

**Prototipo de Videojuego formativo para apoyar la pronunciación de palabras en niños de grado primero del Colegio Gimnasio San Diego de Tunja**

**Juan Esteban Niño Gamboa**

**Universidad de Boyacá  
Facultad de Ciencias e Ingeniería  
Ingeniería en Multimedia  
Tunja  
2024**

**Prototipo de Videojuego formativo para apoyar la pronunciación de palabras en niños de grado primero del Colegio Gimnasio San Diego de Tunja**

**Juan Esteban Niño Gamboa**

**Trabajo de grado de Semillero de investigación SAMI para optar al título de:  
Ingeniero en Multimedia**

**Director:**

**Mauricio Ochoa**

**Ing. Sistemas**

**Codirector:**

**Miguel Sanabria Mendivelso**

**Ing. Multimedia**

**Universidad de Boyacá  
Facultad de Ciencias e Ingeniería  
Ingeniería en Multimedia**

**Tunja**

**2024**

**Nota de aceptación**

---

---

---

---

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

Tunja, 20 de mayo de 2024

“Únicamente el graduando es responsable de las ideas expuestas en el presente trabajo”.  
(Lineamientos constitucionales, legales e institucionales que rigen la propiedad intelectual).

### **Agradecimientos**

Agradecimiento especial al docente orientador Ingeniero Miguel Stiven Sanabria Mendivelso, por el apoyo recibido para darle cuerpo a la investigación realizada.

Al colegio gimnasio San Diego de Tunja, en cabeza de su directora Gloria Judith Puerto Estupiñán, los docentes, padres de familia y estudiantes del grado primero por abrir sus puertas y permitir el desarrollo de esta investigación.

**Contenido**

	Pág.
Introducción .....	14
Analizar la mecánica de reconocimiento de voz del prototipo de videojuego formativo.....	15
Habilitación de la funcionalidad para voz.....	15
Uso y programación de la funcionalidad para voz.....	16
Flujo de pantallas .....	19
Descripción de las fases.....	20
Pantalla de Selección de personajes:.....	20
Pantalla de Nivel de Juego:.....	21
Pantalla de pausa:.....	21
Pantalla Final: .....	21
Aplicar la metodología ágil Kanban durante la fase de diseño del prototipo de videojuego formativo.....	22
Creación de tablero Kanban y tareas.....	22
Preproducción.....	24
Sinopsis del Juego: .....	24
Diseños y Bocetos: .....	25
Elaboración del mundo para la prueba piloto. ....	25
Personaje para la prueba piloto. ....	26
Producción.....	28
Prueba Piloto .....	29
Crear un prototipo de videojuego formativo mediante pautas dadas en la fase de producción....	31
Personajes.....	31
Conceptualización y Boceto .....	31
Diseño Detallado .....	32
Modelado 3D.....	32
Texturización.....	33
Escenario .....	34
Fase Blocking .....	34

Fase detalle .....	34
Heads up display (HUDS).....	35
Aspectos técnicos .....	37
Hardware y software de desarrollo.....	37
Evaluar la funcionalidad del prototipo de videojuego formativo en la fluidez de lectura con los niños de grado primero .....	48
Grado de dificultad en el uso de los controles del videojuego.....	49
Grado de aceptación del ambiente .....	52
Grado de comunicación de la interfaz.....	55
Conclusiones .....	58
Recomendaciones .....	59
Referencias.....	60
Anexos .....	61

**Lista de Tablas**

	Pág.
Tabla 1. Grado de dificultad en el uso de los controles del videojuego .....	49
Tabla 2. Grado de aceptación del ambiente .....	52
Tabla 3. Grado de comunicación de la interfaz .....	55



**Lista de Figuras**

	<b>Pág.</b>
Figura 1. Pasos 1 y 2 programación del reconocimiento de voz .....	16
Figura 2. Pasos 3-4-5 programación del reconocimiento de voz.....	17
Figura 3. Paso 6 programación del reconocimiento de voz.....	17
Figura 4. Prueba programación meánica de voz.....	18
Figura 5. Script de programación de reconocimeinto de voz .....	19
Figura 6. Esquema wireframes y wireflows .....	20
Figura 7. Tablero de la metodología kanban .....	24
Figura 8. Blocking Primer mundo para prueba Piloto .....	25
Figura 9. Mundo prueba piloto integrado en Unity .....	26
Figura 10. Personaje prueba piloto .....	27
Figura 11. Scripts utilizados para la prueba piloto .....	28
Figura 12. Gameplay prueba piloto .....	29
Figura 13. Boceto de personajes .....	31
Figura 14. Detallado de personajes.....	32
Figura 15. Blocking modelado 3D personajes.....	33
Figura 16. Texturizado de personajes 3D .....	33
Figura 17. Blocking escenario prototipo.....	34
Figura 18. Detallado y texturizado escenario prototipo.....	35
Figura 19. HUDS de pantalla del prototipo .....	36
Figura 20. Pantalla inicial prototipo de Videojuego .....	38
Figura 21. Script de programación botón inicio .....	38
Figura 22. Script de programación botón salida .....	39
Figura 23. Pantalla Selección de Personaje .....	40
Figura 24. Script programación pantalla de selección de personajes .....	41
Figura 25. Pantalla Nivel Principal.....	42
Figura 26. Script de activación de mensajes de ayuda.....	43
Figura 27. Script programación movimiento del personaje pantalla principal .....	44
Figura 28. Pantalla de pausa .....	45

Figura 29. Script de programación pantalla de pausa .....	46
Figura 30. Pantalla final .....	47
Figura 31. Gráfica de barras uso de mando .....	50
Figura 32. Gráfica de barras ubicación controles .....	50
Figura 33. Gráfica de barras uso de controles – ordenador .....	51
Figura 34. Gráfica de barras colorimetría del escenario .....	53
Figura 35. Gráfica de barras aceptación del personaje .....	53
Figura 36. Gráfica de barras aceptación del escenario .....	54
Figura 37. Gráfica de barras mecánica de lectura de palabras.....	56
Figura 38. Gráfica de barras entendimiento instrucciones de pantalla .....	56
Figura 39. Gráfica de barras entendimiento activación mecánica de voz.....	57

**Lista de Anexos**

	Pág.
Anexo A. Anteproyecto .....	62
Anexo B. Encuestas Completadas .....	89
Anexo C. Consentimientos completados .....	90
Anexo D. Galería de fotografías .....	91
Anexo E. Video prueba Piloto .....	92
Anexo F. Formato Consentimiento.....	93
Anexo G. Audio entrevista prueba piloto .....	94
Anexo H. Carta de aceptación de investigación .....	95
Anexo I. Carta de autorización - Bioética UDB .....	96
Anexo J. Prototipo de videojuego.....	97

## Resumen

### **Prototipo de Videojuego formativo para apoyar la pronunciación de palabras en niños de grado primero del Colegio Gimnasio San Diego de Tunja:**

El siguiente documento muestra los resultados de la investigación realizada bajo la supervisión del Semillero de investigación en Animación, aplicaciones y ambientes Multimedia Interactivos (SAMI), orientada al análisis de la mecánica de reconocimiento de voz, el diseño y creación de un prototipo de videojuego formativo aplicando la metodología ágil Kanban.

Durante la investigación se desarrolla un prototipo de Videojuego formativo para el fortalecimiento de la correcta pronunciación de palabras en niños de grado primero del Colegio Gimnasio San Diego de Tunja, en el diseño se hace uso de las pautas dadas por el usuario final.

La metodología utilizada en este proyecto es la investigación mixta, pues permite el estudio de la calidad de las actividades, las relaciones, los asuntos, los medios, los materiales o los instrumentos en una determinada situación. Como herramienta de recolección de datos se hace uso de entrevistas para describir los pensamientos, opiniones y sentimientos de los usuarios.

Como conclusión principal se define la correcta aceptación del prototipo de videojuego formativo frente a la usabilidad, accesibilidad y diseño, a su vez se da como recomendación la mejora y ampliación del nivel principal, la implementación de personajes secundarios que ayuden dando nuevas instrucciones y ayudas al usuario final, por otra parte, la implementación de jefes de nivel que aporten a la superación de un reto mayor, que genere una recompensa igualmente mayor al usuario final.

**Palabras claves:** Prototipo, Videojuego formativo, Script, Unity, reconocimiento de voz

### **Abstract**

#### **Formative video game prototype to support the pronunciation of words in first grade children of Colegio Gimnasio San Diego, Tunja:**

The following document shows the results of the research conducted under the supervision of Semillero de investigación en Animación, aplicaciones y ambientes Multimedia Interactivos (SAMI), oriented to the analysis of the mechanics of speech recognition, design, and creation of a prototype of an educational video game applying the agile Kanban methodology.

During the research a prototype of a formative video game is developed for the strengthening of the correct pronunciation of words in first grade student from Colegio Gimnasio San Diego, Tunja, the design makes use of the guidelines given by the end user.

The methodology used in this project is mixed research, since it allows the study of the quality of activities, relationships, issues, means, materials, or instruments in a specific situation. As a data collection tool, interviews are used to describe the thoughts, opinions, and feelings of the users.

As main conclusion, is mentioned the correct acceptance of the formative video game prototype in terms of usability, accessibility and design, at the same time as recommendations it gives the improvement and expansion of the main level, the implementation of secondary characters that help giving new instructions and help to the end user, on the other hand, the implementation of level bosses that contribute to the overcoming of a greater challenge, which generates an equally greater reward to the end user.

**Keywords:** Prototype, Training video game, Script, Unity, voice recognition.

## Introducción

El presente documento de tesis de semillero tuvo como objetivo principal describir el proceso de desarrollo de un prototipo de videojuego formativo, orientado a la correcta pronunciación de palabras en niños de grado primero del Colegio Gimnasio San Diego de Tunja. La importancia de fomentar habilidades comunicativas en la etapa escolar temprana ha sido ampliamente reconocida junto con el uso de herramientas tecnológicas, como los videojuegos formativos, que son una alternativa innovadora y atractiva para abordar esta necesidad.

La elección de enfocar el desarrollo de un videojuego formativo como herramienta para el fortalecimiento de la correcta pronunciación en niños de grado primero se fundamentó en la relevancia de esta etapa en el proceso de adquisición de habilidades comunicativas, así como en la oportunidad de aprovechar el potencial educativo y motivador que ofrecen los medios digitales interactivos.

El Colegio Gimnasio San Diego de Tunja se ha caracterizado por su compromiso con la excelencia académica y el desarrollo integral de sus estudiantes, y el presente proyecto surge como respuesta a la necesidad identificada por el cuerpo docente de fortalecer la correcta pronunciación de palabras en el primer grado a través de herramientas innovadoras y atractivas para los niños.

A lo largo de este documento se abordaron los fundamentos teóricos que sustentan la importancia de la correcta pronunciación en la educación inicial, así como el papel de los videojuegos formativos como herramientas pedagógicas. Se describirá el proceso de diseño y desarrollo del prototipo de videojuego formativo, destacando las etapas, metodologías y tecnologías empleadas. Además, se presentarán los resultados obtenidos a partir de pruebas piloto realizadas con los estudiantes, así como las conclusiones y recomendaciones derivadas de la implementación del prototipo.

### **Analizar la mecánica de reconocimiento de voz del prototipo de videojuego formativo**

Según el motor de videojuegos Unity (Microsoft Build, 2021) existen tres maneras de agregar entrada de voz a la aplicación. La primera es de tipo `PhraseRecognizer`, que le permite a la aplicación escuchar una frase completa. La segunda, `KeywordRecognizer` que permite a la aplicación escuchar una matriz de comandos de cadena predefinidos. Y la tercera, `GrammarRecognizer`, que le permite a la aplicación utilizar un archivo SRGS para definir una gramática específica que pueda ser escuchada.

Por otro lado, el `DictationRecognizer` permite a la aplicación escuchar cualquier palabra y proporcionar al usuario una nota u otra presentación de su voz, brindando así una mayor flexibilidad en la interacción con la aplicación (Microsoft Build, 2021)

Estas librerías son completamente desarrolladas por Microsoft, lo que significa que la base de datos para el reconocimiento del lenguaje es proporcionada por el ordenador o computador. Esto implica que, al buscar una palabra en español o inglés, el programa la reconocerá de forma predeterminada gracias a la base de datos proporcionada por Microsoft.

### **Habilitación de la funcionalidad para voz**

Para que esta funcionalidad este activada y el usuario la pueda usar de manera más cómoda Microsoft establece los siguientes pasos para el motor de videojuegos de Unity (Microsoft Build, 2021):

1. En el Editor de Unity, vaya a **Editar > reproductor de configuración del > proyecto**.
2. Seleccione la pestaña **Tienda Windows**.
3. En la sección **Funcionalidades de configuración > de publicación**, compruebe la funcionalidad **Micrófono**.
4. Concesión de permisos a la aplicación para el acceso de micrófono en el dispositivo HoloLens
  - Se le pedirá que lo haga en el inicio del dispositivo, pero si ha hecho clic accidentalmente en "no", puede cambiar los permisos en la configuración del dispositivo.

Por lo tanto, el último paso se omite ya que este serious game será desarrollado para ordenadores y no se necesitará la autorización de uso de micrófono solo que el usuario si tenga el hardware necesario para el uso de esta funcionalidad, es decir, un micrófono conectado al ordenador.

### Uso y programación de la funcionalidad para voz

Para una correcta implementación y programación de la funcionalidad se usaron 6 pasos que fueron dados por la empresa Microsoft Build (2021), los cuales se pueden observar en las siguientes imágenes:

#### Figura 1

*Pasos 1 y 2 programación del reconocimiento de voz*

## KeywordRecognizer

Namespace: *UnityEngine.Windows.Speech*

Types: *KeywordRecognizer, PhraseRecognizedEventArgs, SpeechError, SpeechSystemStatus*

We'll need a few using statements to save some keystrokes:

```
using UnityEngine.Windows.Speech;
using System.Collections.Generic;
using System.Linq;
```

Then let's add a few fields to your class to store the recognizer and keyword->action dictionary:

```
KeywordRecognizer keywordRecognizer;
Dictionary<string, System.Action> keywords = new Dictionary<string, System.Action>();
```

Fuente: Tomado de Microsoft Build.



**Figura 2***Pasos 3-4-5 programación del reconocimiento de voz.*

Now add a keyword to the dictionary, for example in of a `Start()` method. We're adding the "activate" keyword in this example:

```
Copy  
  
//Create keywords for keyword recognizer  
keywords.Add("activate", () =>  
{  
    // action to be performed when this keyword is spoken  
});
```

Create the keyword recognizer and tell it what we want to recognize:

```
Copy  
  
keywordRecognizer = new KeywordRecognizer(keywords.Keys.ToArray());
```

Now register for the `OnPhraseRecognized` event

```
Copy  
  
keywordRecognizer.OnPhraseRecognized += KeywordRecognizer_OnPhraseRecognized;
```

Fuente: Tomado de Microsoft Build.

**Figura 3***Paso 6 programación del reconocimiento de voz.*

Finally, start recognizing!

```
Copy  
  
keywordRecognizer.Start();
```

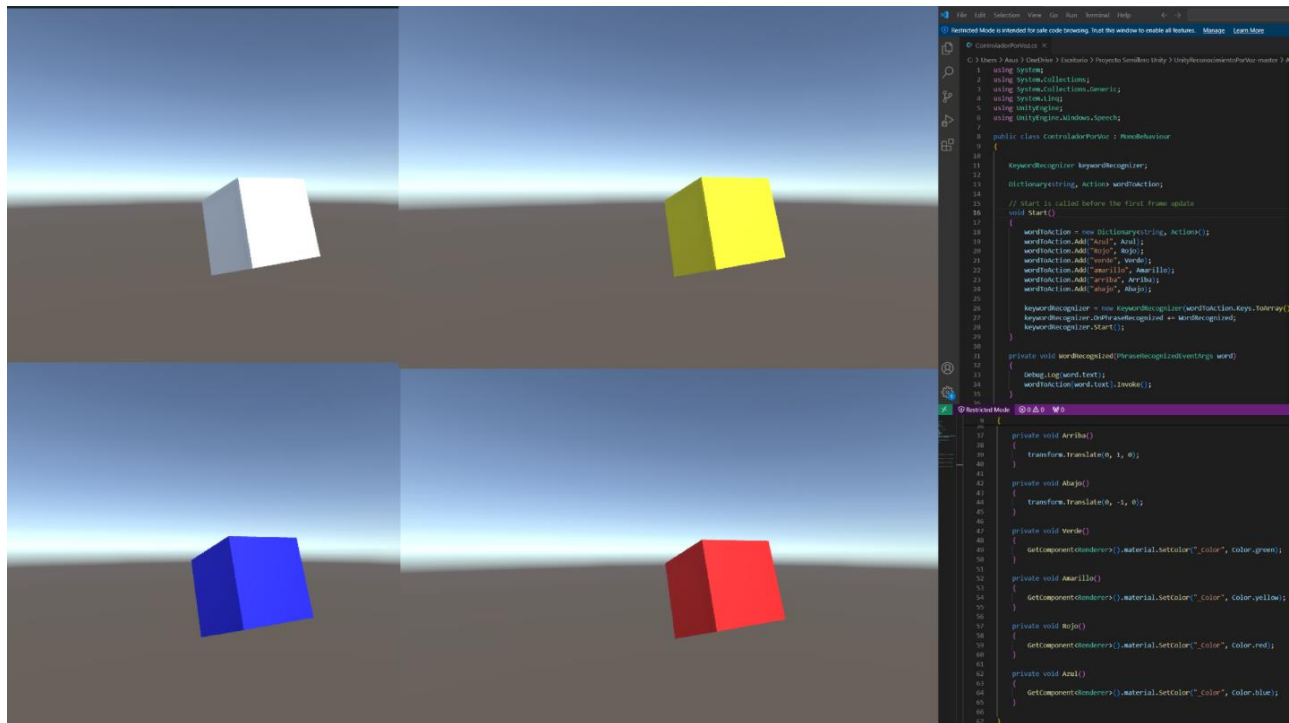
Fuente: Tomado de Microsoft Build.

Con el fin de verificar la funcionalidad sobre la mecánica de reconocimiento de voz del

prototipo de videojuego y que el usuario pueda desarrollar los distintos puzzles mientras refuerza la lectura, se realizó la prueba de los Scripts citados y se obtuvo la programación, como se evidencia en la figura 4.

**Figura 4**

*Prueba programación mecánica de voz*



Fuente: Elaboración propia

En el motor de videojuegos Unity, se requirió del script de programación en el lenguaje nativo de Unity C#, para el prototipo de videojuego realizado, el reconocimiento de voz modifica la apariencia de diferentes objetos, cuando el usuario cumple a cabalidad con la instrucción dada en el videojuego.

La figura 4 muestra la creación del diccionario que usó las bases de datos nativas de Microsoft para modificar el material del Objeto creado en el motor de Unity, en las pruebas realizadas se obtuvo como resultado la correcta implementación, ya que al pronunciar los distintos colores programados en el diccionario se denotó la correcta funcionalidad e implementación de esta mecánica para el prototipo de videojuego. En la figura 5 se presenta script del prototipo de

videojuego para el reconocimiento de voz.

**Figura 5**

*Script de programación de reconocimiento de voz.*

```

1 using System;
2 using System.Collections;
3 using System.Collections.Generic;
4 using System.Linq;
5 using UnityEngine;
6 using UnityEngine.Windows.Speech;
7
8 public class ControladorPorVoz : MonoBehaviour
9 {
10     KeywordRecognizer keywordRecognizer;
11     Dictionary<string, Action> wordToAction;
12
13     // Start is called before the first frame update
14     void Start()
15     {
16         wordToAction = new Dictionary<string, Action>();
17         wordToAction.Add("Azul", Azul);
18         wordToAction.Add("Rojo", Rojo);
19         wordToAction.Add("Verde", Verde);
20         wordToAction.Add("Amarillo", Amarillo);
21         wordToAction.Add("Arriba", Arriba);
22         wordToAction.Add("Abajo", Abajo);
23
24         keywordRecognizer = new KeywordRecognizer(wordToAction.Keys.ToArray());
25         keywordRecognizer.OnPhraseRecognized += WordRecognized;
26         keywordRecognizer.Start();
27     }
28
29     private void WordRecognized(PhraseRecognizedEventArgs word)
30     {
31         Debug.Log(word.text);
32         wordToAction[word.text].Invoke();
33     }
34 }
35
36
37 private void Arriba()
38 {
39     transform.Translate(0, 1, 0);
40 }
41
42 private void Abajo()
43 {
44     transform.Translate(0, -1, 0);
45 }
46
47 private void Verde()
48 {
49     GetComponent<Renderer>().material.SetColor("_color", color.green);
50 }
51
52 private void Amarillo()
53 {
54     GetComponent<Renderer>().material.SetColor("_color", color.yellow);
55 }
56
57 private void Rojo()
58 {
59     GetComponent<Renderer>().material.SetColor("_color", color.red);
60 }
61
62 private void Azul()
63 {
64     GetComponent<Renderer>().material.SetColor("_color", color.blue);
65 }
66
67
68
    
```

Fuente: Elaboración propia

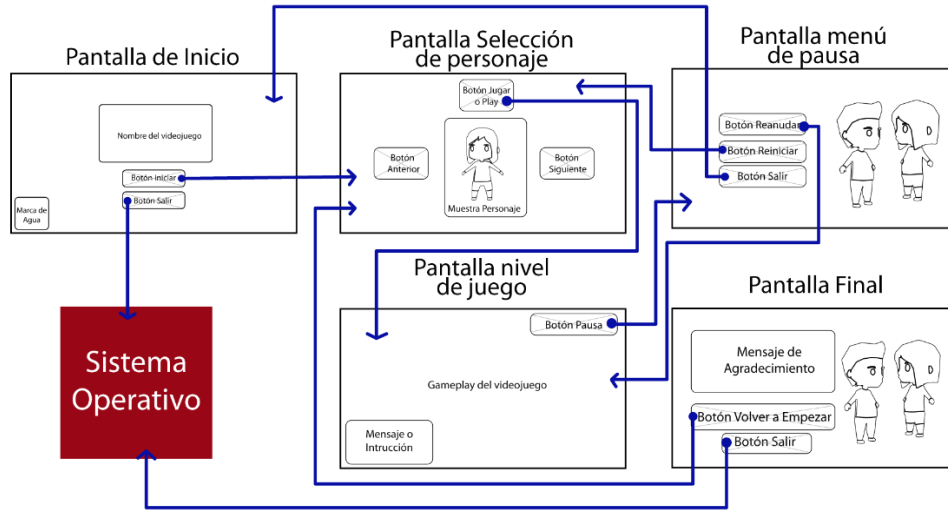
### Flujo de pantallas

En el contexto de la creación del prototipo de videojuego formativo, el diseño y la estructura del flujo de pantallas jugaron un papel fundamental en la experiencia de aprendizaje del jugador y fue una herramienta fundamental en el desarrollo del prototipo (Miro, s.f.). El flujo de pantallas abarcó desde la pantalla de inicio hasta la pantalla final y actuó como el hilo conductor que guio al usuario a través de las diferentes etapas del juego educativo.

Cada pantalla no sólo cumplió una función específica en términos de interacción y presentación de contenido, sino que también contribuyó a la inmersión del jugador en el entorno virtual, facilitando la asimilación de conocimientos de manera dinámica y entretenida. En la figura 6 se presenta el esquema de flujo de pantallas, que orientó el uso y la programación del videojuego a través del esquema de wireframes y wireflows.

**Figura 6**

*Esquema wireframes y wireflows*



Fuente: Elaboración propia

**Descripción de las fases**

El prototipo de videojuego inicia con una pantalla de introducción la cual le da al usuario la opción de iniciar el juego o de salir. Las siguientes pantallas tienen un funcionamiento distinto que se relata a continuación.

**Pantalla de selección de personajes.** En esta pantalla el usuario tuvo la posibilidad de seleccionar el personaje de su preferencia. Pudo seleccionar una de las tres opciones preestablecidas.

**Pantalla de nivel de juego.** En esta pantalla el usuario tuvo la mayor interacción con el videojuego, es a través de esta pantalla en la que se desarrollaron las diferentes mecánicas planteadas: tanto las mecánicas básicas de movimiento como las mecánicas especializadas de activación por voz y resolución de puzzles. A su vez tiene el botón que activa la pantalla de pausa y una vez finalizadas todas las pruebas guía al usuario a la pantalla final.

**Pantalla de pausa.** Esta pantalla se puede activar desde la pantalla de nivel de juego y queda superpuesta, para que el usuario pueda salir al sistema operativo (escritorio del computador), reiniciar la partida y continuar la partida actual en el punto en que fue pausada.

**Pantalla Final.** En ella el usuario encontrará un mensaje de agradecimiento y las opciones de reiniciar el nivel de juego o salir al sistema operativo.

### **Aplicar la metodología ágil Kanban durante la fase de diseño del prototipo de videojuego formativo**

La metodología ágil Kanban se ha convertido en una herramienta popular para la gestión eficiente de proyectos, y su aplicación en el desarrollo de videojuegos resulta de gran interés en el ámbito académico y profesional (Martins, 2024). Razón por la cual, se usó esta metodología tanto en el diseño del prototipo de videojuego como para su desarrollo y programación, ya que permitió el uso y el cambio de prioridad según las tareas requeridas para el proyecto.

Además, se utilizó una metodología iterativa que implicó un enfoque flexible y continuo en el proceso de investigación y desarrollo (Jimbeas, 2019). Esto significa que se usaron ciclos de diseño, implementación, evaluación y retroalimentación, con el objetivo de mejorar y refinar el prototipo en función de los resultados obtenidos. Los comentarios de los niños, los educadores y los expertos en el campo de la educación y el desarrollo infantil fueron fundamentales en este proceso iterativo.

Para las fases de diseño del prototipo de videojuego se creó un tablero Kanban en la plataforma Trello que permitió la creación y gestión de una manera organizada, el cual se presenta en la figura 7.

#### **Creación de tablero Kanban y tareas**

El tablero contó con las columnas Diseño, Por hacer, Desarrollando, Revisión de tarea, pruebas y finalizado, cumplió una función en concreto para desarrollar en el ciclo de diseño, se hizo uso eficiente del tiempo y se realizaron los cambios pertinentes. Se crearon las siguientes tareas para el diseño y desarrollo del prototipo:

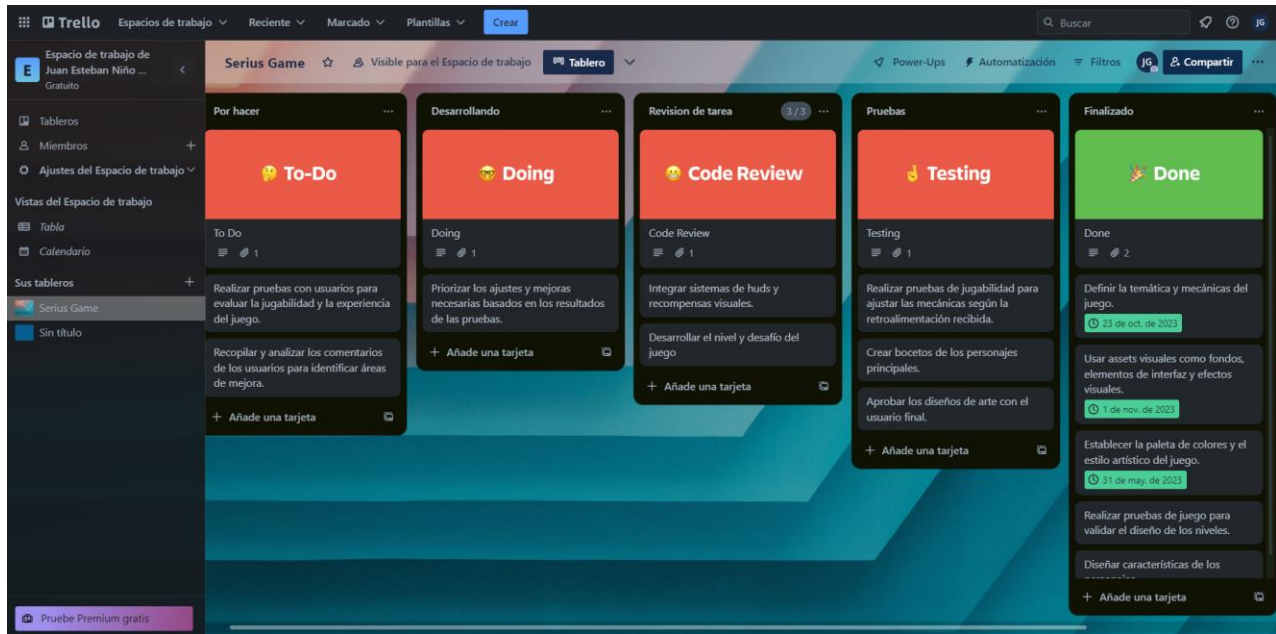
1. Ideación:
  - Temática y mecánicas del juego.
2. Diseño de personajes:
  - Creación de bocetos de los personajes principales.
  - Diseño de características de los personajes.
  - Retroalimentación del usuario final y ajustes.
3. Diseño de niveles:

- Niveles y desafíos del juego.
  - Pruebas de juego para validar el diseño de los niveles.
4. Desarrollo de arte y gráficos:
- Uso de assets visuales como fondos, elementos de interfaz y efectos visuales.
  - Establecer la paleta de colores y el estilo artístico del juego.
  - Aprobar los diseños de arte con el usuario final.
5. Implementación de mecánicas de juego:
- Programar la funcionalidad de movimiento y controles del personaje.
  - Integrar sistemas de HUDs y recompensas visuales.
  - Realizar pruebas de jugabilidad para ajustar las mecánicas según la retroalimentación recibida.
6. Pruebas de usabilidad:
- Realizar pruebas con usuarios para evaluar la jugabilidad y la experiencia del juego.
  - Recopilar y analizar los comentarios de los usuarios para identificar áreas de mejora.
  - Priorizar los ajustes y mejoras necesarias basados en los resultados de las pruebas.

La figura 7 ilustra el tablero de control para el desarrollo del proyecto de investigación y cada una de sus fases.

**Figura 7**

*Tablero de la metodología Kanban*



Fuente: Elaboración propia usando plataforma web Trello

En resumen, en la fase de preproducción se establecieron los conceptos, diseños y mecánicas del videojuego; en la fase de producción se implementó los elementos visuales, sonoros y de programación; y en la fase de postproducción, se optimizó, depuró y mejoró el juego.

Una vez elaborado el prototipo inicial del videojuego mediante el uso de Assets (escenarios y personajes) gratuitos de la biblioteca de Unity (AssetStore, s.f.), se procedió a desarrollar una prueba piloto para implementar cambios según lo referido durante la prueba.

A continuación, se describen las fases mencionadas y los aspectos desarrollados en cada una de ellas.

**Preproducción**

*Sinopsis del juego*

El videojuego formativo es una experiencia única que fusionó la resolución de puzzles, la



narrativa emotiva y la interacción de voz, brindó a los jugadores una experiencia de juego intuitiva y divertida en un mundo mágico lleno de misterios por descubrir, en el que la pronunciación es una habilidad comunicativa fundamental.

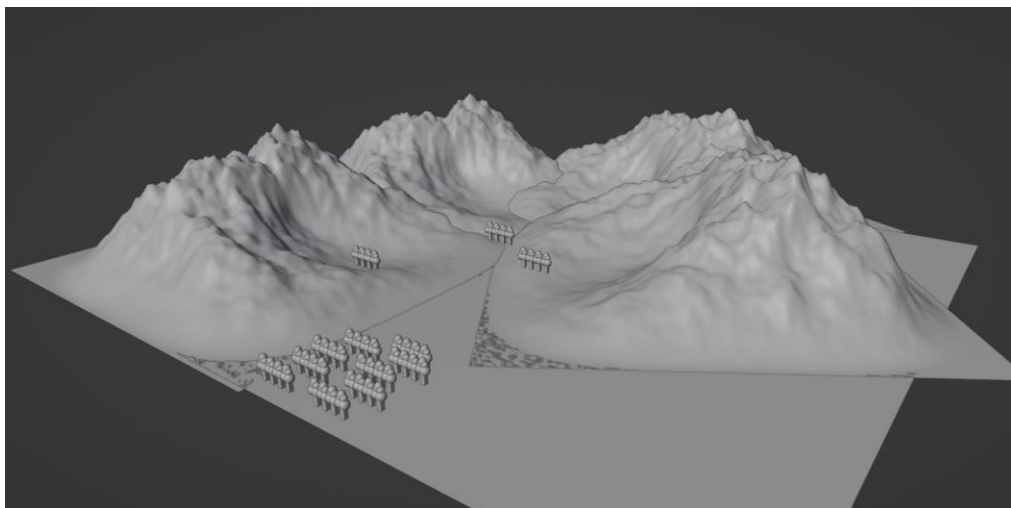
### *Diseños y bocetos*

Para que el usuario tenga un mayor entendimiento del videojuego se utilizaron assets (escenarios y personajes) de uso gratuito antes de las pruebas piloto, una vez realizada la prueba se elaboraron diseños y bocetos de personajes, escenarios y elementos interactivos que representaron el universo del videojuego. Estos diseños buscaban combinar elementos visuales atractivos con funcionalidades educativas, priorizando la claridad y la coherencia con las habilidades comunicativas del usuario, como se ve a continuación:

**Elaboración del mundo para la prueba piloto.** Este es el escenario en el que el usuario va a interactuar principalmente desde el videojuego, en el que se encuentran los diferentes niveles y retos. El escenario inicial consta de un paisaje a manera de bosque, que se puede observar en las figuras 8 y 9.

### **Figura 8**

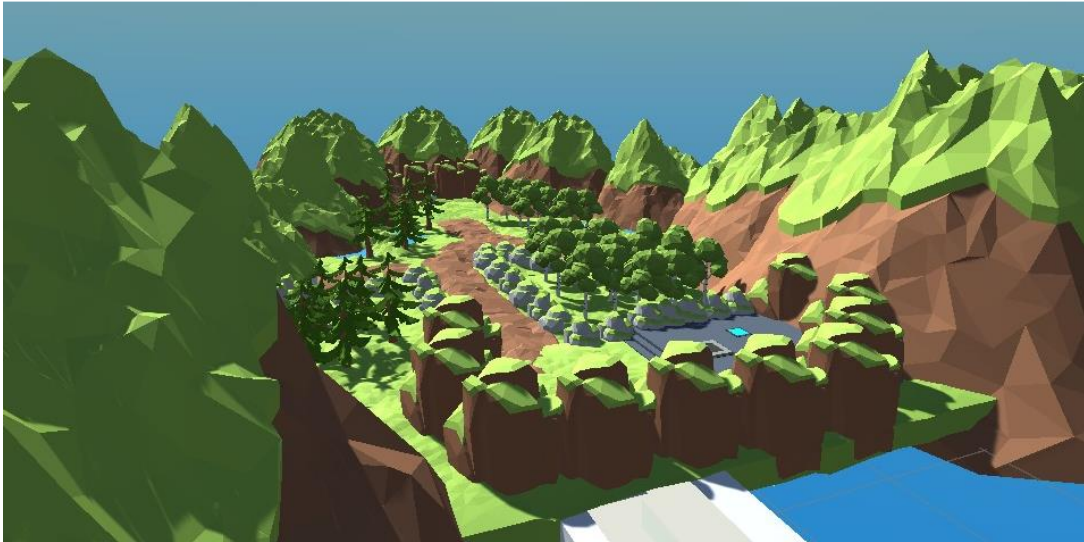
*Blocking Primer mundo para prueba Piloto*



Fuente: Elaboración propia

**Figura 9**

*Mundo prueba piloto integrado en Unity*

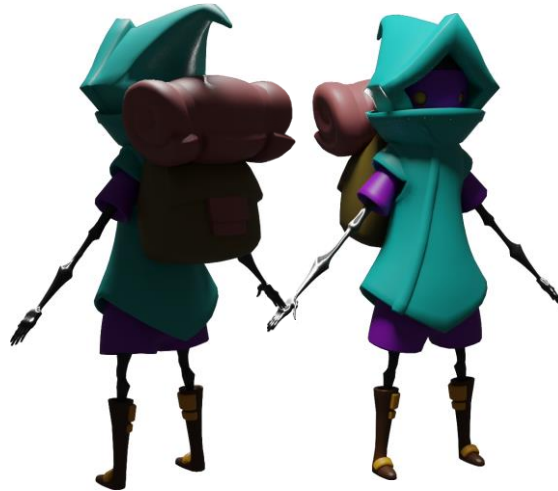


Fuente: Biblioteca digital AssetStore de Unity

**Personaje para la prueba piloto.** En la generación del videojuego desde un mundo fantástico, se seleccionó un personaje base del motor de videojuegos Unity, y que guarda coherencia con el escenario seleccionado. Esta elección se realizó tomando como premisa la convicción del estilo fantástico del usuario final, el personaje seleccionado fue el relacionado en la figura 10.

**Figura 10**

*Personaje prueba piloto*



Fuente: Biblioteca digital AssetStore de Unity

Una vez seleccionado el personaje, se procedió a delimitar las mecánicas del juego, estas fueron investigadas y programadas con el propósito de fomentar el refuerzo en la correcta pronunciación de los usuarios finales mediante un ambiente divertido e intuitivo, por lo anterior, las mecánicas definidas para este videojuego formativo fueron:

- **Comandos de Voz:** Los jugadores utilizaron comandos de voz para interactuar con el entorno, activar interruptores y resolver acertijos. Por ejemplo, pudieron decir "rojo" para activar un objeto del puzle.
- **Rompecabezas Ambientales:** El entorno tenía un rompecabezas que requirió de observación y lógica. Los jugadores pudieron encontrar objetos en el escenario y debieron descifrar su significado utilizando comandos de voz para resolver el puzle para así seguir avanzando. Por ejemplo: unido a la dinámica anterior el usuario debió ubicar el objeto (cubo) en una plataforma para poder cumplir con el rompecabezas.
- **Movimiento básico:** Esta mecánica se refiere a el movimiento que tuvo el personaje durante el uso del prototipo de videojuego formativo, los usuarios podían caminar por el entorno libremente y podían manejar la dirección de la cámara a voluntad; haciendo uso de las teclas: W o Flecha hacia arriba para ir hacia adelante,

S o Flecha hacia abajo para ir hacia atrás, D o Flecha hacia la derecha para ir a la derecha y A o Flecha para la izquierda para ir a la izquierda y mouse para el control horizontal de la cámara del videojuego o del control externo el joystick izquierdo para mover el personaje en distintas direcciones.

- **Agarre de Objetos:** Esta mecánica le permitió al usuario llevar objetos a las diferentes plataformas de puzzles propuestas en el nivel principal.

**Producción**

Durante esta fase se inició la programación de las mecánicas básicas de movimiento del personaje, la mecánica del reconocimiento de voz y los puzzles como se muestra a continuación:

**Figura 11**

*Scripts utilizados para la prueba piloto*



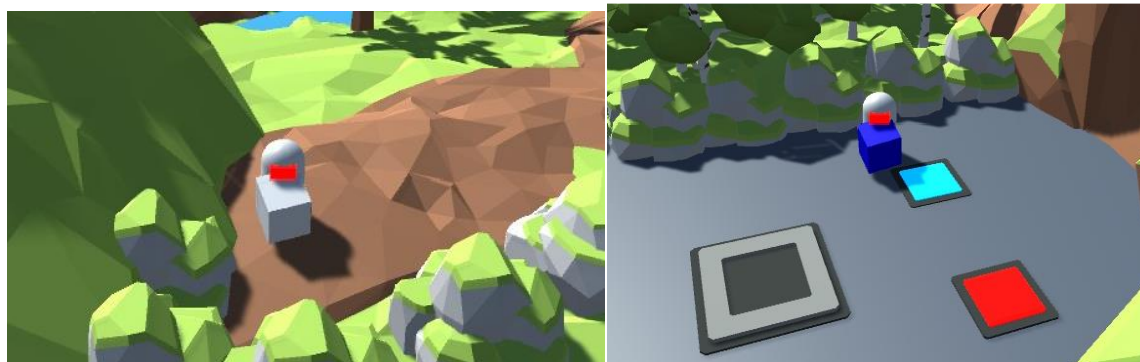
Fuente: Elaboración propia

En la figura anterior se observan los diferentes scripts de programación usados para la funcionalidad del prototipo, estos aspectos serán enfatizados en el capítulo 3 del presente trabajo.

Como resultado se obtuvo el videojuego inicial, en el que se verificaron la funcionalidad de las diferentes mecánicas programadas ya previamente mencionadas.

**Figura 12**

*Gameplay prueba piloto*



Fuente: Elaboración propia

**Prueba Piloto**

Durante esta fase se realizó la observación de la interacción del usuario final con el videojuego, y se recopiló la retroalimentación. Como resultado de esta evaluación se realizaron ajustes y mejoras en la funcionalidad, usabilidad y accesibilidad del prototipo del videojuego formativo en relación con lo referido por el usuario.

Para esta fase se realizó entrevista guiada con preguntas para la obtención de información acerca del videojuego, con las siguientes variables:

- Controles de uso: Grado de dificultad en el uso de los controles del videojuego (teclado y mouse o mando externo).
- Aceptación del ambiente en el que se desarrolla el videojuego. Referido al escenario o mundo y al personaje.
- Comunicación de la interfaz con el usuario: comprensión de los mensajes que aparecen en pantalla del prototipo.

Como resultado de la prueba piloto se encontraron los siguientes aspectos por modificar: Cambio del ambiente a una forma similar a la del colegio (Gimnasio San Diego) y que el personaje tuviera una similitud a un estudiante (opción de seleccionar niño o niña) que usara el uniforme de la institución. De otro lado, se identificó que el usuario requiere movilizar la cámara para mejorar la exploración del mundo e indicaciones o textos que desde su aspecto faciliten la identificación

de las instrucciones.

Respecto al uso de los comandos para la movilidad del personaje, el usuario de una manera autónoma entendió el uso de los controles para el movimiento e interacción tanto con el teclado como con el control de consola para la exploración del ambiente o mundo y el desarrollo de las pruebas del nivel.

Por otro lado, desde la observación del investigador se pudo confirmar la mecánica del reconocimiento de voz. Como evidencia de la aplicación de dicha prueba piloto se adjunta el Anexo E. Video Prueba piloto.

**Crear un prototipo de videojuego formativo mediante pautas dadas en la fase de producción**

En este capítulo se presentan los ajustes realizados en el prototipo inicial del videojuego formativo con base en lo referido por el usuario final en la prueba piloto. Los ajustes realizados se relacionan a continuación.

**Personajes**

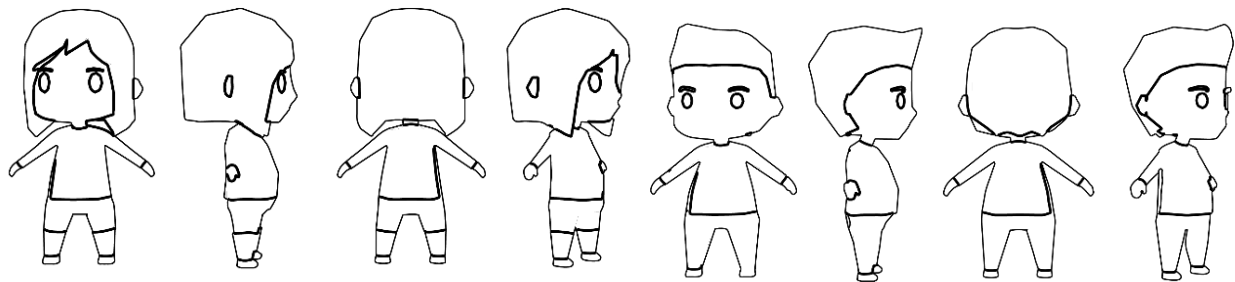
El proceso de creación de un personaje 3D, desde el boceto inicial hasta el modelo 3D, implicó varios pasos importantes, cada paso de este proceso se describe en detalle a continuación al igual que la forma en que se dio respuesta a las observaciones del usuario final referidas en la prueba piloto.

***Conceptualización y Boceto***

Durante esta etapa inicial, el personaje se conceptualizó mediante bocetos hechos a mano o digitales. Estos bocetos sirvieron como base para visualizar la apariencia general, la personalidad y los detalles estéticos del personaje. En la figura 13 se muestran las dos versiones de niño y niña que se generaron según lo referido en la entrevista con el usuario final.

**Figura 13**

*Boceto de personajes*



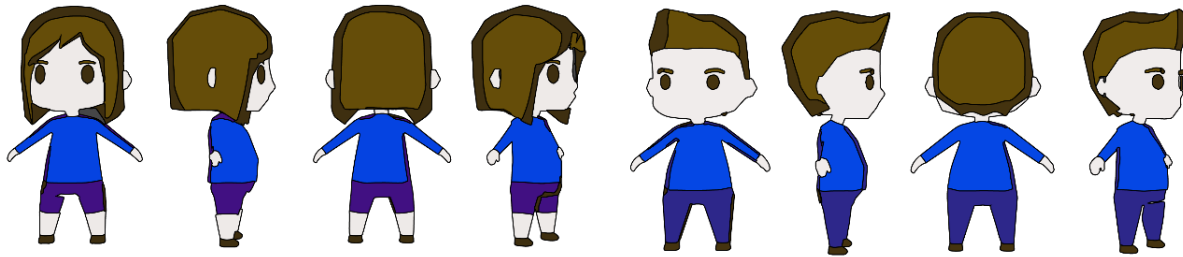
Fuente: Elaboración propia

### *Diseño Detallado*

A partir de bocetos iniciales, se creó un diseño de personaje detallado. Por lo tanto, se detallaron aspectos como: anatomía, vestimenta, accesorios, color y todas las características importantes para su identidad visual. Para este proyecto, se usaron elementos clave de la institución educativa Gimnasio San Diego de la ciudad de Tunja.

### **Figura 14**

#### *Detallado de personajes*



Fuente: Elaboración propia

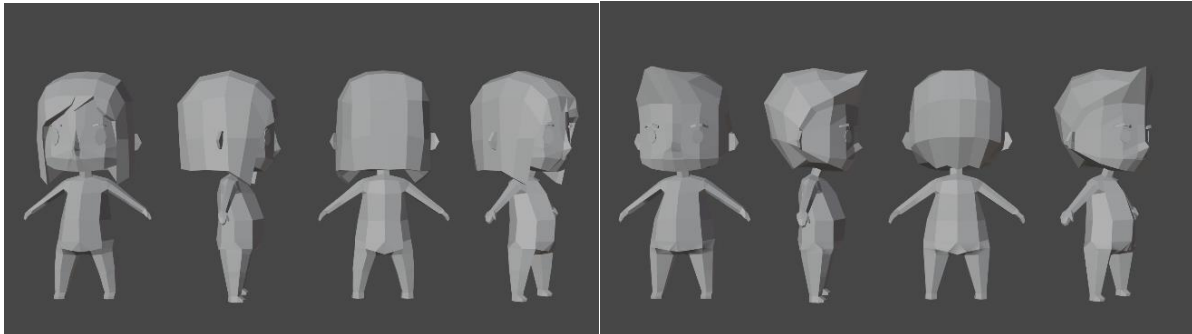
### *Modelado 3D*

Después de realizar el diseño detallado, comenzó el proceso de modelado de personajes en 3D. Este paso implicó la creación de la presentación en 3D utilizando el software especializado Blender el cual como lo dice su página oficial “Con Blender, es posible crear visualizaciones 3D, imágenes fijas, animaciones 3D y efectos visuales.” (Blender.org, s.f.) y por esto mediante la construcción de mallas poligonales se crearon los personajes adicionales para el prototipo. Identificando los diversos perfiles y ángulos en los que se puede ver el personaje.



**Figura 15**

*Blocking modelado 3D personajes*



Fuente: Elaboración propia

***Texturización***

Una vez modelado, se añadieron las texturas al personaje, es decir, color, detalle y realismo. Este proceso implicó dibujar texturas a mano o utilizar técnicas de mapeo para lograr el aspecto deseado.

**Figura 16**

*Texturizado de personajes 3D*



Fuente: Elaboración propia

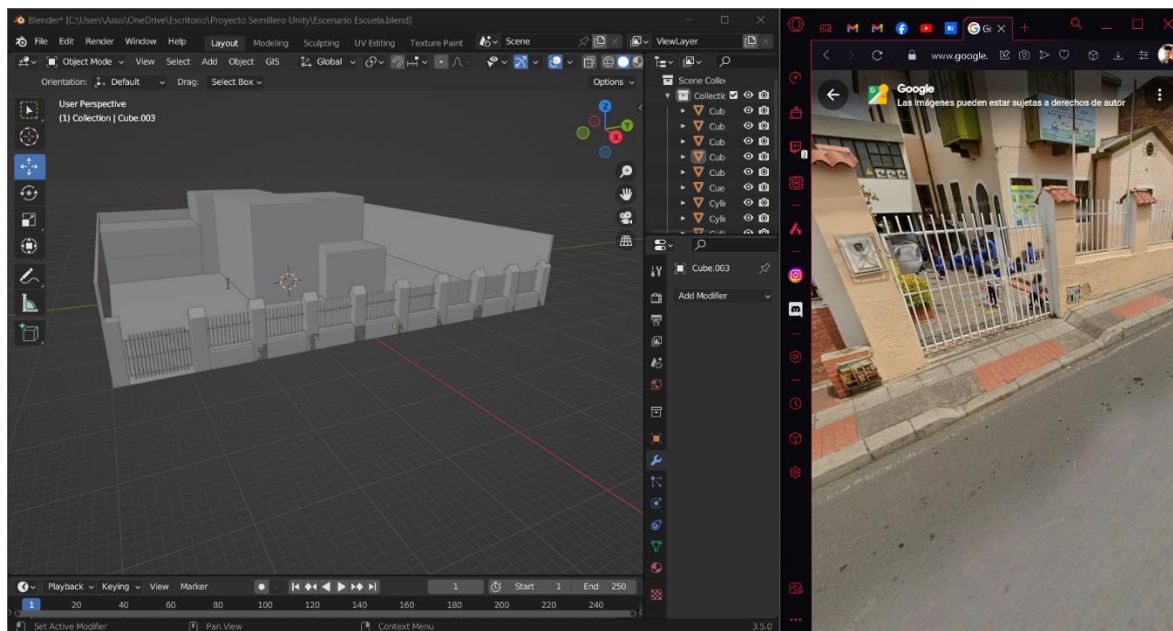
## Escenario

### *Fase Blocking*

El proceso de blocking implicó la creación de una versión básica y simplificada del entorno en la que se definen las principales formas y estructuras sin detallar (Criscuolo, 2021). Esto permite establecer la disposición general del escenario, los espacios, las dimensiones y la navegabilidad antes de pasar a la etapa de refinamiento y detalle. El blocking ayuda a los desarrolladores a visualizar y ajustar la composición del escenario de manera rápida y efectiva antes de abordar los aspectos más minuciosos.

### Figura 17

#### *Blocking escenario prototipo*



Fuente: Elaboración propia

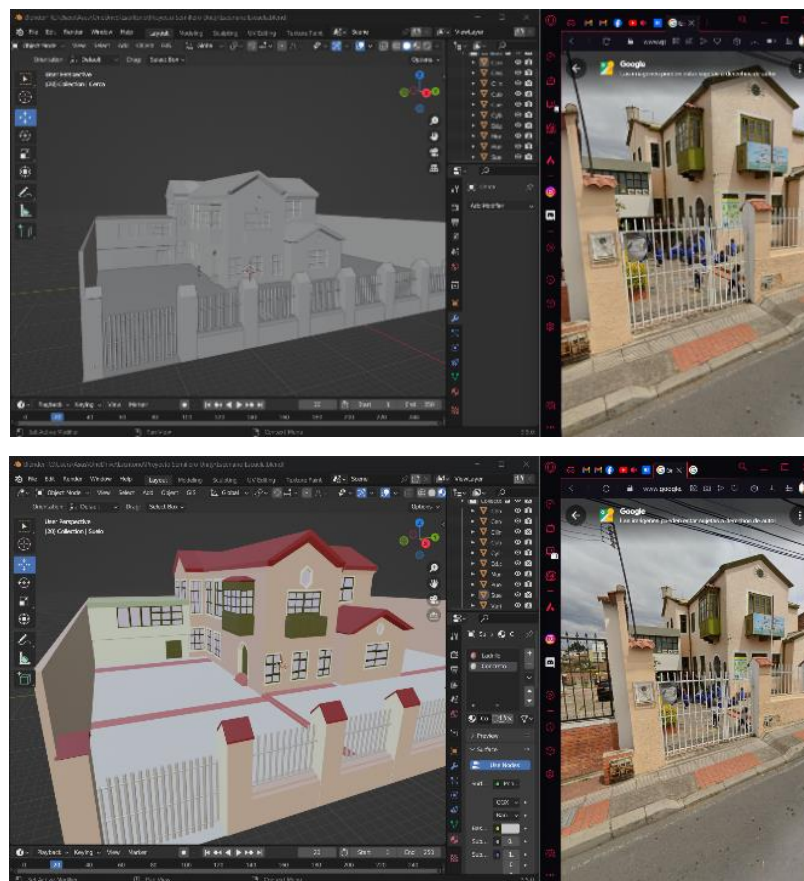
### *Fase detalle*

La fase de detallado o refinamiento implicó en este caso, al ser un prototipo la

incorporación de materiales y elementos decorativos específicos con el objetivo de enriquecer la calidad visual y estética del escenario (Animum & Solano, 2023). Durante esta fase, se prestó especial atención a cada detalle de la referencia, buscando crear una experiencia inmersiva y realista para el usuario ya que el escenario es un edificio del colegio Gimnasio San Diego, al que el usuario asiste. Como se muestra a continuación:

**Figura 18**

*Detallado y texturizado escenario prototipo*



Fuente: Elaboración propia

**Heads up display (HUDS)**

El HUD (Heads Up Display) en los videojuegos educativos es una interfaz gráfica superpuesta en la pantalla del jugador para proporcionar información relevante e instrucciones

interactivas durante el juego (Hernandez, 2019). Este elemento visual incluye indicadores de progreso, instrucciones, marcadores de objetivos, evaluaciones y otros elementos que ayudan a los usuarios a comprender y navegar en el entorno de aprendizaje virtual. El diseño de HUD en videojuegos educativos tiene como objetivo ayudar a la comprensión del usuario al priorizar la claridad, la usabilidad y la coherencia visual para mejorar el aprendizaje de una manera efectiva y atractiva. Para ello se establecieron los siguientes HUDS para que el usuario tuviera las instrucciones necesarias para el uso del videojuego.

**Figura 19**

*HUDS de pantalla del prototipo*



Fuente: Elaboración propia

## Aspectos técnicos

### *Hardware y software de desarrollo*

Este videojuego formativo fue diseñado para computadoras ya que esta será la plataforma principal debido a su versatilidad y capacidad para implementar la característica específica de la mecánica de reconocimiento de voz. A continuación, se detallan las razones de la selección de esta plataforma.

Principal herramienta tecnológica: En las instalaciones de la institución educativa se cuenta con esta herramienta para que los estudiantes desarrollen actividades de tecnología y consulta. El computador permite ejecutar algoritmos complejos para el reconocimiento de voz, el procesamiento de señales de audio y la interpretación de comandos verbales.

Desarrollo de Software: gracias al Windows Speech de Unity el computador permite implementar de una manera sencilla la funcionalidad de reconocimiento de voz, sin necesidad de instalar o crear bases de datos de idiomas. Para garantizar el buen funcionamiento del reconocimiento de voz se hace uso del motor de videojuegos Unity con el lenguaje nativo C# y con el apoyo del programa Blender para la creación de elementos 3D y animación.

Experiencia del Usuario: teniendo presente que los estudiantes de la institución educativa cuentan con habilidades en el manejo de los computadores, se facilitó la comprensión de los comandos, botones y acciones requeridas para la movilidad del personaje y el cumplimiento de las tareas del nivel.

En lo que respecta al software del juego formativo se hizo uso de lenguaje de programación por cada pantalla en la que el usuario interactuó por medio de los scripts y realizó interacciones con botones, a continuación, se ilustran las pantallas y el aspecto del videojuego.

**Figura 20**

*Pantalla inicial prototipo de Videojuego*



Fuente: Elaboración propia

En la figura anterior se observa la pantalla inicial con la que se encontró el usuario al ingresar al prototipo del videojuego formativo, esta pantalla presenta dos botones interactivos; el botón para iniciar el juego y el botón de salida. Cada uno cuenta con su respectiva programación como se observa en las próximas figuras.

**Figura 21**

*Script de programación botón inicio*

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class InicioMenu : MonoBehaviour
7  {
8      public void changeEscene()
9      {
10         SceneManager.LoadScene(1);
11     }
12 }
13

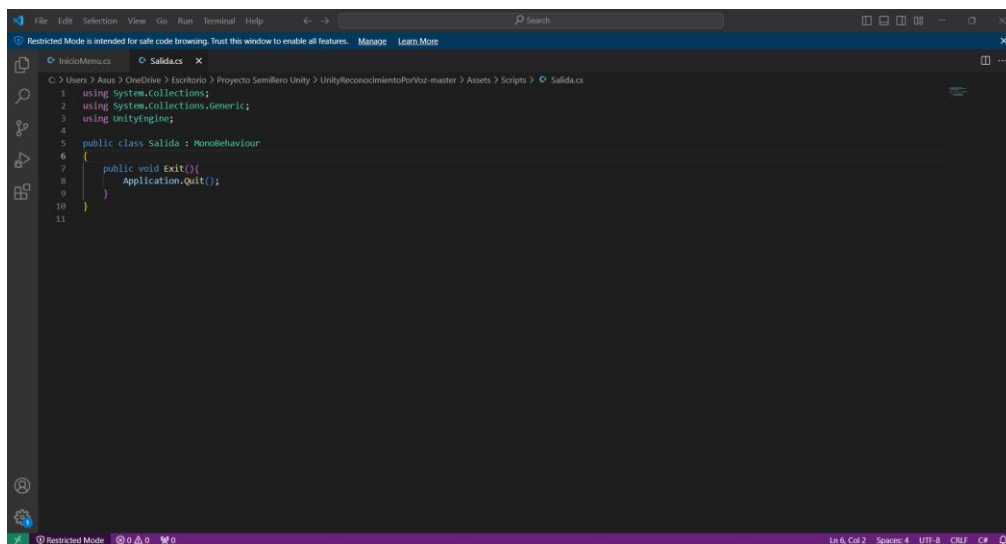
```

Fuente: Elaboración propia

En la figura anterior se hace llamado a la programación del botón “Iniciar”, en este se programa la función `changeEscene` mediante el uso de la librería propia de Unity `SceneManager` se obtuvo la línea de código `SceneManager.LoadScene(1)`; que nos permitió hacer el cambio de pantalla a la selección de personajes como se mostró anteriormente en el flujo de pantallas.

## Figura 22

*Script de programación botón salida*

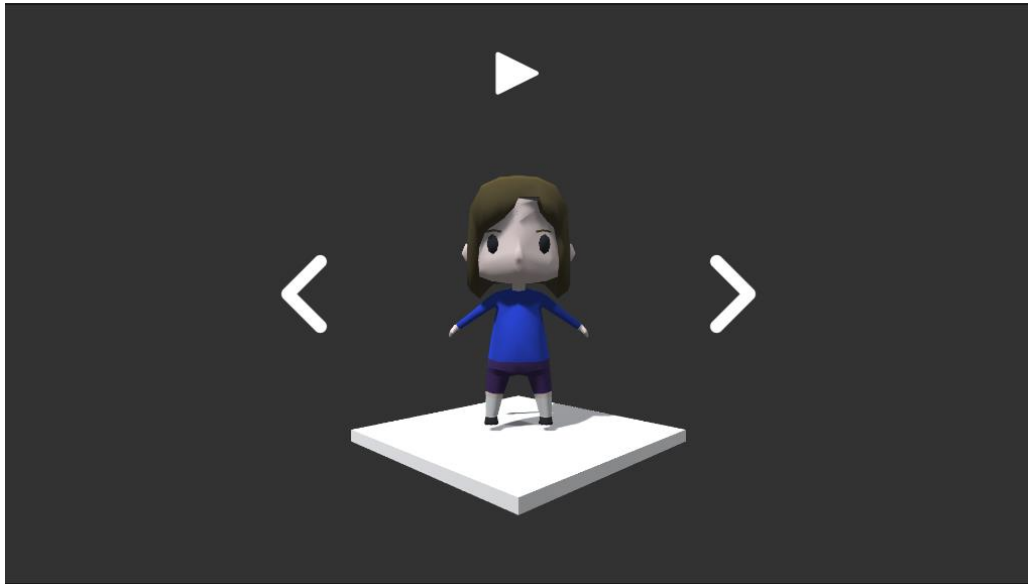


```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Salida : MonoBehaviour
6 {
7     public void Exit(){
8         Application.Quit();
9     }
10 }
11
```

Fuente: Elaboración propia

En la figura 22 se observa la programación del botón “Salir” de la pantalla inicial, en esta se programa la función `Exit` que es propia del lenguaje nativo de Unity, es decir, esta no necesita del llamado de una librería adicional de Unity para su funcionamiento y mediante la línea de código de `Application.Quit` el usuario pudo cerrar el prototipo de videojuego formativo y salir al sistema operativo (escritorio del computador).

Cuando el usuario ingresó al prototipo de videojuego, seleccionó el botón iniciar, se desplegó la pantalla para escoger el personaje de su preferencia, como se muestra en la siguiente ilustración.

**Figura 23***Pantalla Selección de Personaje*

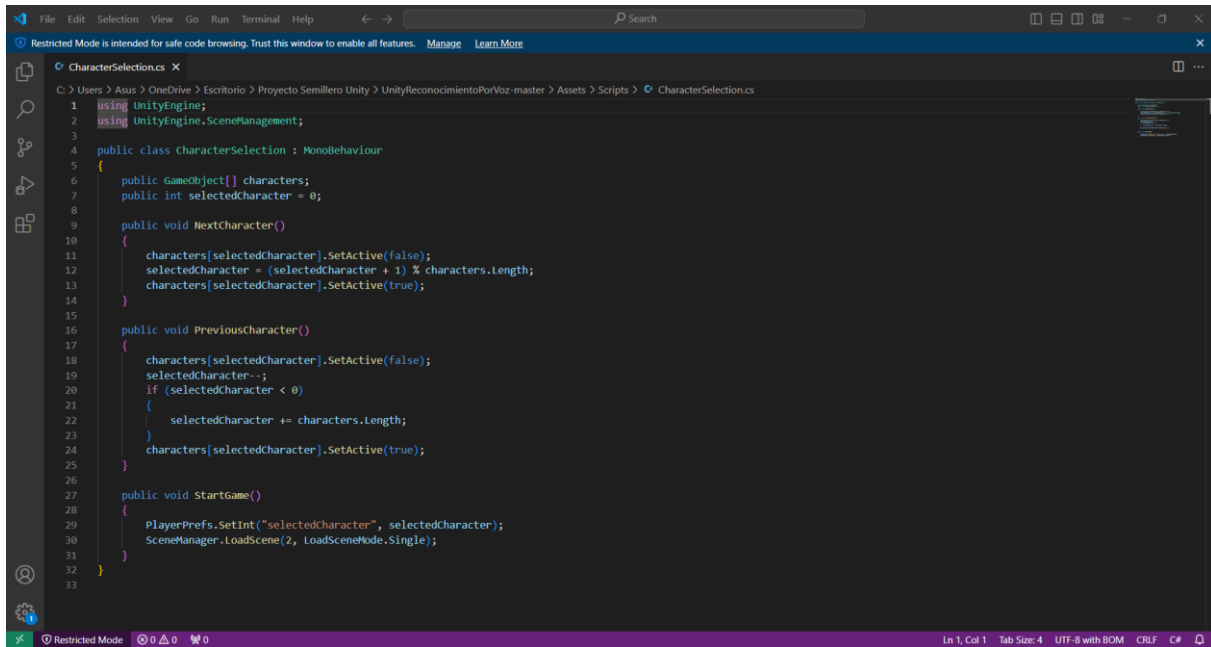
Fuente: Elaboración propia

En la figura anterior se observa la pantalla de selección de personajes en donde el usuario encontraba tres botones de interacción, los cuales eran: en la parte superior el botón start (comenzar) y a los costados los botones de siguiente (>) y anterior (<) para la selección del personaje de la preferencia del usuario. Para esta pantalla se hizo uso del siguiente script de programación.



**Figura 24**

*Script programación pantalla de selección de personajes*



```
1 using UnityEngine;
2 using UnityEngine.SceneManagement;
3
4 public class CharacterSelection : MonoBehaviour
5 {
6     public GameObject[] characters;
7     public int selectedCharacter = 0;
8
9
10    public void NextCharacter()
11    {
12        characters[selectedCharacter].SetActive(false);
13        selectedCharacter = (selectedCharacter + 1) % characters.Length;
14        characters[selectedCharacter].SetActive(true);
15    }
16
17    public void PreviousCharacter()
18    {
19        characters[selectedCharacter].SetActive(false);
20        selectedCharacter--;
21        if (selectedCharacter < 0)
22        {
23            selectedCharacter += characters.Length;
24        }
25        characters[selectedCharacter].SetActive(true);
26    }
27
28    public void StartGame()
29    {
30        PlayerPrefs.SetInt("selectedCharacter", selectedCharacter);
31        SceneManager.LoadScene(2, LoadSceneMode.Single);
32    }
33 }
```

Fuente: Elaboración propia

En la figura 24 se encuentra la construcción del script en donde está la funcionalidad de cada botón y la creación de un array (listado) de personajes con 3 posibilidades, en donde cada uno escogió un personaje de los que previamente se habían mencionado en el documento (1. Niño, 2. Niña y 3. Personaje inicial – mago) y los botones anterior y siguiente para desplazarse en la lista de personajes disponibles. Por último, el botón start (comenzar) que seleccionaba el personaje en pantalla y así empezaba el nivel principal.

En la siguiente figura se muestra la pantalla del nivel principal, siendo esta en la que el usuario podía explorar el escenario y cumplir con los puzzles del juego formativo.

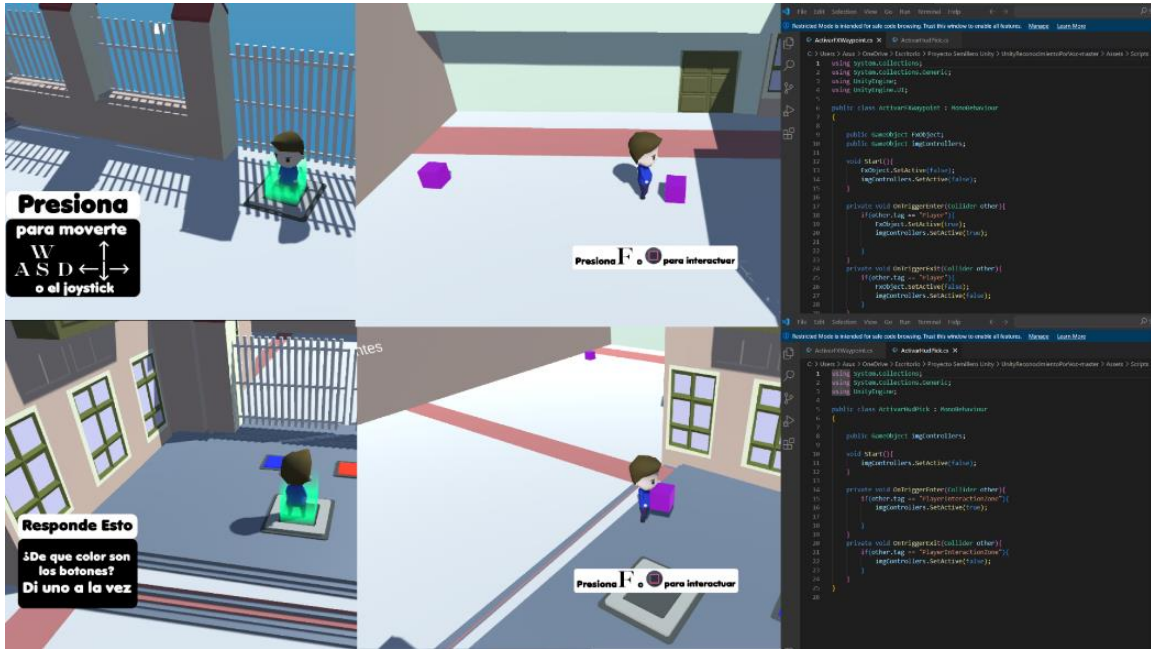
**Figura 25***Pantalla Nivel Principal*

Fuente: Elaboración propia

En la figura anterior se observan las opciones del nivel principal con mensajes e instrucciones que se activaban mediante la exploración e interacción del usuario con el entorno. Cada uno contó con la programación respectiva para permitir que el usuario pudiera hacer uso del juego formativo.

**Figura 26**

*Script de activación de mensajes de ayuda*



Fuente: Elaboración propia

En la figura anterior se observa la programación que dio inicio y activación a los distintos mensajes de pantalla que llevaban las instrucciones para el usuario, para ello se programó una serie de colisiones que cuando entraban en interacción activa mostraban el mensaje y cuando se perdía la interacción se desactivan.

Figura 27

Script programación movimiento del personaje pantalla principal

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PlayerController : MonoBehaviour
6 {
7
8     public float horizontalMove;
9     public float verticalMove;
10    private Vector3 playerInput;
11    public CharacterController player;
12    public float playerSpeed;
13    private Vector3 movePlayer;
14
15    public Animator anim;
16
17    public float gravity = 9.8f;
18    public float fallVelocity;
19    public float jumpForce;
20
21    public bool isOnSlope = false;
22    private Vector3 hitNormal;
23    public float slideVelocity;
24    public float slopeForceDown;
25
26    public Camera mainCamera;
27    private Vector3 camForward;
28    private Vector3 camRight;
29
30    // Start is called before the first frame update
31    void Start()
32    {
33        player = GetComponent<CharacterController>();
34    }
35
36
37
38    // Update is called once per frame
39    void Update()
40    {
41
42        horizontalMove = Input.GetAxis("Horizontal");
43        verticalMove = Input.GetAxis("Vertical");
44
45        playerInput = new Vector3(horizontalMove, fallVelocity, verticalMove);
46        playerInput = Vector3.ClampMagnitude(playerInput, 1);
47        camDirection();
48        movePlayer = playerInput.x * camRight + playerInput.z * camForward;
49        movePlayer = movePlayer * playerSpeed;
50        player.transform.LookAt(player.transform.position + movePlayer);
51        SetGravity();
52        PlayerSkills();
53        transform.Rotate(0, horizontalMove * Time.deltaTime * playerSpeed, 0);
54        transform.Translate(0, 0, verticalMove * Time.deltaTime * playerSpeed);
55        anim.SetFloat("VelX", horizontalMove);
56        anim.SetFloat("VelY", verticalMove);
57        player.Move(movePlayer * Time.deltaTime);
58
59    void PlayerSkills()
60    {
61        if (player.isGrounded && Input.GetButtonDown("Jump"))
62        {
63            fallVelocity = jumpForce;
64            movePlayer.y = fallVelocity;
65        }
66    }
67
68    void SetGravity()
69    {
70        if (player.isGrounded)
71        {
72            fallVelocity = -gravity * Time.deltaTime;
73            movePlayer.y = fallVelocity;
74        }
75        else
76        {
77            fallVelocity = gravity * Time.deltaTime;
78            movePlayer.y = fallVelocity;
79        }
80    }
81
82    void SetGravity()
83    {
84        SlideDown();
85    }
86
87    void camDirection()
88    {
89        camForward = mainCamera.transform.forward;
90        camRight = mainCamera.transform.right;
91        camForward.y = 0;
92        camRight.y = 0;
93        camForward = camForward.normalized;
94        camRight = camRight.normalized;
95    }
96
97    public void SlideDown()
98    {
99        isOnSlope = Vector3.Angle(Vector3.up, hitNormal) >= player.slopeLimit;
100        if (isOnSlope)
101        {
102            movePlayer.x += ((if - hitNormal.y) * hitNormal.x) * slideVelocity;
103            movePlayer.z += ((if - hitNormal.y) * hitNormal.z) * slideVelocity;
104            movePlayer.y += slopeForceDown;
105        }
106    }
107
108    void onControllerColliderHit(ControllerColliderHit hit)
109    {
110        hitNormal = hit.normal;
111    }
112

```

Fuente: Elaboración propia

En la figura 27 se muestra la programación para el movimiento del personaje del prototipo de videojuego formativo, en este script, se definió el comportamiento del jugador en términos de movimiento, gravedad, habilidades y detección de colisiones con pendientes. A continuación, se describen las funciones principales del código:

- Se declararon variables para el movimiento horizontal y vertical, la velocidad del jugador, la gravedad, la fuerza de salto, entre otros parámetros.
- En el método Update(), se capturaron las entradas del jugador para controlar el movimiento en el plano x-z relativo a la cámara, la rotación del jugador, la aplicación de gravedad y la ejecución de habilidades del jugador; como el salto.
- El método PlayerSkills() gestionó el salto del jugador cuando está en el suelo.
- El método SetGravity() aplicó la gravedad al jugador, ajustando la velocidad de caída en función de si está en el suelo o en el aire.
- La función camDirection() calculó la dirección hacia adelante y hacia la derecha relativa a la cámara.

- SlideDown() detectó si el jugador está en una pendiente y aplicó fuerzas para simular el deslizamiento hacia abajo.
- Por último, la función onControllerColliderHit() registró la normal de la colisión con una superficie, útil para el cálculo del deslizamiento en pendientes.

### Figura 28

#### *Pantalla de pausa*

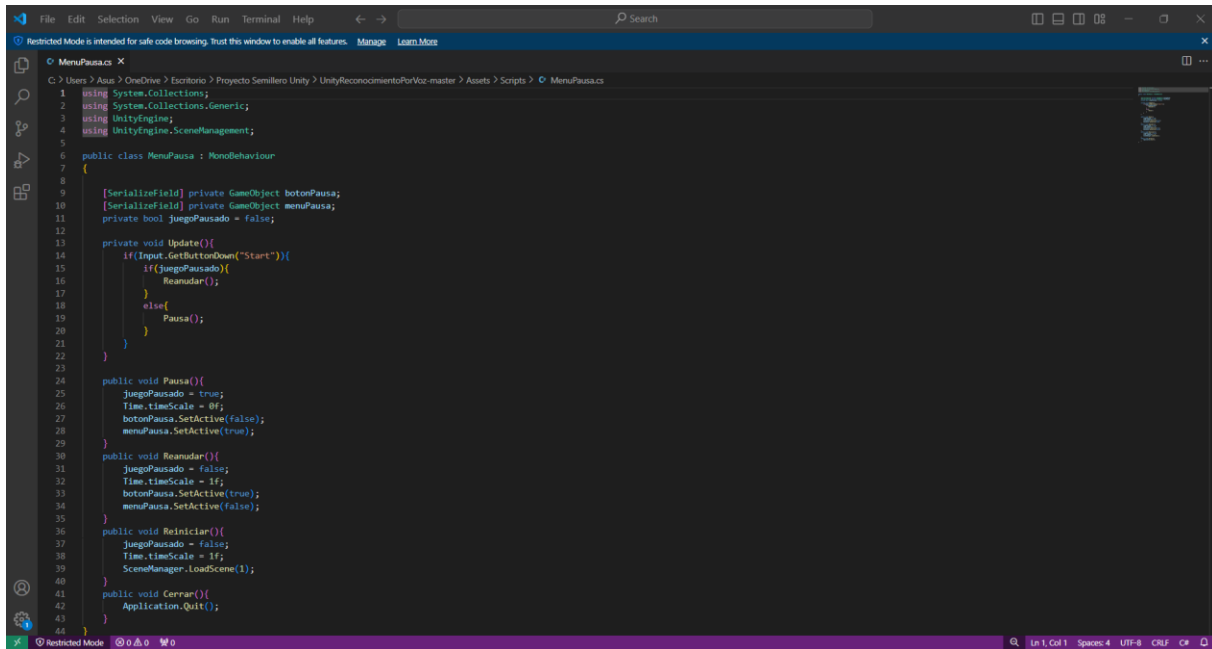


Fuente: Elaboración propia

En la figura 28 se observa la pantalla de pausa la cual brindó al usuario tres opciones de selección o interacción estas son los botones reiniciar, salir y volver, la lógica usada para el botón de reiniciar es la misma presentada anteriormente con el uso de la librería adicional SceneManagement, el cual permitió iniciar desde cero el nivel actual. Por otra parte, el botón de salir utilizó la lógica previamente vista en la figura 22 que hace uso de la línea Application.Quit para ir al escritorio del computador y para el botón volver su programación se ve expresada en la siguiente figura y permite reanudar el nivel en el punto en que se seleccionó pausa.

**Figura 29**

*Script de programación pantalla de pausa*



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class MenuPausa : MonoBehaviour
7 {
8
9     [SerializeField] private GameObject botonPausa;
10    [SerializeField] private GameObject menuPausa;
11    private bool juegoPausado = false;
12
13    private void Update()
14    {
15        if (Input.GetButtonDown("Start"))
16        {
17            if (juegoPausado)
18            {
19                Reanudar();
20            }
21            else
22            {
23                Pausa();
24            }
25        }
26    }
27
28    public void Pausa()
29    {
30        juegoPausado = true;
31        Time.timeScale = 0f;
32        botonPausa.SetActive(false);
33        menuPausa.SetActive(true);
34    }
35
36    public void Reanudar()
37    {
38        juegoPausado = false;
39        Time.timeScale = 1f;
40        botonPausa.SetActive(true);
41        menuPausa.SetActive(false);
42    }
43
44    public void Reiniciar()
45    {
46        juegoPausado = false;
47        Time.timeScale = 1f;
48        SceneManager.LoadScene(1);
49    }
50
51    public void Cerrar()
52    {
53        Application.Quit();
54    }
55 }
```

Fuente: Elaboración propia

En esta figura 29 se observa la programación del botón volver, este hace uso de la línea de la función propia de Unity `Time.timeScale` que hace referencia a la activación de pausa y reanudación de la escena en la que es llamada, en este caso su valor es 0 y pausa la partida y cuando su valor es 1 reanuda la partida justo donde el usuario la pausa.

Por último, se presenta la pantalla final del juego, esta aparece al concluir el juego formativo.

**Figura 30***Pantalla final*

Fuente: Elaboración propia

En la figura 30 se observa la pantalla final en esta se le da un mensaje de agradecimiento al usuario por probar el prototipo de videojuego, tiene dos opciones de interacción el botón salir que tiene la misma lógica de los botones salir que se pueden encontrar en otras pantallas y el botón volver a empezar que carga nuevamente todo el prototipo de videojuego formativo para que el usuario pueda volver a probar y a empezar la prueba de este maravilloso prototipo de videojuego formativo y la programación de este hace uso de la librería adicional de Unity SceneManager que ya previamente se ha explicado su funcionamiento.

### **Evaluar la funcionalidad del prototipo de videojuego formativo en la correcta pronunciación con los niños de grado primero**

En la evaluación de la funcionalidad del prototipo del videojuego formativo en lo que respecta a la correcta pronunciación de los niños de primer grado se podrá observar la evidencia de la participación de los usuarios en el **Anexo D. Galería de Fotografías**, por ello se buscó determinar si el prototipo logró obtener una correcta funcionalidad sobre el reconocimiento de voz, así como identificar posibles obstáculos o áreas de mejora en relación con el reconocimiento de voz y algunas opciones en la distribución del HUD.

Esta evaluación incluyó pruebas prácticas con los niños, observación de su interacción con el juego, y la recopilación de retroalimentación de los niños. Los resultados de esta evaluación permitieron ajustar y mejorar el prototipo para que cumpliera de manera efectiva con su propósito formativo en relación con la correcta pronunciación de los niños de primer grado del gimnasio San Diego de Tunja.

Según lo establecido en el artículo 11 de la Resolución 8430 De 1993 (Ministerio de Salud, 1993) por la cual se establecen las normas científicas, técnicas y administrativas para la investigación en salud.

La presente investigación hace parte una investigación con riesgo mínimo por tratarse de un estudio prospectivo en el que se emplea el registro de datos a través de procedimientos comunes para el caso pruebas a grupos e individuos en los que no se manipulará la conducta del sujeto.

En respuesta a la misma resolución, el rechazo de los sujetos a intervenir o retiro de su consentimiento durante el estudio, no afectará su situación escolar y los resultados de la investigación no serán utilizados en perjuicio de los participantes.

Como respuesta a resolución y siguiendo la normativa se adjuntos los formatos de consentimiento informado en el **Anexo F. Formato Consentimiento** y los consentimientos diligenciados en el **Anexo C. Consentimientos Completados**.

En esta investigación se aplica una metodología mixta, es decir, los datos se recopilan utilizando métodos tanto numéricos como descriptivos. Por ejemplo, esto podría incluir la realización de encuestas para recopilar, así como la realización de entrevistas en profundidad para explorar cualitativamente las experiencias y perspectivas de los usuarios (Ortega, 2021).

Los resultados de la prueba final del prototipo se muestran a continuación, en las siguientes



tablas de datos y gráficos de barras respectivas.

**Grado de dificultad en el uso de los controles del videojuego**

Uno de los principales aspectos que se evaluó fue el uso de los controles del videojuego, recordando que se hizo uso de un mando externo, teclado y mouse.

**Tabla 1**

*Grado de dificultad en el uso de los controles del videojuego*

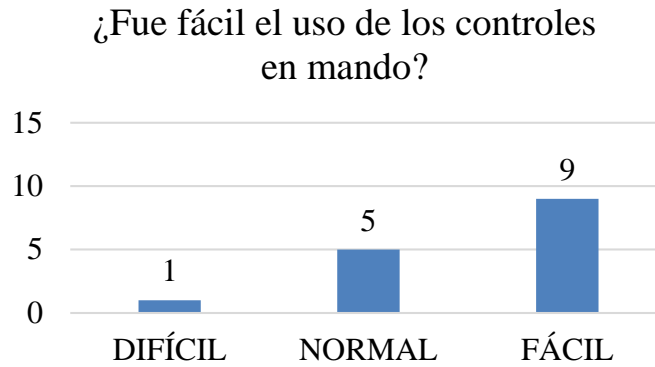
<b>Grado de dificultad en el uso de los controles del videojuego</b>			
<b>VARIABLES</b>	<b>DIFÍCIL</b>	<b>NORMAL</b>	<b>FÁCIL</b>
¿Fue fácil el uso de los controles en mando?	1	5	9
<b>VARIABLES</b>	<b>DIFÍCIL</b>	<b>NORMAL</b>	<b>FÁCIL</b>
¿Para ti fue fácil ubicarte en los controles?	6	4	5
<b>VARIABLES</b>	<b>DIFÍCIL</b>	<b>NORMAL</b>	<b>FÁCIL</b>
¿Para ti el uso de los controles del ordenador fue fácil?	2	5	8

Fuente: Encuesta elaborada en el Anexo B. a niños de grado primero

En la tabla anterior se puede observar los resultados obtenidos respecto al Grado de dificultad en el uso de los controles del videojuego y cómo fue desglosada la variable de evaluación de Controles de uso, definida en el anexo A, para ello se formularon tres preguntas para poder obtener más datos hacia la variable principal, obteniendo las siguientes gráficas con los respectivos datos:

**Figura 31**

*Gráfica de barras uso de mando*

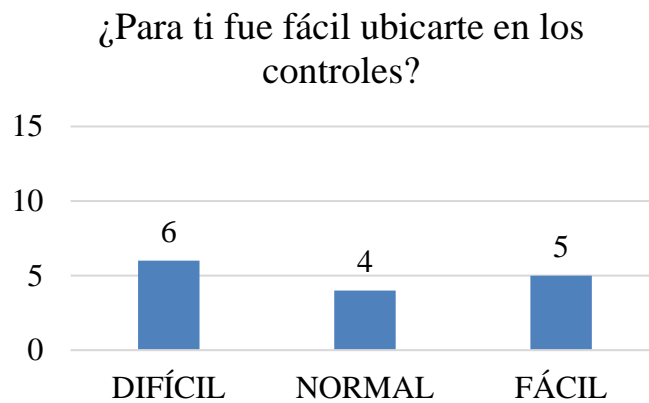


Fuente: Resultados de las encuestas del anexo B. aplicadas a niños de grado primero

En la gráfica anterior se obtuvieron datos acerca de la utilización de periféricos extras para el funcionamiento del prototipo, ya que los diferentes usuarios pudieron usar un mando externo para interactuar de manera más cómoda con el prototipo, el 60% de los usuarios manifestaron que fue fácil el uso de los controles en mando, mientras que el 33% respondieron un uso normal de este periférico y solo el 7% es decir, un usuario manifestó dificultad en el uso del mando.

**Figura 32**

*Gráfica de barras ubicación controles*



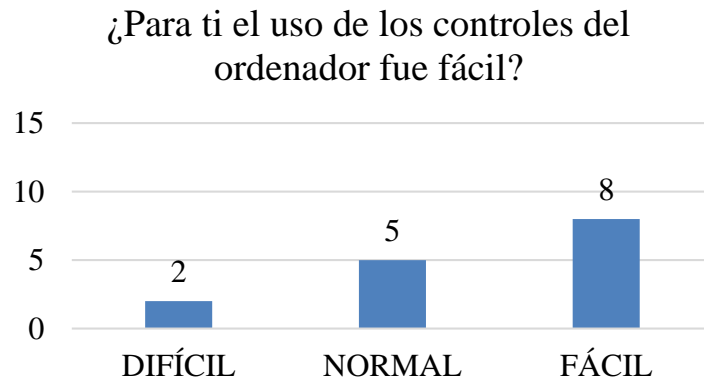
Fuente: Resultados de las encuestas del anexo B. aplicadas a niños de grado primero

En la gráfica anterior se puede observar que el usuario debe de manera intuitiva ubicarse

en los controles, ya sea en ordenador como en el mando externo, es decir, los usuarios sin necesidad de recibir la instrucción externa del investigador pueden ubicarse y utilizar los controles, por ende, se obtiene un resultado del 40% denotan una alta dificultad en la ubicación de los controles, es decir, necesitaron de una indicación para ubicarse en ellos, para 27% la ubicación de los controles fue normal pero necesitaron apoyo para reubicarse en ellos, en cambio el 33% de los usuarios mostraron una facilidad en la ubicación de los controles debido a que tienen conocimientos previos en el uso de videojuegos.

**Figura 33**

*Gráfica de barras uso de controles – ordenador*



Fuente: Resultados de las encuestas del anexo B. aplicadas a niños de grado primero

En la gráfica anterior se obtienen datos respecto a la utilización de periféricos del ordenador para el funcionamiento del prototipo, ya que los diferentes usuarios pueden usar teclado y mouse para interactuar con el prototipo, por ende el 53% tuvo como resultado facilidad en el uso de los controles utilizando el teclado y mouse del computador, para el 33% manifestó uso normal del teclado y mouse, mientras que el 14% es decir, 2 usuarios demostraron dificultad en el uso del hardware del ordenador.

**Grado de aceptación del ambiente**

Otro aspecto que se evaluó fue la aceptación del ambiente comprendido por el escenario, los personajes y los colores del videojuego.

**Tabla 2**

*Grado de aceptación del ambiente*

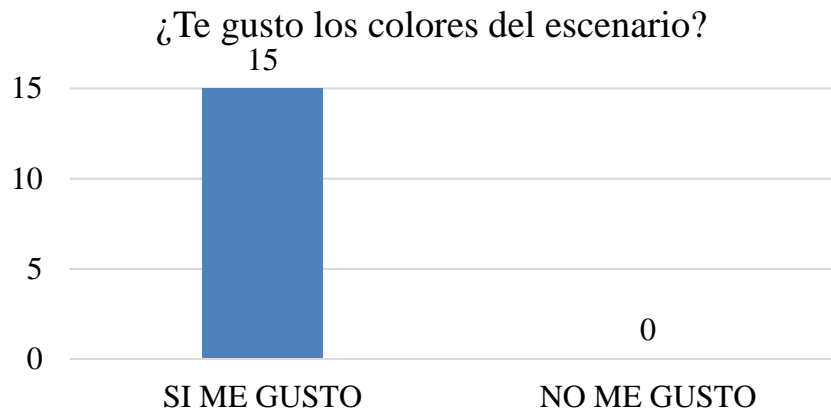
<b>Grado de aceptación del ambiente</b>		
<b>VARIABLES</b>	<b>SI ME GUSTO</b>	<b>NO ME GUSTO</b>
¿Te gusto los colores del escenario?	15	0
<b>VARIABLES</b>	<b>SI ME GUSTO</b>	<b>NO ME GUSTO</b>
¿Te gusto el personaje del videojuego?	14	1
<b>VARIABLES</b>	<b>SI ME GUSTO</b>	<b>NO ME GUSTO</b>
¿Te gusto el escenario en donde se desarrolla el videojuego?	15	0

Fuente: Encuesta elaborada en el Anexo B. a niños de grado primero

La tabla muestra los resultados respecto a la variable de aceptación del ambiente por parte de los usuarios, la variable está definida en el anexo A., haciendo referencia a si el diseño del escenario fue del gusto del usuario y si los diferentes parámetros de diseño de personajes tuvieron aprobación por los usuarios, las preguntas designadas para la obtención de datos fueron menos técnicas debido a la naturaleza del usuario final y se centraron en lo subjetivo, es decir en el gusto o no del ambiente, personajes y escenarios propuestos en el videojuego.

**Figura 34**

*Gráfica de barras colorimetría del escenario*

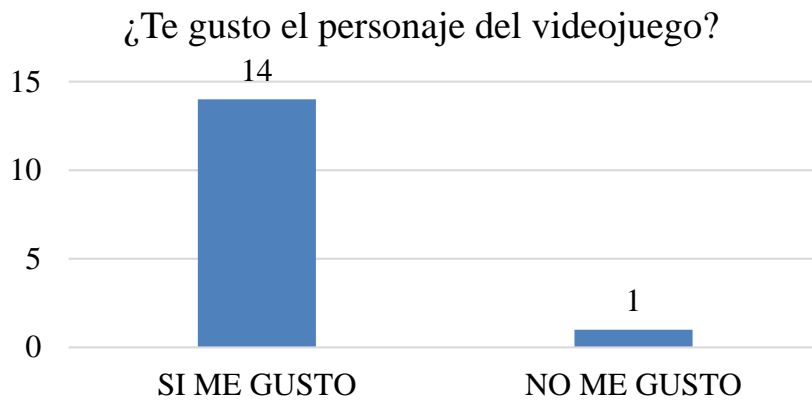


Fuente: Resultados de las encuestas del anexo B. aplicadas a niños de grado primero

La gráfica muestra las respuestas referentes a la colorimetría del escenario y cómo este fue visto y aprobado por la gran mayoría de los usuarios. Dando como resultado un 100% de grado de aceptación de los colores del escenario.

**Figura 35**

*Gráfica de barras aceptación del personaje*

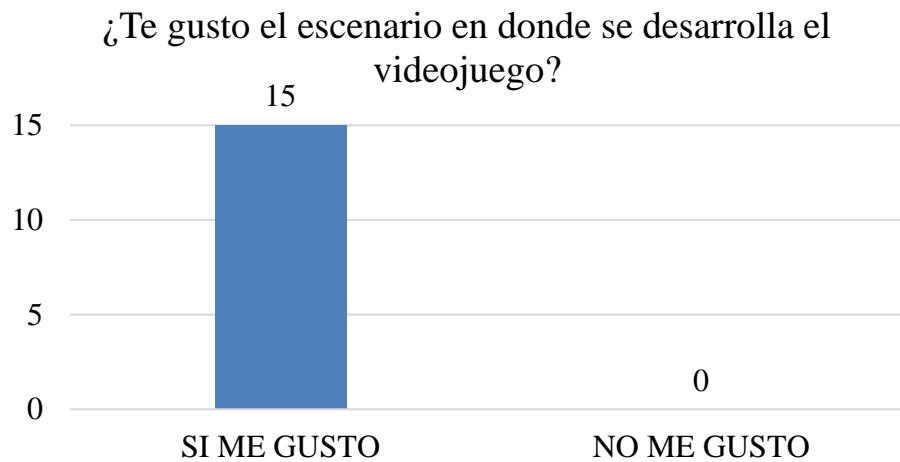


Fuente: Resultados de las encuestas del anexo B. aplicadas a niños de grado primero

La gráfica representa los datos referentes a la integración de los diferentes personajes en el prototipo de videojuego y a su apariencia, la cual es evaluada por los distintos usuarios. En donde al 93% de los usuarios si les gustaron los diferentes personajes presentados en el videojuego y tan solo al 7% es decir, un usuario manifestó desagrado hacia el personaje del mago debido a su estilo.

**Figura 36**

*Gráfica de barras aceptación del escenario*



Fuente: Resultados de las encuestas del anexo B. aplicadas a niños de grado primero

La gráfica muestra los resultados de la valoración de la edificación del escenario, pues por las respuestas obtenidas en la prueba piloto hubo necesidad de desarrollar un escenario que fuera conocido por el jugador con el que estuviera familiarizado, se evalúa entonces si el escenario gusta o no, los resultados mostraron que el 100% de los usuarios manifestaron gusto por el escenario, aparte de reconocer el entorno debido al parentesco con el entorno escolar.

**Grado de comunicación de la interfaz**

**Tabla 3**

*Grado de comunicación de la interfaz*

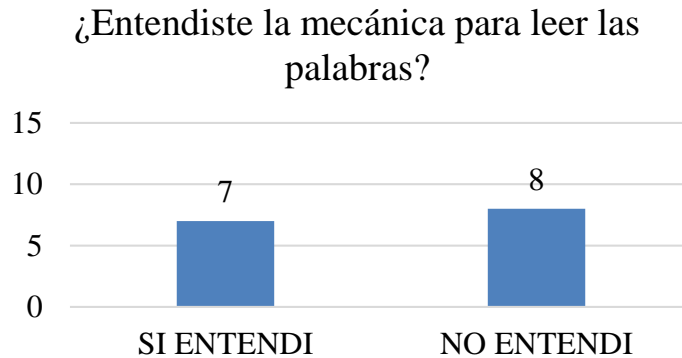
<b>Variables</b>	<b>SI ENTENDI</b>	<b>NO ENTENDI</b>
<b>¿Entendiste la mecánica para leer las palabras?</b>	7	8
<b>Variables</b>	<b>SI ENTENDI</b>	<b>NO ENTENDI</b>
<b>¿Entendiste las indicaciones dadas en la pantalla?</b>	6	9
<b>Variables</b>	<b>SI ENTENDI</b>	<b>NO ENTENDI</b>
<b>¿Entendiste la activación de la mecánica de lectura de palabras?</b>	6	9

Fuente: Encuesta elaborada en el Anexo B. a niños de grado primero

En la tabla anterior se puede observar los resultados obtenidos respecto al Grado de comunicación de interfaz o HUD, es decir aquellos elementos en pantalla interactivables tales como botones o mensajes de interacción o de instrucciones para el usuario, dicho aspecto se encuentra definido en el anexo A, obteniendo las siguientes gráficas con los respectivos datos:

**Figura 37**

*Gráfica de barras mecánica de lectura de palabras*

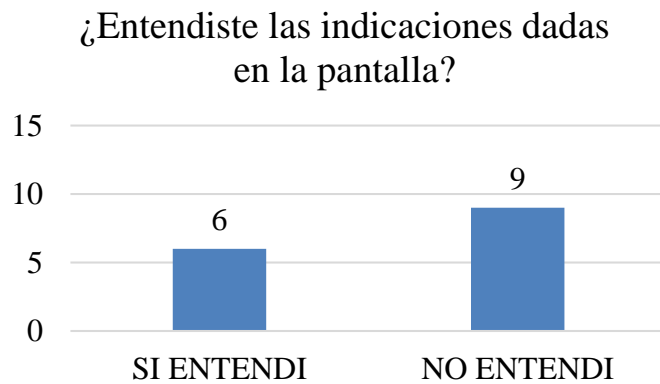


Fuente: Resultados de las encuestas del anexo B. aplicadas a niños de grado primero

La gráfica muestra los resultados respecto a la correcta interacción del usuario con el mensaje puesto en la pantalla del nivel principal respecto a la instrucción para que el usuario pronuncie de manera correcta las palabras, el 53% no entendieron las instrucciones al leer el mensaje que mostraba la pantalla, mientras que el 47% si entendió correctamente la instrucción dada en el mensaje.

**Figura 38**

*Gráfica de barras entendimiento instrucciones de pantalla*



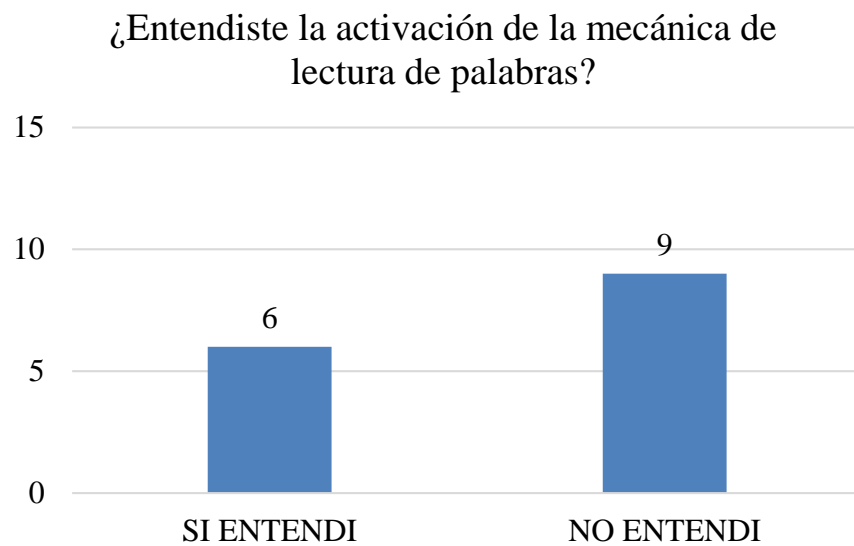
Fuente: Resultados de las encuestas del anexo B. aplicadas a niños de grado primero



La gráfica muestra los resultados respecto a la comprensión de los mensajes e instrucciones dadas en todas las pantallas del prototipo, el 60% de los usuarios no entendían las instrucciones y solicitaban ayuda del investigador para realizar la instrucción solicitada por el mensaje del prototipo, en cambio el 40% de los usuarios observaban la instrucción en pantalla y la realizaban con facilidad.

**Figura 39**

*Gráfica de barras entendimiento activación mecánica de voz*



Fuente: Resultados de las encuestas del anexo B. aplicadas a niños de grado primero

La gráfica corresponde a los resultados respecto a la comprensión de la activación del cambio del color del objeto con la lectura de la palabra, es decir, se necesitaba de una pronunciación fuerte para la activación de la mecánica, en la gráfica anterior se muestra que el 60% de los usuarios al no entender la instrucción no comprendieron que se necesitaba pronunciar con tono fuerte las palabras solicitadas mientras que el 40% si realizo la correcta pronunciación de las palabras solicitadas.

## Conclusiones

Al analizar la mecánica de reconocimiento de voz del prototipo de videojuego formativo, se puede concluir que su integración con el motor de videojuegos UNITY es óptima, ya que, ésta reconoce cuando hay una correcta pronunciación de la palabra, pero se necesita de mayor tiempo de desarrollo para poder establecer la identificación de una mala pronunciación y de párrafos completos.

Se aplicó la metodología ágil Kanban durante la fase de diseño del prototipo de videojuego formativo, lo que facilitó la creación de una prueba piloto en donde el usuario final estableció las pautas de diseño y desarrollo del prototipo obteniendo como resultado: cambio del ambiente, personaje a un estudiante, movimiento libre de cámara y cambio de aspectos de HUDS.

Para la ejecución de este proyecto de investigación se tuvo en cuenta las diferentes fases y tareas propuestas para la construcción del prototipo llevando así la fase artística en la creación de escenarios y personajes al punto de una personalización hacia el usuario final, mediante la puesta en práctica de los conocimientos propios de la carrera, ya que por medio de la utilización del motor de videojuegos UNITY, se creó un prototipo de videojuego formativo utilizando el lenguaje de programación nativo (C#) y el programa de modelado y animación BLENDER.

Al evaluar la funcionalidad del prototipo de videojuego formativo en la correcta pronunciación de palabras en los niños de primer grado, se puede concluir que para los diferentes usuarios manifiestan el deseo del desarrollo del prototipo de videojuego para celulares, también se pudo determinar que este prototipo a manera técnica en cuanto a usabilidad y accesibilidad tiene mayor porcentaje de aceptación por parte de los usuarios, en el ámbito de jugabilidad se obtuvo un balance en la respuesta ya que algunos usuarios si entendieron las mecánicas del prototipo y otros mostraron la necesidad de un apoyo para entenderla.

El desarrollo del prototipo de videojuego formativo, muestra diferentes complicaciones en su aplicación ya que al ser una fase inicial se obtuvo un correcto diseño y programación de las mecánicas básicas y será la base inicial para el reconocimiento de voz en el prototipo, ya que este al buscar convertirse en una herramienta de medición y aprendizaje de la fluidez y pronunciación necesita de más tiempo de diseño y construcción para lograr un correcto desarrollo para ser la herramienta dicha anteriormente, a su vez esta conclusión es compartida con una recomendación expuesta en el siguiente capítulo.

### **Recomendaciones**

Como recomendación general se busca la continuación de la investigación de este proyecto pues se denota motivación por parte de los usuarios ya que esperan obtener la versión final del videojuego formativo y a su vez la institución busca ser la primera en obtener una herramienta especializada en la medición de la fluidez en la lectura de los estudiantes.

Se propone que se continúe el desarrollo del videojuego junto a otras facultades tales como ciencias y humanidades (Programa de pedagogía infantil) para que se obtenga métodos de verificación y métodos de medida para comprobar la efectividad del videojuego formativo en la educación de los estudiantes.

Este prototipo y su desarrollo busca ampliarse y aplicarse a un grado superior, segundo o tercero de básica primaria en donde los estudiantes pronuncien frases completas y conozcan una mayor cantidad de vocabulario, y tengan también un mejor manejo y conocimiento en el uso del ordenador ya que este es la herramienta principal que poseen las instituciones educativas en sus salas de tecnología e informática.

En investigaciones futuras, se buscará una forma de satisfacer las necesidades detectadas. Por ejemplo, si la institución carece de micrófonos para la utilización del videojuego, se explorarán métodos u alternativas que permitan a los usuarios probar el videojuego sin obstáculos.

## Referencias

- Animum, y Solano, L. (2023, 6 de octubre). *Proceso de creación de un escenario 3D*.  
<https://www.animum3d.com/blog/proceso-de-creacion-de-un-escenario-3d/>
- Assetstore. (s. f.). *Biblioteca virtual de Unity*. <https://assetstore.unity.com>
- Blender.org. (s. f.). *Blender 4.2 Manual Introducción*.  
[https://docs.blender.org/manual/es/dev/getting\\_started/about/introduction.html](https://docs.blender.org/manual/es/dev/getting_started/about/introduction.html)
- Criscuolo, I. (2021, 5 de febrero). *¿Qué es el blocking en animación y por qué es tan importante?*.  
<https://www.domestika.org/es/blog/6656-que-es-el-blocking-en-animacion-y-por-que-es-tan-importante>
- Hernández, R. (2019, 12 de abril). *HUD*. <https://www.geekno.com/glosario/hud>.
- Jmbeas. (2019, 22 de septiembre). *Desarrollo iterativo e incremental*.  
<https://blog.jmbeas.es/2019/09/22/desarrollo-iterativo-e-incremental/>.
- Martins, J. (2024, 19 de enero). *¿Qué es la metodología Kanban y cómo funciona?*  
<https://asana.com/es/resources/what-is-kanban>.
- Microsoft Build. (2021, 5 de mayo). *Voice input in Unity*. <https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unity/voice-input-in-unity>
- Ministerio de Salud. (1993). *Por la cual se establecen las normas científicas, técnicas y administrativas para la investigación en salud*. [Resolución 8430 de 1993].  
<https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/DIJ/RESOLUCION-8430-DE-1993.pdf>
- Miro. (s. f.). *Plantilla de flujo de pantalla y ejemplo para equipos*.  
<https://miro.com/es/plantillas/flujo-de-pantalla/>
- Ortega, C. (2021, 14 de abril). *Investigación mixta. Qué es y tipos que existen*.  
<https://www.questionpro.com/blog/es/investigacion-mixta/>