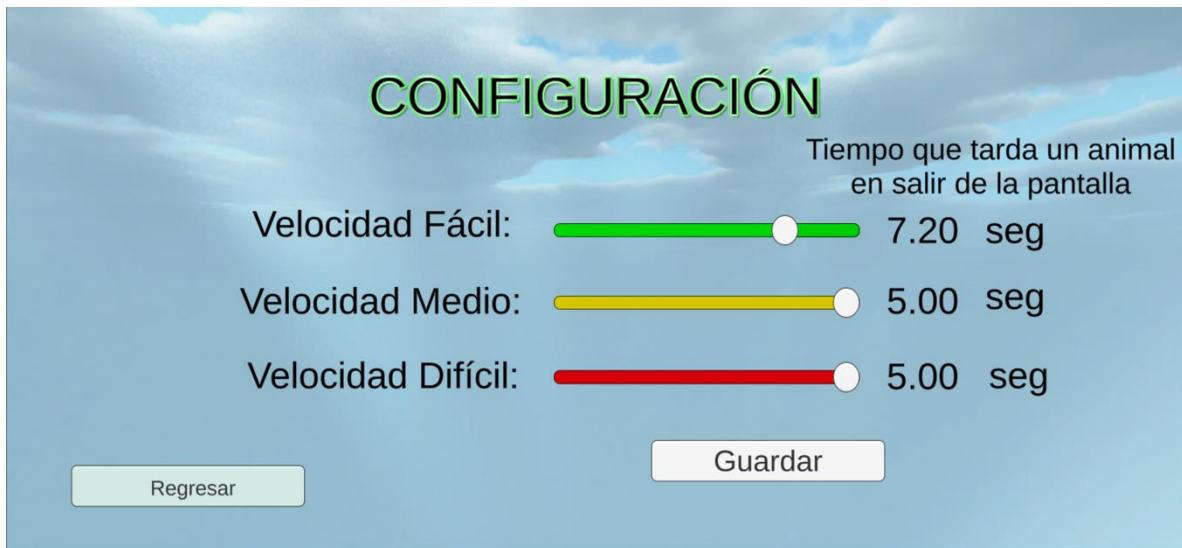


Figura 18: Configurar velocidad

Como resultado al momento de diseñar en el editor de Unity la escena de configuración de velocidad presentado en la **Figura 18**

5.6 Modelo de Datos

En la **Figura 19** podemos ver el modelo de datos relacional creado para guardar tanto los diferentes jugadores con su información, como la información de todas sus partidas en las diferentes dificultades, fácil, medio y difícil.

La tabla Players será la encargada de guardar la información de todos los jugadores. Cada jugador contiene un ID, un diagnóstico, la fecha de nacimiento, el sexo y su nivel escolar actual. Tiene una relación de uno a muchos a la tabla Games puesto que cada jugador puede tener varias partidas.

Por otro lado, están las tablas Games y Results que contendrán el progreso de un jugador guardando la información de los jaguares que aparecen en el lado izquierdo y derecho y cuantos capturó en cada uno de los lados, sea con la mano correcta e incorrecta, así como también todos los jaguares no capturados en cada uno de los lados, y la cantidad de animales diferentes al jaguar capturados.

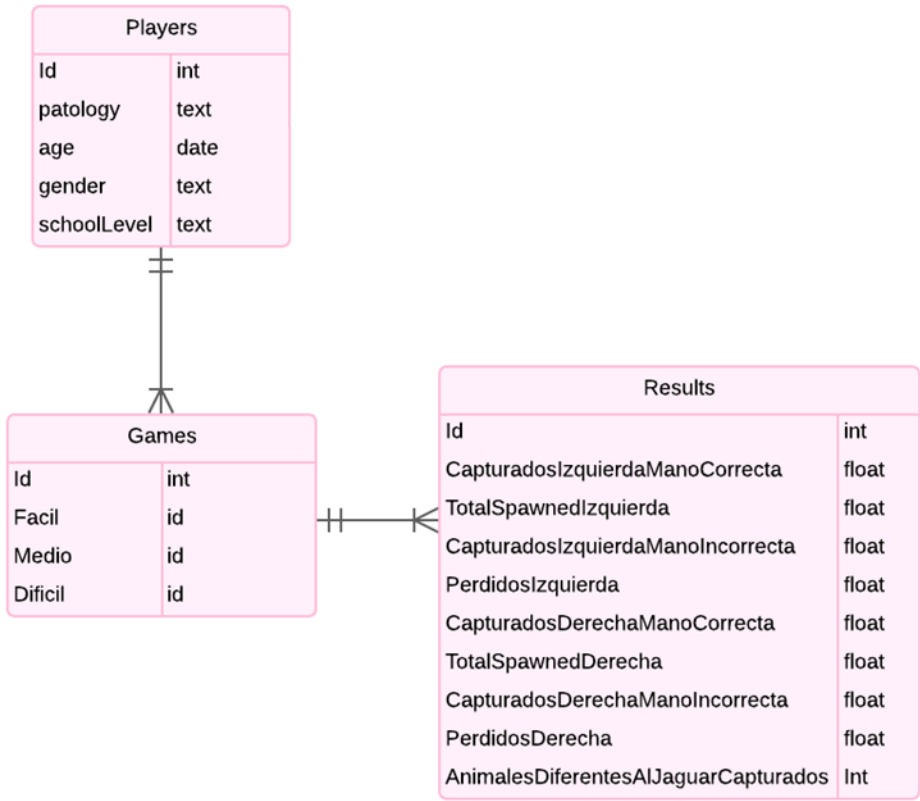


Figura 19 Modelo de datos

5.7 Diseño de Algoritmo

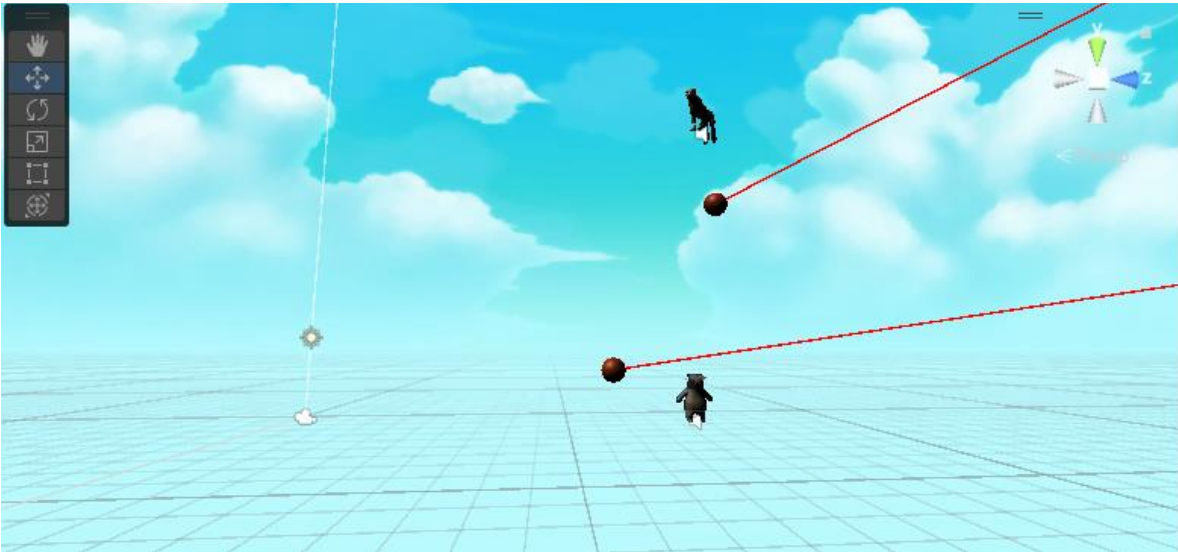


Figura 20: Vista lateral captura de animales

Una de las partes de diseño del algoritmo más interesantes que tenemos es la manera de detectar colisiones entre esferas y animales, cada una tiene una línea roja que sale directamente del centro de la esfera hacia la dirección opuesta de la cámara. Así se asegura de que desde el punto de vista de la cámara cuando una esfera toca un animal, realmente lo que sucede es que la línea roja toca al animal, esta detecta la colisión y el animal desaparece y aumentan los contadores, como se puede observar en la **Figura 20**.

6 Implementación

6.1 Elección del motor de videojuegos

En el motor de videojuegos Unity permite a los diseñadores en hacer juegos. Lo que lo hace importante es que no se necesitan años de experiencia programando para hacerlos. Primero se crean escenas con los objetos del juego, que pueden usarse para crear el menú principal, niveles individuales, formularios y cualquier otra escena importante en el contexto del juego; cada escena se puede acoplar a las necesidades que implica el proyecto, por lo que se puede colocar ambiente, obstáculos y decoraciones; el diseño y la construcción del juego por partes [21].

En cada escena de nuestro proyecto en Unity se definen objetos que son los GameObjects, estos son unos de los tipos más importantes en Unity. Estos necesitan propiedades o instrucciones antes de que se pueda convertir en algo más, como puede ser un personaje, un ambiente o un efecto especial, pero cada uno de estos objetos tiene una función diferente, ya que los GameObjects funcionan como contenedores, es decir, pueden guardar diferentes funciones que son requeridas para hacer un personaje, una luz, un sonido, un botón o cualquier cosa que sea necesaria en el proyecto; estas funciones son llamadas Components.

Ya sabiendo lo que son las escenas, su composición de manera general y su funcionamiento, podemos explicar que el sistema contiene siete (7) escenas:

- Menú
- Niveles
- Seleccionar jugador
- Crear nuevo jugador
- Mostrar resultados
- Juego
- Configuración de velocidad

En el menú se añaden diferentes botones y textos para que el usuario pueda seleccionar la opción que desea y así poder cambiar de escena. En esta escena el

usuario profesional encargado puede seleccionar si desea crear un nuevo jugador en la base de datos o si quiere ingresar al juego con un jugador ya creado. Si el usuario profesional encargado selecciona crear un nuevo jugador, lo lleva a otra escena tipo formulario donde tendrá que llenar los campos: identificador, patología, fecha de nacimiento, sexo y nivel escolar que se almacenan en la clase Save Update Data; una vez llenado esos campos, lo llevará a seleccionar la dificultad y dependiendo de la dificultad se cambiará las velocidades en las que jugará el usuario jugador/niño; una vez seleccionado la dificultad lo llevará al juego y comenzará de forma inmediata. Por otro lado, si el profesional escoge seleccionar jugador lo llevará a otra escena donde podrá consultar el ID del jugador para así poder pasar a la escena de seleccionar nivel y seguiría la misma lógica anterior; en esta escena de seleccionar jugador, el profesional también puede ver, modificar y borrar la información del jugador o al jugador, y también puede ver por medio de una gráfica el rendimiento del ID del jugador que busque. Una vez empieza el juego, se puede pausar con la letra P y se desplegará un menú donde se puede reanudar o reiniciar el juego, lo que permite al profesional volver a iniciar con el jugador/niño el juego o pausarlo si se presenta algún inconveniente.

Dificultades:

- Fácil: Velocidad (A elección del profesional encargado)
- Intermedio: Velocidad (A elección del profesional encargado)
- Difícil: Velocidad (A elección del profesional encargado)

La escena VR del juego, el usuario jugador/niño con el entorno mediante gestos para capturar jaguares utilizando sus manos. Se utiliza el dispositivo Kinect, el cual detecta el cuerpo del niño, hace que solo se enfoque en la detección de las manos por medio de sensores del dispositivo, guardándolos en un Script llamado Body Source View. Una vez el juego inicia, un contador de tiempo manejado por la clase TimerController registra los contadores jaguares capturados por derecha con la mano derecha, jaguares capturados por derecha con la mano izquierda (capturó mal al jaguar), jaguares capturados por izquierda con la mano izquierda, jaguares capturados por la izquierda con la mano derecha (capturó mal al jaguar), jaguares

que no capturó por derecha, jaguares que no capturó por izquierda y total de animales que capturó. Al finalizar los 5 minutos de juego, se muestran los resultados totales.

6.2 Descripción del sistema inicial

Para el desarrollo del proyecto, se proporcionó un sistema básico que sirvió como punto de partida para la implementación final del videojuego. Este sistema inicial contenía una escena simplificada con los siguientes elementos y mecánicas:

1. Aparición y Movimiento de Animales

El sistema inicial incluía una mecánica básica donde diferentes animales eran generados la parte superior de la pantalla. Estos animales caían hacia la parte inferior de la pantalla en un movimiento continuo y repetitivo. La gravedad o un simulacro de la misma estaba presente en el sistema, haciendo que los animales descendieran de forma constante.

- **Mecánica de Reaparición:** Si algún animal alcanzaba la parte inferior de la pantalla sin ser capturado, este desaparecía y volvía a generar en la parte superior, comenzando nuevamente su caída. Este ciclo continuo aseguraba un flujo constante de animales en la pantalla, proporcionando un escenario básico para la interacción del jugador.

2. Esferas y Rayos de Captura

En la escena también estaban presentes dos esferas, cada una posicionada en un lado opuesto de la pantalla. Estas esferas emitían un rayo rojo en orientación horizontal que actuaba como el mecanismo de captura para los animales.

- **Rayo de Captura:** El rayo rojo emitido por las esferas era el elemento clave de interacción en esta fase inicial. Cuando un animal en caída entraba en contacto con el rayo, este era "capturado" instantáneamente. La captura causaba que el animal desapareciera de su ubicación actual y volviera a generar en la parte superior, listo para comenzar otra caída.

Este sistema básico permitía capturar animales sin la intervención directa del jugador en la detección o en el control de las esferas, lo que establecía las bases para una mayor personalización e interactividad en fases posteriores del desarrollo.

6.3 Metodología para Configurar la Gravedad en la Caída de Animales

Para la configuración de la velocidad de caída de animales se hizo una pantalla para personalizar el tiempo que tardaba un animal en salir en la parte superior de la pantalla y desaparecer en la parte inferior. Para esto se creó una nueva escena llamada configuración donde se podía personalizar cada uno de los niveles fácil, medio y difícil, como se muestra en la **Figura 21**.



Figura 21: Escena configuración de velocidad

Para personalizar la velocidad de caída de los animales de debe modificar el valor de la variable de gravedad de Unity, por lo que se optó a crear una función que transformará la cantidad de segundos en caer un animal, hasta desaparecer por la parte inferior de la pantalla, haciendo así el cambio de valor que debe tomar la variable de gravedad de Unity para que esto suceda.

Para esto se hicieron varias pruebas con cronómetro, asociando el tiempo y velocidad para sacar puntos discretos de la función que relacione el tiempo en el

que un animal debe salir de la pantalla, y la gravedad necesaria para que esto suceda. Se tomaron los datos de la tabla de la **Tabla 1**.

tiempo	velocidad
6,68	0,44
5,14	0,65
4,43	0,95
3,5	1,32
7,51	0,33
13	0,11

Tabla 1: Puntos discretos tiempo x velocidad

Por lo que se usó Excel para sacar la función que relaciona tiempo x velocidad, de modo que pueda ingresar el tiempo que el usuario profesional requiera dejar caer un animal, y se retorne la gravedad que se debe poner en Unity para que el animal tome la velocidad necesaria en caer. En la **Figura 22** se puede ver la función y los puntos discretos de la tabla.

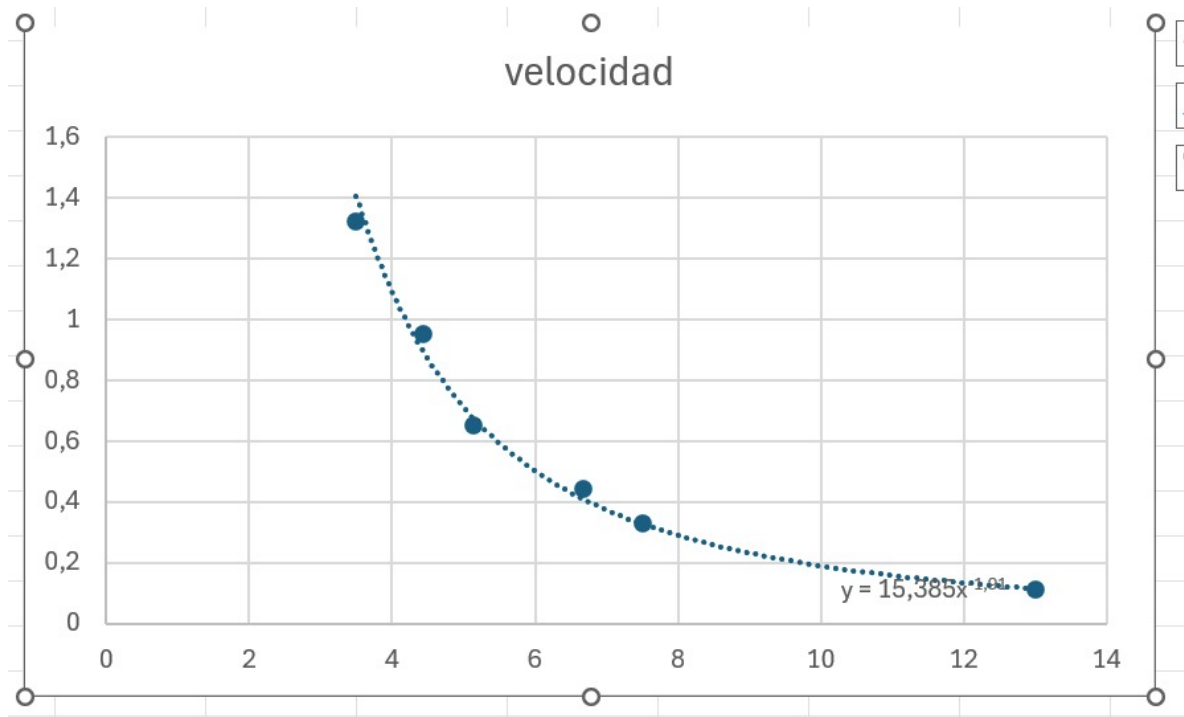


Figura 22: Función tiempo x velocidad y puntos discretos

La función resultante es:

$$v(t) = 15.385x^{-1,01} ,$$

Esta representa la relación tiempo/velocidad en la que el usuario puede ingresar la cantidad de tiempo que requiere dejar caer un animal de la pantalla, y el programa pueda ingresar el valor necesario a la variable de gravedad de Unity para hacer esto posible.

6.3 Líneas de código relevantes

6.3.1 Aparición de animales

Para la aparición de animales se creó la función `SpawnFBX()` que se encarga de generar animales en el juego de manera semi aleatoria. La función recorre una lista de índices y en cada iteración espera un tiempo aleatorio entre uno (1) y cuatro (4) segundos antes de crear un nuevo animal en una posición aleatoria dentro de la pantalla. Además, ajusta la velocidad de caída de cada animal y le asigna la capa "Animales". Si el nombre del objeto contiene "Jaguar", también se le asigna la

etiqueta "Jaguar". Este proceso continúa hasta que todos los animales en la lista hayan sido generados, asegurando una distribución variada y dinámica de los animales en el juego, como se puede observar en la **Figura 23**.

```
IEnumerator SpawnFBX()
{
    int i = 0;
    while (i < facesIndex.Count)
    {
        yield return new WaitForSeconds(Random.Range(1, 4));

        int faceIndex = facesIndex[i];
        GameObject spawnedFBX = Instantiate(fbxFaces[faceIndex], new Vector3(Random.Range(-15, 15), 12, 16), Quaternion.identity);
        Rigidbody[] rigidbodies = GetComponentsInChildren<Rigidbody>(true);

        // Iterar sobre cada Rigidbody encontrado
        foreach (Rigidbody rb in rigidbodies)
        {
            // Verificar si el Rigidbody es diferente de nulo
            if (rb != null)
            {
                // Cambiar la velocidad del Rigidbody
                rb.velocity = new Vector3(0, -fallVelocity, 0);
            }
        }

        // Asignar capa "Animales" al objeto principal y sus hijos
        AssignLayerAndTag(spawnedFBX, "Animales", spawnedFBX.name.Contains("Jaguar") ? "Jaguar" : "");

        i++;
    }
}
```

Figura 23: Función para generar animales de forma aleatoria

6.3.2 Controlador de colisiones

Como se explica en la sección 5.6, para controlar las colisiones se creó la clase "SphereJoint", que en su función "Update" se encarga de detectar si se capturó un animal o no.

Primero, calcula la dirección desde la esfera hacia la cámara en el origen y crea un rayo en esa dirección; luego, utiliza un "Raycast" para verificar si el rayo golpea algún objeto en un rango de 30 unidades. Si se detecta una colisión, se verifica si el objeto golpeado tiene la etiqueta "Condor", "Jaguar", "Oso" o "Titi". Si el objeto es un "Jaguar", se incrementa un contador específico basado en la posición del objeto (izquierda o derecha) y la mano utilizada. Si el objeto no es un "Jaguar", se incrementa un contador general. Después de esto, el objeto golpeado se desactiva visualmente y se reproduce un sonido asociado al objeto, según la **Figura 24**.

```

void Update()
{
    // Calcular la dirección desde la esfera hacia la cámara en el origen
    Vector3 directionToCamera = transform.position.normalized;

    Ray ray = new Ray(transform.position, directionToCamera);
    RaycastHit hit;

    // Verificar si el rayo golpea algún objeto
    if (Physics.Raycast(ray, out hit, 30f)){
        GameObject fbx = hit.collider.gameObject;

        if (fbx.CompareTag("Condor") || fbx.CompareTag("Jaguar") || fbx.CompareTag("Oso") || fbx.CompareTag("Titi")){
            string handUsed = gameObject.tag;

            // Calcular la posición del objeto golpeado con respecto al centro
            float posX = fbx.transform.position.x;
            if (fbx.CompareTag("Jaguar")){
                // Incrementar el contador correspondiente a la mano
                if (posX >= 0) // Si está a la derecha
                {
                    jaguarsRight.GetComponent<TextController>().UpdateCounter(1, false, handUsed);
                }
                else{
                    jaguarsLeft.GetComponent<TextController>().UpdateCounter(1, false, handUsed);
                }
            }
            else{
                total.GetComponent<TextController>().UpdateCounter(1, false, "NOTJaguar");
            }

            fbx.GetComponent<Renderer>().enabled = false;
            FBXController fbxController = fbx.GetComponent<FBXController>();
            fbxController.audios.Play();
            fbxController.ShowFBX(fbx, false);
            fbxController.ShowFBX(fbx, true);
        }
    }
}

```

Figura 24: Función para detectar las colisiones entre las esferas y animales

De esta manera se asegura que el juego registre las interacciones de la esfera con los objetos relevantes de manera adecuada, y actualice los contadores correspondientes.

6.4 Sistema de almacenamiento de información jugadores

Para almacenar la información de los jugadores y sus datos asociados, se implementó un sistema de serialización a formato JSON en archivos planos. A continuación, se presenta el código utilizado para serializar la clase PlayerData y guardarla en un archivo local:

```

public void SaveData()
{
    isEditMode = false;
    PlayerInfo newPlayerInfo = new PlayerInfo();
    newPlayerInfo.identifier = textIdentifier.text;
    newPlayerInfo.pathology = textPathology.text;
    newPlayerInfo.age = textAge.text;
    newPlayerInfo.gender = dropdown.options[dropdown.value].text;
    newPlayerInfo.schoolLevel = textSchoolLevel.text;

    bool playerExists = false;
    foreach (PlayerInfo info in playerData.playerInfoList)
    {
        info.identifier = info.identifier.Trim().Replace("\u200B", "");
        newPlayerInfo.identifier = newPlayerInfo.identifier.Trim().Replace("\u200B", "");
        if (info.identifier == newPlayerInfo.identifier)
        {
            info.pathology = newPlayerInfo.pathology;
            info.age = newPlayerInfo.age;
            info.gender = newPlayerInfo.gender;
            info.schoolLevel = newPlayerInfo.schoolLevel;
            playerExists = true;
            break;
        }
    }

    // Si el jugador no existe en la lista, agregarlo
    if (!playerExists)
    {
        playerData.playerInfoList.Add(newPlayerInfo);
    }

    // Convierte PlayerData a JSON y guarda en el archivo
    string jsonData = JsonUtility.ToJson(playerData);
    File.WriteAllText(filePath, jsonData);

    Debug.Log("Archivo Guardado");
    Debug.Log(Application.persistentDataPath);
}

```

Figura 25: Función para almacenar la información del jugador

El método “SaveData” de la **Figura 25** se encarga de actualizar la información del jugador en la lista “playerData.playerInfoList”, o agregar un nuevo jugador en caso de no existir. Luego, convierte “PlayerData” a formato JSON utilizando “JsonUtility.ToJson()” y guarda el resultado en un archivo en el directorio persistente de la aplicación (Application.persistentDataPath), debido a que este directorio es el estándar designado para almacenar datos de un juego de Unity en Windows y Mac, además de que es una carpeta segura, puesto que los datos de esta carpeta están protegidos por el mismo sistema operativo.

El método "LoadData" carga los datos de jugadores almacenados previamente desde el archivo JSON "data.json" en el directorio persistente de la aplicación; verifica si el archivo existe antes de intentar cargarlo y maneja adecuadamente los errores si el archivo no se encuentra.

6.5 Sistema de almacenamiento de información de partidas

Las clases "WeaponData" y "WeaponDataPorNivel" de la **Figura 26** se emplean para estructurar y gestionar datos estadísticos durante las partidas de un juego, facilitando el almacenamiento y acceso a métricas específicas relacionadas con la captura y pérdida de elementos. En "WeaponData", se definen listas de tipo Double que registran diversas métricas de juego, como capturas y pérdidas realizadas con diferentes manos y la cantidad total de elementos spawnados. Por otro lado, la clase "WeaponDataPorNivel" actúa como un contenedor que agrupa múltiples instancias de "WeaponData", cada una representando datos correspondientes a diferentes niveles de dificultad del juego, facilitando así la comparación y análisis de desempeño entre niveles. Este diseño permite una organización estructurada de datos cruciales para la evaluación y ajuste de la jugabilidad del juego en distintos escenarios de dificultad.

```

public class WeaponData
{
    public List<double> CIMC; //Capturados Izquierda mano correcta
    public List<double> TSI; //Total spawned Izquierda
    public List<double> CIMI; //Capturados Izquierda Mano Izquierda
    public List<double> PI; //Perdidos Izquierda
    public List<double> CDMC; //Capturados Derecha Mano Derecha
    public List<double> TSD; //Total Spawned Derecha
    public List<double> CDMI; //Capturados Derecha Mano incorrecta
    public List<double> PD; //Perdidos Derecha
    public List<double> ADJC; //Animales Diferentes al Jaguar Capturados
}

2 references
public class WeaponDataPorNivel
{
    public WeaponData facil;
    public WeaponData medio;
    public WeaponData dificil;
}

```

Figura 26: Clases para gestionar los datos estadísticos durante el juego

Toda la información recopilada de las clases “WeaponData” y “WeaponDataPorNivel” se centraliza y guarda de manera estructurada en la variable “jsonData”, como se puede observar en la **Figura 27**.

```

string jsonSave = JsonUtility.ToJson(dataSave, true);

File.WriteAllText(Application.persistentDataPath + idJugador, jsonSave);

string path = Application.persistentDataPath + idJugador;

```

Figura 27: Variable para almacenar la información del jugador

7 Pruebas y Validación

7.1 Descripción de la prueba

Para probar el sistema, se realizaron pruebas de usuario en el Instituto para Niños Ciegos y Sordos del Valle del Cauca, donde se llevaron a cabo siete (7) pruebas en total con tres (3) niños evaluados. En estas pruebas se recolectaron tres (3) tipos de datos importantes: los resultados de las partidas jugadas, una encuesta a los niños sobre su opinión del juego y una encuesta para los educadores donde pudieron dar observaciones más puntuales y adaptativas al sistema.

Las pruebas de usuario sobre la aplicación nos ayudan a obtener valoraciones y comentarios adicionales sobre la experiencia de los usuarios. A través de la observación directa, se tomaron notas sobre las posibles dificultades y errores que los niños tuvieron durante la ejecución de la aplicación. Los resultados de las partidas jugadas en las pruebas quedaron almacenados en el sistema, proporcionando indicadores del desempeño de los niños al interactuar con el sistema. Esto también permitió evaluar qué tan fácil o difícil fue para cada niño completar las tareas designadas, considerando sus diferentes discapacidades.

La información recolectada de los jugadores en relación con los resultados obtenidos en las partidas, nos permitió analizar cómo las dificultades y deficiencias auditivas o visuales afectan la capacidad de uso de la aplicación. Finalmente, la encuesta a los educadores y profesionales del instituto, ayudó a validar la utilidad de la aplicación como medio de educación y como herramienta para evaluar el desarrollo cognitivo de los niños. Además, la encuesta proporcionó comentarios valiosos sobre los aspectos del sistema que podrían mejorarse para optimizar la interacción y el aprendizaje de los niños.

7.2 Resultados de las pruebas

Los resultados de las partidas de los jugadores se pueden ver representados en la tabla de la **Figura 28**. En total, se realizaron siete (7) pruebas y en la primera columna está el identificador de cada jugador que participó.

La información de las discapacidades de los jugadores se presenta en la **Figura 29**. En la segunda columna de la tabla se indica la dificultad en la que se ejecutó cada prueba, por ejemplo, el jugador 1 en la prueba 1 jugó en la dificultad fácil, en la prueba 2 en dificultad intermedia y en la prueba 3 en difícil.

En las columnas subsiguientes se detallan las capturas correctas e incorrectas de jaguares, segmentadas por el uso de la mano izquierda o derecha. Esto nos proporciona una visión detallada de la precisión y destreza de los jugadores en cada nivel de dificultad. La penúltima columna muestra el total de animales capturados (diferentes al jaguar) por cada jugador.

Adicionalmente, las columnas finales presentan el porcentaje de capturas correctas realizadas con la mano izquierda y la derecha, respectivamente. Estos datos permiten un análisis detallado del desempeño de los niños, facilitando la identificación de patrones y áreas que necesitan mejora.

Por ejemplo, en el nivel fácil, el jugador 1 logró un 72% de capturas correctas con la mano izquierda, mientras que en el nivel difícil, el porcentaje de capturas correctas con la mano izquierda disminuyó a 48%. Esto demuestra cómo la dificultad del juego puede afectar el rendimiento de los jugadores. Similarmente, el jugador 2 mostró un rendimiento más consistente, capturando correctamente el 78% de los jaguares con la mano izquierda en nivel intermedio y el 60% en nivel difícil.

Los resultados obtenidos indican que en general, la precisión de las capturas disminuye con el aumento de la dificultad del juego. Por otra parte, el porcentaje de capturas incorrectas y animales dejados caer, también aumenta; sugiriendo una mayor dificultad y complejidad en los niveles superiores.

Este análisis permite a los profesionales encargados entender mejor las capacidades y limitaciones de los niños, ajustando el nivel de dificultad de manera adecuada para cada jugador según sus características y capacidades visuales, auditivas y edad. Además, proporciona una base sólida para mejorar la experiencia de los usuarios y adaptar el juego para maximizar su efectividad educativa y evaluativa.

ID	Nivel de dificultad	Jaguares capturados por izquierda con la mano izquierda(Correcto)	Jaguares capturados por derecha con la mano izquierda(Incorrecto)	Jaguares capturados por izquierda que dejo caer	Jaguares capturados por derecha con la mano derecha (Correcto)	Jaguares capturados por izquierda con la mano derecha(Incorrecto)	Jaguares capturados por derecha que dejo caer	Total animales capturados(diferentes al jaguar)			Izquierda	Derecha
1	Facil	72%	8%	20%	65%	20%	15%	22			100%	100%
1	Intermedio	60%	0%	40%	63%	27%	10%	13			100%	100%
1	Dificil	48%	4%	48%	45%	15%	40%	21			100%	100%
2	Intermedio	78%	10%	12%	64%	30%	6%	24			100%	100%
2	Dificil	60%	32%	8%	78%	9%	13%	17			100%	100%
3	Intermedio	53%	7%	40%	46%	50%	4%	15			100%	100%
3	Dificil	33%	18%	49%	52%	0%	48%	24			100%	100%

Figura 28: Tabla de resultados obtenidos por los jugadores en las pruebas

7.3 Información de jugadores y preguntas

La encuesta sobre la opinión de los niños se hizo de manera directa tomando nota de sus respuestas. En la tabla de la **Figura 29** se puede observar algunas de sus respuestas e igualmente las discapacidades de los niños.

ID	Te gusto el videojuego?	Te parecio facil, intermedio, dificil?	Algun comentario adicional?	Diagnostico	Edad	Grado escolar
1	Si	Intemedio	Que fuera mas lento. Debe durar más	retinosquisis izquierda	9	2do
2	Si	Intemedio	Le gustó los jaguares y fue divertido	hipoacusia izquierda	10	1ro
3	Si	Intemedio	Le gusto todo. Los animales son faciles de identificar	No tiene	7	1ro

Figura 29: Tabla con la información de los jugadores y las preguntas realizadas a los jugadores

Como podemos observar en la **Figura 29**, a todos los participantes de las pruebas les gustó el videojuego. Los tres jugadores consideraron que el nivel de dificultad fue intermedio. Los comentarios adicionales reflejan sus experiencias y sugerencias: el jugador 1 mencionó que el juego debería ser más lento y durar más tiempo, el jugador 2 disfrutó particularmente de los jaguares y encontró el juego divertido, mientras que el jugador 3 destacó que los animales eran fáciles de identificar.

Entre los participantes, se encuentran diversas condiciones: retinosquisis izquierda, hipoacusia izquierda, y un jugador sin ninguna discapacidad. El jugador 1, de nueve

(9) años de edad y cursando segundo (2do) grado, tiene retinosquiasis izquierda. El jugador 2, de diez (10) años de edad y en primer (1er) grado, tiene hipoacusia izquierda. El jugador 3, de siete (7) años de edad y en primer grado (1er), no presenta ninguna discapacidad.

De los resultados de las pruebas y las discapacidades de los jugadores, podemos deducir que en términos generales, los jugadores con discapacidades auditivas obtuvieron mejores resultados que los jugadores con visión reducida. Por ejemplo, el jugador 2 con hipoacusia izquierda, mostró un rendimiento consistente en comparación con el jugador con retinosquiasis (jugador 1). Además, es importante destacar que las características individuales y las adaptaciones necesarias pueden influir en la experiencia de juego y en los resultados obtenidos. Por ejemplo, ajustar la velocidad del juego y la duración, puede mejorar la experiencia para ciertos jugadores, como sugiere el comentario del jugador 1.

Estos datos son fundamentales para ajustar y mejorar el diseño del videojuego, asegurando que sea accesible y atractivo para todos los niños, independientemente de sus capacidades visuales o auditivas. Las observaciones de los jugadores y los comentarios de los educadores son sólidas para futuras iteraciones del proyecto, para maximizar su efectividad educativa y evaluativa.

7.4 Encuesta a profesionales del Instituto para niños Ciegos y Sordos del Valle del Cauca

Para la encuesta de los profesionales encargados, se realizó un formulario de Google para obtener las respuestas fácilmente. A continuación, vemos las preguntas con las respuestas generales de la encuesta:

- La pregunta **¿Usaría el videojuego para evaluar el desarrollo cognitivo de los niños?** busca validar que los profesionales encargados tengan una herramienta interactiva útil adicional, para evaluar diferentes aspectos cognitivos de los niños con discapacidad.

Como se observa en la **Figura 30**, todos los encuestados (100%) respondieron afirmativamente, indicando su disposición a utilizar el videojuego para este fin.

¿Usaría el videojuego para evaluar el desarrollo cognitivo de los niños?

 Copiar

1 respuesta

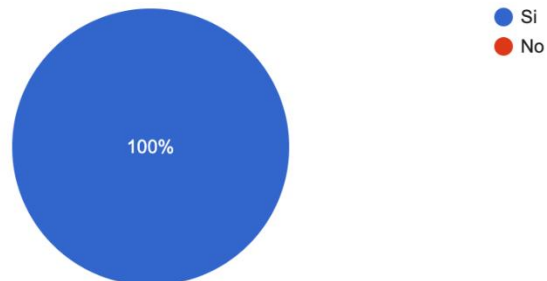


Figura 30: Pregunta 1 - encuesta a profesionales encargados

- La pregunta **¿Considera que el sistema es fácil de usar?** esto ayuda a evaluar la usabilidad del sistema desde la perspectiva de los profesionales. Como se muestra en la **Figura 31**, los dos profesionales que respondieron la encuesta consideran que el sistema es fácil de utilizar.

¿Considera que el sistema es fácil de usar?

 Copiar

2 respuestas

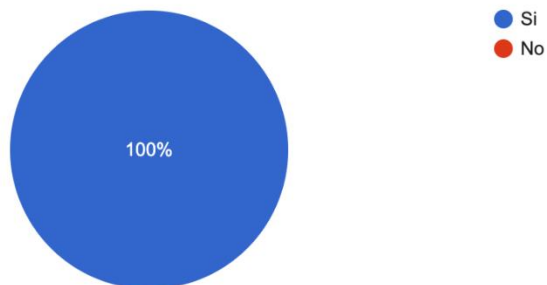


Figura 31: Pregunta 2 - encuesta a profesionales encargados

- La pregunta **¿Considera que la implementación de sonidos y efectos visuales mejora la experiencia de los niños durante el uso del videojuego?** se enfoca en la percepción de los profesionales sobre la efectividad de los elementos multimedia integrados en el videojuego.

Según la **Figura 32**, ambos encuestados creen que los sonidos y efectos visuales mejoran la experiencia de los niños.

¿Considera que la implementación de sonidos y efectos visuales mejora la experiencia de los niños durante el uso del videojuego?

 Copiar

2 respuestas

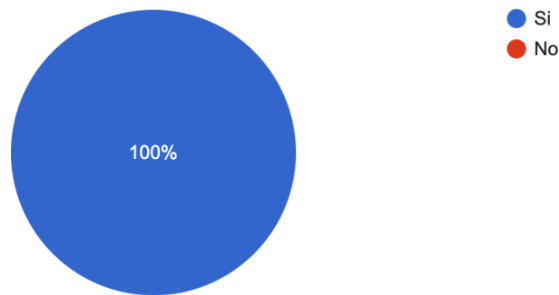


Figura 32: Pregunta 3 - encuesta a profesionales encargados

- Finalmente, se solicitó a los profesionales que indicaran **¿Qué aspectos cambiaría del sistema/videojuego?** los comentarios recibidos incluyen sugerencias como la adición de una línea media punteada para distinguir entre derecha e izquierda, un mensaje al final de la partida que indique el resultado y sirva de refuerzo positivo, y mejoras en la visibilidad de la división entre los lados izquierdo y derecho de la pantalla. Además, se sugirió modificar los símbolos en los lados, ya que pueden interferir con la visibilidad de los animales que pasan por los extremos.

Qué aspectos cambiaría del sistema/videojuego?

2 respuestas

Linea media punteada para distinguir derecha e izquierda
Mensaje al final de la partida que indique el resultado y sirva de refuerzo positivo
Ocapacidad en el HUD

Hacer visible la división entre el lado izquierdo y derecho de la pantalla, ya que cuando los animales pasan muy cerca de la mitad no se diferencia con que mano se deben tomar

Modificar los símbolos que están a los lados ya que no permiten ver los animales que pasan por los extremos

Figura 33: Pregunta 4 - encuesta a profesionales encargados

Aunque los profesionales consideran el sistema fácil de utilizar, se observó a nivel de implementación algunas dificultades detalladas después en la **Figura 33**. Las observaciones y comentarios proporcionados son fundamentales para realizar mejoras en el diseño y funcionalidad del videojuego, asegurando así una mejor experiencia para los niños y una herramienta más eficaz para los educadores.

7.5 Validación

Según los requisitos establecidos en la sección 4.2, se llevó a cabo la validación del sistema para asegurar que cumpliera con todos los requerimientos especificados. A continuación, se describen los métodos y procedimientos utilizados para verificar el correcto funcionamiento de cada uno de los requisitos del proyecto.

7.5.1 Requisitos funcionales

Para garantizar que el sistema desarrollado cumpliera con los requisitos funcionales establecidos, se realizaron pruebas funcionales detalladas. Luego se describen los métodos y procedimientos utilizados para verificar el correcto funcionamiento de cada requisito funcional del proyecto.

- RF-01 Interfaz de usuario

Descripción: El sistema debe permitir la interacción con una interfaz entre usuario y sistema.

Pruebas realizadas: Se diseñaron escenarios de prueba en los que diferentes usuarios interactuaron con la interfaz. Se verificó que todas las opciones del menú fueran accesibles y que la navegación entre pantallas fuera intuitiva y sin errores. Se utilizaron herramientas de testeo manual para simular interacciones y recopilar datos sobre la usabilidad y la respuesta del sistema.

Resultado: La interfaz permitió una interacción fluida y sin interrupciones, cumpliendo con el requisito funcional especificado.

- RF-02 Menú inicial

Descripción: Al iniciar el juego, el sistema debe mostrar un menú para darle acceso a la aplicación.

Pruebas realizadas: Se verificó que al iniciar la aplicación, el menú inicial se desplegara correctamente. Se comprobó que todas las opciones del menú estuvieran visibles y que al seleccionarlas, llevaran al usuario a la sección correspondiente del juego.

Resultado: El menú inicial se mostró correctamente al iniciar el juego, y todas las opciones del menú funcionaron según lo esperado.

- RF-03 Puntaje

Descripción: El sistema debe mostrar mediante la interfaz un puntaje del sistema de evaluación psicométrico completado por el usuario jugador.

Pruebas realizadas: Se jugaron múltiples sesiones del juego para asegurar que el puntaje se calculaba y se mostraba correctamente al finalizar cada partida. Se verificó que los puntajes reflejaran con precisión el desempeño del jugador.

Resultado: El sistema mostró el puntaje correcto al finalizar cada partida, cumpliendo con el requisito de evaluación psicométrica.

- RF-04 Seleccionar Jugador

Descripción: El sistema debe permitir al usuario profesional encargado seleccionar el jugador por medio de un ID.

Pruebas realizadas: Se ingresaron diferentes ID de jugadores en el sistema para asegurar que se podía acceder a la información de cada jugador de manera precisa. Se verificó que el sistema cargara los datos correctos para cada ID ingresado.

Resultado: El sistema permitió la selección de jugadores mediante un ID efectiva y precisa.

- RF-05 Contraste de elementos y videojuego

Descripción: Los elementos visuales del sistema deben tener alto contraste para facilitar la visualización.

Pruebas realizadas: Se revisaron todos los elementos visuales del juego para asegurar que tenían un alto contraste. Se utilizaron herramientas de accesibilidad para evaluar el contraste de los colores utilizados.

Resultado: Todos los elementos visuales del sistema presentaron un alto contraste, cumpliendo con los requisitos de accesibilidad visual.

- RF-06 Dificultades del juego

Descripción: Sin importar el modo escogido, el usuario profesional encargado debe poder seleccionar la dificultad del juego antes de empezar.

Pruebas realizadas: Se comprobó que el sistema permitiera la selección de diferentes niveles de dificultad antes de iniciar el juego. Se verificó que cada nivel presentara un desafío adecuado según la dificultad seleccionada.

Resultado: El sistema permitió seleccionar la dificultad del juego sin problemas, y cada nivel de dificultad funcionó como se esperaba.

- RF-07 Creación nuevo jugador

Descripción: El sistema debe contar con una interfaz que permita al usuario profesional encargado ingresar la información de un nuevo jugador.

Pruebas realizadas: Se realizaron pruebas para agregar nuevos jugadores al sistema. Se ingresó información en los campos de datos y se comprobó que esta se guardara correctamente.

Resultado: El sistema permitió la creación de nuevos jugadores y guardó la información ingresada sin errores.

- RF-08 Resumen de partida

Descripción: Al finalizar cada modo del juego (niveles), el sistema debe mostrar un resumen del puntaje obtenido por el usuario jugador.

Pruebas realizadas: Al finalizar las partidas, se verificó que el sistema mostrara un resumen detallado del puntaje obtenido, incluyendo detalles sobre el desempeño del jugador.

Resultado: El resumen de la partida se mostró correctamente al finalizar cada nivel, cumpliendo con el requisito.

- RF-09 Visualización de resultados

Descripción: El sistema debe permitir al usuario profesional encargado visualizar los resultados de las partidas realizadas por cada jugador.

Pruebas realizadas: Se realizaron sesiones de juego y se guardaron los resultados. Después, se verificó que estos resultados fueran accesibles para el usuario profesional encargado.

Resultado: El sistema permitió la visualización de los resultados de las partidas de manera precisa y efectiva.

- RF-10 Captura de animales

Descripción: El sistema debe generar el sonido respectivo del animal que el usuario jugador capturó.

Pruebas realizadas: Se capturaron diferentes animales durante las sesiones de juego y se verificó que cada captura generara el sonido correspondiente.

Resultado: El sistema generó correctamente el sonido respectivo para cada animal capturado.

- RF-11 Captura por el lado izquierdo/derecho

Descripción: El sistema debe registrar y medir si la captura de los animales se realiza por el lado izquierdo o derecho, permitiendo así una evaluación detallada de la coordinación visomotora del usuario.

Pruebas realizadas: Se capturaron animales usando ambas manos en diferentes sesiones de juego. Se verificó que el sistema registrara correctamente el lado utilizado para cada captura.

Resultado: El sistema registró y midió correctamente las capturas por el lado izquierdo y derecho.

- RF-12 Medición de capturas correctas e incorrectas

Descripción: El sistema debe registrar y medir el número de capturas correctas e incorrectas realizadas por el usuario, proporcionando datos precisos sobre su desempeño durante el juego.

Pruebas Realizadas: Se llevaron a cabo múltiples sesiones de juego para asegurar que el sistema registrara correctamente el número de capturas correctas e incorrectas. Se verificó que los datos recopilados fueran precisos y coherentes.

Resultado: El sistema registró y midió con precisión las capturas correctas e incorrectas, cumpliendo con el requisito funcional.

Estas pruebas funcionales aseguraron que el sistema no solo cumpliera con todos los requisitos funcionales especificados, sino que también proporcionara una experiencia de usuario confiable. La validación exhaustiva del sistema garantiza que este sea una herramienta efectiva y útil para la evaluación psicométrica y visomotora de los niños en el Instituto para Niños Ciegos y Sordos del Valle del Cauca.

7.5.2 Requisitos no funcionales

- **RNF-01 Usabilidad:**

Descripción: La interfaz debe ser intuitiva y fácil de navegar, con instrucciones claras y accesibles para los jugadores de diferentes edades.

Pruebas realizadas: Durante las pruebas de usabilidad, se realizaron sesiones en las que participaron niños de diferentes edades, así como profesionales que actuarán como usuarios finales del sistema. Se observó cómo los niños interactuaban con la interfaz y si lograban entender las instrucciones y navegar por el juego de manera autónoma.

Resultado: Los resultados indicaron que los niños pudieron navegar por la interfaz de manera natural, sin necesidad de asistencia constante, lo que sugiere que la interfaz es intuitiva. Además, como se muestra en la **figura 31**, los profesionales consideraron que el sistema es fácil de usar, cumpliendo con el requisito de usabilidad.

- **RNF-02 Compatibilidad:**

Descripción: El videojuego debe funcionar correctamente en los principales sistemas operativos que soporten Kinect, como Windows 10 o versiones superiores, y no debe requerir software adicional que no esté ampliamente disponible.

Pruebas realizadas: Se instaló el sistema en dos computadoras diferentes, una con Windows 10 y otra con Windows 11, para verificar su compatibilidad con estos sistemas operativos. Se evaluó el funcionamiento del videojuego en ambas configuraciones.

Resultado: El videojuego funcionó correctamente en ambas plataformas, cumpliendo con el requisito de compatibilidad. No se presentaron problemas de instalación o ejecución, confirmando que el sistema es compatible tanto con Windows 10 como con Windows 11.

RNF-03 Mantenibilidad:

Descripción: El código y los componentes del sistema deben estar bien documentados para facilitar el mantenimiento y las futuras actualizaciones.

Pruebas realizadas: Se revisó el código fuente para asegurar que estuviera debidamente comentado y que la estructura del software estuviera organizada de manera clara. Se verificó que el código estuviera correctamente dividido en clases y módulos, facilitando así el mantenimiento y las futuras actualizaciones.

Resultado: El código del sistema está adecuadamente comentado, proporcionando explicaciones claras sobre la funcionalidad de cada sección. Además, el sistema está bien estructurado, con componentes divididos en clases que permiten un mantenimiento sencillo y eficiente. Esto asegura que futuras modificaciones o actualizaciones puedan realizarse de manera efectiva, cumpliendo con el requisito de mantenibilidad.

8 Conclusiones y trabajos futuros

8.1 Conclusiones

- A partir del proceso realizado se puede evidenciar que se lograron completar los objetivos propuestos para el proyecto, según la sección 2.2.
- Se logró integrar un sistema interactivo gestual desarrollado en lenguaje de programación C# para el videojuego desarrollado en Unity.
- A partir de la metodología Scrum, fue posible desarrollar la lista de actividades planeadas, según la sección 4.1.2. La iteración continua y la colaboración constante con el Instituto para Niños Ciegos y Sordos del Valle del Cauca, bajo la guía del Dr. Andrés Adolfo Navarro Newball, permitieron la creación de un videojuego interactivo gestual. El uso de herramientas de desarrollo de entornos visuales como medio interactivo fue clave en este proceso.
- Los cuestionarios aplicados a los profesionales del Instituto proporcionaron valiosa retroalimentación, evidenciando una alta aceptabilidad y utilidad del prototipo. Las pruebas realizadas mostraron que el 100% de los profesionales consideraron el sistema útil para evaluar el desarrollo cognitivo de los niños, fácil de usar y mejorado por la implementación de sonidos y efectos visuales. Los comentarios adicionales sugirieron mejoras específicas consideradas para futuras iteraciones del proyecto.
- El código preexistente desarrollado por Martín Vladimir Alonso Sierra resultó ser una ventaja en las etapas de diseño y desarrollo, ya que estaba adecuadamente comentado y creado, siguiendo buenas prácticas de programación. Esto permitió una fácil comprensión y modificación del código para cumplir con los objetivos.
- Con las pruebas se lograron cumplir los requerimientos definidos inicialmente. El prototipo tiene el potencial de apoyar significativamente el proceso de evaluación del desarrollo cognitivo y visomotor de los niños. La investigación realizada y la constante retroalimentación con los profesionales

del Instituto, han validado la viabilidad y efectividad del sistema, estableciendo una sólida base para su implementación y mejora continua.

8.2 Trabajos futuros

- Agregar modificaciones y evaluar comentarios detallados que hicieron los profesionales encargados del Instituto, según la sección 7.4.
- Un colaborador de la salud (psicólogo, profesor, evaluador, pediatra, entre otros) evaluará cómo medir otros elementos como la percepción espacial, la coordinación motora del jugador y la adaptación al sistema.
- Finalmente, es esencial prolongar las pruebas durante un período más extenso para verificar si la aplicación realmente facilita la evaluación y el seguimiento del desarrollo de los niños.

9 Bibliografía

- [1] J. H. Tangarife y Y. V. Nieto Acevedo, «Hemeroteca,» 22 10 2015. [En línea]. Available: <https://hemeroteca.unad.edu.co/index.php/publicaciones-e-investigacion/article/view/1431>.
- [2] innovae, «innovae,» 20 07 2022. [En línea]. Available: <https://www.innovae.com/la-tecnologia-de-realidad-virtual/#:~:text=Las%20primeras%20gafas%20de%20Realidad,usuario%20disfrutar%20de%20experiencias%20multisensoriales..>
- [3] Y. Fernández, «xataka,» 21 04 2021. [En línea]. Available: <https://www.xataka.com/historia-tecnologica/el-primer-simulador-vr-de-la-historia-tenia-forma-de-recreativa-y-se-invento-a-finales-de-los-50>.
- [4] Y. Fernández, «Xataka,» 16 05 2022. [En línea]. Available: <https://www.xataka.com/basics/que-metaverso-que-posibilidades-ofrece-cuando-sera-real>.
- [5] J. L. R. Tamayo, «ResearchGate,» 04 07 2015. [En línea]. Available: https://www.researchgate.net/publication/303472215_El_Continuo_de_la_Virtualidad_o_Virtuality_Continuum_en_el_Marco_de_Convergencia_de_la_Computacion_la_Narrativa_y_los_Diferentes_Sistemas_de_Interaccion_Desarrollo_y_Aproximacion_a_Modelos_para_la_Des.
- [6] Levis, «Scielo,» 2006. [En línea]. Available: https://d1wqtxts1xzle7.cloudfront.net/30471870/que_es_rv-libre.pdf?1391848087=&response-content-disposition=inline%3B+filename%3DQue_es_la_realidad_virtual.pdf&Expires=1695515094&Signature=SUBzRnGgpguDhKdQMjYDqIUySa0VHZq-UX5idme-iLOPhbyqmoMFRboFzi-a8zL4aW.
- [7] M. Farshid, J. Paschen, T. Eriksson y J. Kietzmann, «ScienceDirect,» 09 2018. [En línea]. Available: <https://www.sciencedirect.com/science/article/pii/S000768131830079X>.
- [8] S. Tech, «Smowl Tech,» 29 06 2022. [En línea]. Available: <https://smowl.net/es/blog/test-psicometrico-definicion-tipos-consejos/>.
- [9] P. Williams, L. Weiss y E. Rolfhus, «Pearson Assessments,» 01 06 2003. [En línea]. Available: <https://www.pearsonassessments.com/content/dam/school/global/clinical/us/assets/wisc-iv/wisc-iv-technical-report-1.pdf>.
- [10] S. Valea, «Universidad de Valladolid,» 2013. [En línea]. Available: <https://uvadoc.uva.es/bitstream/handle/10324/4001/TFG-G%20373.pdf;jsessionid=418AEBD4AE2836AD6B2F390E37DCE993?sequence=1>.
- [11] L. C. C. Liñán, «UNIVERSIDAD INCA GARCILASO DE LA VEGA,» 2018. [En línea]. Available: http://repositorio.uigv.edu.pe/bitstream/handle/20.500.11818/3096/TESIS_MAEST_INFOR.APLIC.EDUCA_IVAN%20ANGEL%20ENCALADA%20D%c3%8dAZ_%26_REN%c3%81N%20DELGADO%20ALVA.pdf?sequence=2&isAllowed=y.

- [12] Y. P. Conde, «RI-UMSA,» 2015. [En línea]. Available: <https://repositorio.umsa.bo/handle/123456789/7401>.
- [13] B. H. Capcha, «UTP,» 2021. [En línea]. Available: <https://repositorio.utp.edu.pe/handle/20.500.12867/4605>.
- [14] L. Arias Ferrer, A. Egea Vivancos y A. García López, «Clío,» 2018. [En línea]. Available: <https://papiro.unizar.es/ojs/index.php/cli/article/view/8668>.
- [15] E. tiempo, «El tiempo,» 21 10 2021. [En línea]. Available: <https://www.eltiempo.com/colombia/cali/valle-investigacion-para-ninos-ciegos-y-sordos-626359>.
- [16] S. Álava Sordo, M. Cantero-García, H. Garrido-Hernansaiz, I. Sánchez-Iglesias, J. Santacreu Más y J. González-Moreno, «Dialnet,» 12 02 2021. [En línea]. Available: <https://dialnet.unirioja.es/descarga/articulo/8175339.pdf>.
- [17] R. Crespo, R. Ernesto, G. Ricardo y D. Cárdenas, «Tecnológico de Monterrey,» 2013. [En línea]. Available: <https://repositorio.tec.mx/handle/11285/621360>.
- [18] A. f. C. M. N. Y. N. U. States, «ACM,» 28 07 2019. [En línea]. Available: <https://dl.acm.org/doi/proceedings/10.1145/3306449>.
- [19] Unity, «Unity,» 2024. [En línea]. Available: <https://unity.com/es>.
- [20] K. Gómez, «DiagramasUML,» [En línea]. Available: <https://diagramasuml.com/casos-de-uso/>.
- [21] Unity, «Unity,» [En línea]. Available: <https://docs.unity3d.com/es/530/Manual/CreatingScenes.html>.
- [22] «Interaction Design Foundation,» [En línea]. Available: <https://www.interaction-design.org/literature/topics/virtuality-continuum>.
- [23] P. J. S. Martínez, «Onirix,» 06 2024. [En línea]. Available: <https://www.onirix.com/es/aprende-sobre-ra/que-es-la-realidad-aumentada/>.
- [24] IES, «IES,» 02 05 2018. [En línea]. Available: <https://www3.gobiernodecanarias.org/medusa/edublog/ieselcalero/la-realidad-como-dos-realidades-opuestas/#:~:text=La%20realidad%20f%C3%ADsica%2C%20como%20indica,diferencia%20de%20a%20f%C3%ADsica%2C%20intangible..>
- [25] «SciELO,» [En línea]. Available: <https://www.scielo.sa.cr/pdf/tem/v28n1/0379-3982-tem-28-01-00121.pdf>.
- [26] J. A. Amador Campos y M. F. i Santacana, «University of Barcelona,» 01 2019. [En línea]. Available: <https://diposit.ub.edu/dspace/bitstream/2445/127676/1/WISC-V.pdf>.
- [27] «IMSS,» [En línea]. Available: <https://www.imss.gob.mx/sites/all/statics/pdf/guarderias/discapacidad-sensorial.pdf>.