

**Pontificia Universidad Católica del Ecuador**

**Facultad De Ingeniería**

**Escuela de Sistemas**



**TEMA:**

Desarrollo de un videojuego para niños de 5-6 para la enseñanza del cuidado medioambiental

**AUTOR:**

Fausto Sebastián Borja Villacís

TRABAJO PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE  
SISTEMAS Y COMPUTACIÓN

**QUITO, JULIO – 2023**

## DEDICATORIA

Querida madre,

Hoy, al finalizar este arduo y emocionante camino hacia la culminación de mi tesis, quiero dedicarte este logro con todo mi corazón. Tu amor incondicional, apoyo y sacrificio han sido pilares fundamentales en cada paso de mi vida y, sin duda, han sido una fuente inagotable de inspiración para alcanzar mis metas.

Desde mi infancia, has sido mi guía y mi ejemplo de fortaleza. Tu dedicación y entrega en cada desafío que has enfrentado me han enseñado el valor del esfuerzo y la perseverancia. Has sido mi mayor motivación para superar obstáculos y nunca rendirme, incluso en los momentos más difíciles.

En cada paso de este camino académico, has estado a mi lado, animándome y brindándome tu aliento. Tus palabras de aliento han sido el combustible que me ha impulsado a seguir adelante cuando la carga parecía abrumadora. Tu fe en mí ha sido un faro de luz en los momentos de duda, recordándome constantemente que soy capaz de lograr cualquier cosa que me proponga.

Tu apoyo incondicional y tus sacrificios han sido el cimiento de mi éxito. A pesar de tus propias responsabilidades y preocupaciones, siempre has encontrado tiempo para escucharme, ayudarme y brindarme ese amor incondicional que solo una madre puede dar. Cada noche en vela, cada sacrificio y cada lágrima compartida han valido la pena porque hoy puedo mirar atrás y ver que todo el esfuerzo ha dado sus frutos.

Con todo mi amor y gratitud eterna,

Tu Hijo

## **AGRADECIMIENTO**

Quiero expresar mi más sincero agradecimiento a todas las personas que hicieron posible la realización de esta tesis.

A mi orientador/a, por su invaluable guía, paciencia y conocimientos compartidos a lo largo de este proceso.

Agradezco a mi familia por su amor, apoyo incondicional y comprensión en cada etapa de este desafío académico.

A mis amigos y seres queridos, por su aliento constante y palabras de aliento que me impulsaron a seguir adelante.

Agradezco a mis compañeros de estudio, por los debates enriquecedores y la colaboración durante este camino.

A todas las fuentes de investigación consultadas, por brindarme el conocimiento necesario para desarrollar mi trabajo.

Agradezco a las instituciones y organizaciones que facilitaron el acceso a la información y recursos necesarios para llevar a cabo esta investigación.

A todos los participantes y voluntarios que generosamente contribuyeron con su tiempo y conocimientos, sin los cuales este estudio no habría sido posible.

Agradezco a las personas anónimas que de alguna manera contribuyeron en este proyecto.

A mi universidad y cuerpo docente, por brindarme la oportunidad de crecer académicamente y por su constante apoyo.

A todos aquellos que, de una forma u otra, dejaron una huella en este camino y ayudaron a enriquecer mi experiencia académica.

Agradezco a la vida misma por presentarme este desafío, que me ha permitido crecer y aprender de manera inigualable.

Gracias a todos por ser parte de este viaje y por su valiosa contribución a mi desarrollo académico y personal.

## RESUMEN

El trabajo de titulación consiste en el desarrollo de un videojuego que permita la difusión de la misma en una unidad educativa que tenga como fin la introducción y enseñanza del cuidado del medio ambiente de la manera más clara y concisa posible. Este ha sido desarrollado utilizando el motor de videojuegos Godot que usa el lenguaje de programación nativo GDScript basado en Python y Javascript. Se ha enfocado en ser accesible de tal forma que el uso de periféricos esenciales que gobiernan un computador puedan ser utilizados sin dar cabida a errores o confusiones. El tema se trata en la clasificación de objetos que son o no basura, en caso de ser basura estos mismos deben ser clasificados. Hace uso de audio e ilustraciones que fueron recolectadas por la Web, éstos que son de uso gratuito o de licenciamiento libre. Los autores los cuales son dueños de sus elementos artísticos fueron reconocidos en el apartado de créditos correspondiente al videojuego. El videojuego tiene accesibilidad en el sentido de cómo se debe operar mediante los periféricos del computador, éstos son visualizados mediante vídeos de tipo tutorial.

## ÍNDICE GENERAL

Portada.....	i
DEDICATORIA.....	ii
AGRADECIMIENTO.....	iii
RESUMEN .....	iv
ÍNDICE GENERAL .....	v
ÍNDICE DE FIGURAS .....	ix
ÍNDICE TABLAS .....	x
ÍNDICE DE ANEXOS .....	xii
CAPÍTULO I .....	1
INTRODUCCIÓN .....	1
1. MARCO DE REFERENCIA .....	1
1.1. JUSTIFICACIÓN .....	1
1.2. Planteamiento del problema .....	1
1.3. Objetivo General .....	2
1.4. Objetivos Específicos.....	2
1.5. Alcance .....	2
CAPÍTULO II .....	4
FUNDAMENTACIÓN TEÓRICA.....	4
2. Marco Teórico .....	4
2.1. Medio ambiente.....	4
2.1.1. El cuidado del medio ambiente – Laudato Si.....	4
2.1.2. Buenas prácticas del cuidado del medio ambiente.....	5
2.1.3. Videojuegos .....	7
2.1.4. Videojuegos y cuidado medioambiental.....	11
2.2. Videojuegos y tecnología.....	12
2.2.1. Uso de la tecnología .....	12
2.2.2. Lenguaje de programación Python .....	16
2.2.3. Frameworks para la creación de videojuegos.....	17

2.2.4. Comparativa y elección de tecnologías para la creación del videojuego .....	21
2.3. Introducción al motor de videojuegos Godot.....	22
2.3.1. Nodos .....	22
2.3.2. Escenas .....	24
2.3.3. Señales.....	25
2.3.4. Editor .....	26
2.4. Análisis de requerimientos .....	29
2.4.1. Idea general del videojuego.....	29
2.4.2. Arquitectura .....	30
2.4.3. Genero.....	30
2.4.4. Audiencia .....	30
2.5. Requerimientos del usuario .....	30
2.5.1. Actores.....	31
2.5.2. Definición de los casos de uso .....	31
2.5.3. Casos de uso.....	32
2.6. Requerimientos del sistema.....	38
2.6.1. Definición de requerimientos.....	38
2.6.2. Requerimientos funcionales.....	39
2.6.3. Requerimientos no funcionales.....	44
CAPÍTULO III .....	46
METODOLOGÍA.....	46
3. Metodología .....	46
3.1. Enfoque de la Investigación .....	46
3.1.1. Enfoque de investigación cualitativa .....	46
3.2. Diseño de investigación .....	47
3.2.1. Diseño no experimental .....	47
3.3. Nivel de investigación .....	48

3.3.1. Investigación exploratoria .....	48
3.4. Tipo de investigación .....	49
3.4.1. Investigación bibliográfica .....	49
3.5. Descripción de la propuesta.....	50
3.6. Población .....	50
3.7. Técnica e instrumentos .....	51
3.8. Metodología de desarrollo del videojuego.....	53
3.8.1. Metodología ágil SCRUM .....	54
CAPÍTULO IV .....	59
DISEÑO Y DESARROLLO DE LA APLICACIÓN .....	59
4. Diseño y desarrollo de la aplicación.....	59
4.1. Recolección de audio y animaciones .....	59
4.1.1. Sprites.....	59
4.1.2. Muestra de Sprites de la aplicación .....	60
4.1.3. Sprite Sheets .....	61
4.1.4. Muestra de Sprite Sheets de la aplicación.....	62
4.1.5. Audio.....	62
4.2. Diseño de la interfaz de usuario.....	63
4.2.1. Escena principal .....	64
4.2.2. Escena cuidado del agua.....	65
4.2.3. Escena cuidado de residuos.....	66
4.3. Diseño de la aplicación en UML.....	67
4.4. Desarrollo de la aplicación.....	68
4.4.1. Análisis .....	68
4.4.2. Historias de usuario .....	69
4.4.3. Product Backlog.....	69
4.4.4. Especificación de roles .....	70

4.4.5. Sprint 1 .....	70
4.4.6. Sprint 2 .....	72
CAPÍTULO V .....	82
CONCLUSIONES Y RECOMENDACIONES .....	82
5.1. Conclusiones:.....	82
5.2. Recomendaciones: .....	82
BIBLIOGRFÍA.....	83
ANEXOS .....	86

## ÍNDICE DE FIGURAS

Figura 1. Bucle de renderización de un videojuego.....	19
Figura 2. Grafo de nodos de Godot.....	23
Figura 3. Estructura de una escena en Godot.....	25
Figura 4. Ejemplo de señales en el editor de Godot.....	26
Figura 5. Editor de Godot. ....	27
Figura 6. Región de sistema archivos. ....	28
Figura 7. Región de inspección .....	28
Figura 8. Región de escenas, nodos y señales.....	28
Figura 9. Flechas de un teclado convencional .....	29
Figura 10. Caso de uso general del videojuego. ....	32
Figura 11. Ciclo de vida SCRUM.....	58
Figura 12. Tortuga animada .....	60
Figura 13. Fondo marino .....	60
Figura 14. Variedad de objetos .....	61
Figura 15. Tortuga caminando .....	62
Figura 16. Día y noche .....	62
Figura 17. Sonido de río y océano con efectos de agua recorriendo .....	63
Figura 18. Sonido de caída de objetos en agua .....	63
Figura 19. Escena principal con sus cuatro elementos .....	64
Figura 20. Escena cuidado del agua con sus elementos .....	65
Figura 21. Escena cuidado de los residuos con sus elementos.....	66
Figura 22. Escenas y jugador como clases.....	68
Figura 23. Criterio de aceptación: .....	75
Figura 24. Criterio de aceptación .....	76
Figura 25. Criterio de aceptación: .....	77
Figura 26. Criterio de aceptación: .....	78
Figura 27. Criterio de aceptación .....	79
Figura 28. Criterio de aceptación .....	80
Figura 29. Criterio de aceptación .....	81

## ÍNDICE TABLAS

Tabla 1. Unity .....	12
Tabla 2. Unreal Engine.....	13
Tabla 3. Godot Engine .....	14
Tabla 4. Lenguaje de programación Python.....	17
Tabla 5. LibGdx Framework .....	18
Tabla 6. Phaser Framework .....	20
Tabla 7. Comparativa y elección de tecnologías para la creación del videojuego	21
Tabla 8. Definición de los casos de uso .....	31
Tabla 9. Caso de uso 1 – Configuración de opciones .....	33
Tabla 10. Caso de uso 2 – Escenario cuidado del agua .....	33
Tabla 11. Caso de uso 3 – Escenario cuidado de residuos .....	34
Tabla 12. Caso de uso escenario cuidado del agu.....	34
Tabla 13. Caso de uso 4 – Configurar opciones .....	35
Tabla 14. Caso de uso 5 – Volver al escenario principal.....	35
Tabla 15. Caso de uso 6 – Accionar personaje.....	36
Tabla 16. Caso de uso escenario cuidado de residuos.....	36
Tabla 17. Caso de uso 7 – Configurar opciones .....	37
Tabla 18. Caso de uso 8 – Volver al escenario principal.....	37
Tabla 19. Caso de uso 9 – Accionar personaje.....	38
Tabla 20. Ejemplo de formato de requisitos funcionales y no funcionales .....	39
Tabla 21. Requisito funcional 1 – Intercambiar escenas .....	40
Tabla 22. Requisito funcional 2 – Cargar escenas .....	40
Tabla 23. Requisito funcional 3 – Mostrar configuraciones .....	41
Tabla 24. Requisito funcional 4 – Cambiar idioma .....	41
Tabla 25. Requisito funcional 5 – Activar o desactivar audio .....	42
Tabla 26. Requisito funcional 6 – Validación de personaje en escenario.....	42
Tabla 27. Requisito funcional 7 – Sugerencias en escenario .....	43
Tabla 28. Requisito funcional 8 – Mostrar el menú de ganar o perder .....	43
Tabla 29. Requisito no funcional 1 – Eficiencia al procesar múltiples entradas de teclado.....	44
Tabla 30. Requisito no funcional 2 – Fiabilidad en los gráficos que se muestran	44
Tabla 31. Requisito no funcional 3 – Disponibilidad de navegabilidad.....	44

Tabla 32. Requisito no funcional 4 – Mantenibilidad de configuraciones .....	45
Tabla 33. Elementos de la escena principal .....	65
Tabla 34. Elementos de la escena cuidado del agua .....	66
Tabla 35. Elementos de la escena cuidado del agua .....	67
Tabla 36. Especificación de roles.....	70
Tabla 37. Sprint Backlog .....	71
Tabla 38. Sprint Backlog .....	73

## ÍNDICE DE ANEXOS

Anexos 1. Guion de la entrevista para recolección de historias de usuario .....	86
---	----

# **CAPÍTULO I**

## **INTRODUCCIÓN**

### **1. MARCO DE REFERENCIA**

#### **1.1. JUSTIFICACIÓN**

La realización de la aplicación se fundamenta en la necesidad de entender la realidad que se vive actualmente: fluctuaciones en la economía mundial, culturales o demográficos (United Nations, s. f.). Todos ellos factores muy importantes en el desarrollo de las actividades de la humanidad, sin embargo, nuestro “hogar” que en otras palabras es el mundo o tierra en el que habitamos está teniendo cambios ambientales que perjudican la habitabilidad de todos los seres vivos que se encuentran en cualquier rincón del planeta. Lamentablemente, según (United Nations, s. f.) algunas cifras de las Naciones Unidas, los residuos sólidos como basura y otros desechos van en aumento cada año, por lo que las autoridades necesitan de nuevas generaciones que tengan educación sobre ecología y cuidado medioambiental.

Un videojuego es necesario desde los primeros años de vida de un ser humano, que es la etapa en que la cognición y proyección del futuro es donde más se define (Piaget & Inhelder, 1997). En los infantes, los juegos o aplicaciones lúdicas son más llamativas y se encuentra más asimilable la información (Piaget & Inhelder, 1997). Por tal motivo, impregnar de una cultura más amigable con el medio ambiente desde los años iniciales del ser humano es vital y se puede apoyar con un videojuego que demuestre ser interesante, pero a la vez básico de jugar y cuidar el medio ambiente.

#### **1.2. Planteamiento del problema**

El Internet está lleno de videojuegos que entretienen y aplicaciones que en la mayoría de los casos no se puede replicar en la vida de las personas, dígame videojuegos como tetris o ajedrez (Montagnana, 2008). La realidad actual requiere

personas, en este caso infantes que entiendan lo que sucede en la realidad específicamente con el cuidado de la casa común. Todo ello lleva a la necesidad de llevar a los niños de manera abierta a través de internet que conozcan, jueguen y aprendan sobre el problema del cambio climático y cómo debemos actuar frente a ella. Con pasos simples el infante aprenderá casos prácticos y qué debe realizar, y éstos mismos quedarán impregnados para su vida adulta.

Utilizando las tecnologías de la información se puede dar un impulso a generaciones jóvenes sobre la conciencia y desarrollo de una cultura de amigabilidad con nuestra casa o medio ambiente. Papa Francisco aconseja a la comunidad católica que se debe cuidar de la casa común (Francisco, 2015), hogar de los seres vivos. Por tanto, en el siglo veintiuno se cuenta con formas de acceder y educar a niños de manera masiva a través de las tecnologías de la información en conjunto con una aplicación, en específico un videojuego.

### **1.3. Objetivo General**

Desarrollar un videojuego para enseñar a los niños el cuidado del medio ambiente.

### **1.4. Objetivos Específicos**

- Analizar videojuegos que tengan un balance entre educación para el cuidado del medio ambiental.
- Definir los requerimientos funcionales para el videojuego.
- Diseñar una aplicación-videojuego que tenga transversalidad con desarrollo de valores ambientales, uso de herramientas tecnológicas y diversión.
- Evaluar la funcionalidad del videojuego a través de pruebas con niño/s, para evaluar sus resultados.

### **1.5. Alcance**

El presente trabajo de titulación culminará con la entrega del videojuego desarrollado como aplicación de escritorio, que enseñe principios de cuidado medio ambiental de una manera lúdica y comprensible.

La aplicación-videojuego se centrará en desarrollar conciencia en los escolares de cómo proceder en la vida real con el cuidado del medio ambiente, dígame clasificación de desechos o tratamiento especial de sustancias tóxicas para la naturaleza.

## CAPÍTULO II

### FUNDAMENTACIÓN TEÓRICA

#### 2. Marco Teórico

##### 2.1. Medio ambiente

###### 2.1.1. El cuidado del medio ambiente – Laudato Si

El medio ambiente o planeta tierra en general, es nuestro hábitat por origen y del cual los seres humanos han evolucionado. Pasando por duras circunstancias de las cuales han hecho el hombre que es hoy en día, se ha ido forjando tanto por adversidades climáticas como de enfermedad y hambruna. (*El cambio climático: sus efectos a nivel mundial y su regulación en el Derecho Internacional.*, s. f.). Ahora el humano se encuentra en el siglo veintiuno el cual está rodeado de tecnología y nuevos ideales que desgraciadamente es causante de grandes dolores al planeta, en su falta de conocimiento y práctica en la organización de la basura, así como la clasificación de desechos tóxicos. Todo ello ha conllevado a que grandes organizaciones se vean obligadas a dar voz y autoridad en canales mundiales, ya sea por televisión, radio o internet.

La solución más pertinente que han realizado las organizaciones de más poder en el mundo para evitar el calentamiento global es exclusiva de ellos, los cuales van desde voceros hasta ejecutores de la ley que multan y aplican leyes a personas que atentan con el medio ambiente a través de sus acciones que perjudican la vitalidad, la pureza y la estética naturalmente perfecta de la naturaleza. Entre la religión, encontramos una en particular que es el catolicismo que nos ha dado un personaje que es la máxima autoridad, el Papa Francisco. Consciente de lo que sucede al planeta se ha dispuesto a crear un escrito que nos ha dado luz sobre cómo proceder cuando nos encontremos con la naturaleza del mundo actual (*Laudato si' (24 de mayo de 2015) | Francisco, 2015*).

Papa Francisco nos insta a proteger y mantener el medio ambiente como si fuese nuestro prójimo pues es lo que Dios no ha dado como refugio y de sustento para

alimentación incluso. Los seres vivos que habitan también son valiosos y deben incluirse dentro de este grupo del cual debemos proteger y mantener. Haciendo analogía con la Biblia, nos indica que la ausencia de valores se ha convertido en un reflejo en nuestros actos hacia la naturaleza. Aunque sin dejar de lado que la falta de consciencia en las personas también es una muestra de que se debe reformar a la sociedad, sin embargo, no todos los países están en condiciones económicas y geográficas para aplicar en totalidad las prácticas medioambientales que mejorarán o mantendrán el planeta en un equilibrio neutral con sus habitantes y viceversa.

## **2.1.2. Buenas prácticas del cuidado del medio ambiente**

### **2.1.2.1. Agua**

El agua es un líquido vital que permite a los humanos y habitantes del planeta tierra mantenerse en forma y poder alimentarse. Tiene entre otras funciones secundarias las de hidratar plantas, bosques y selvas que naturalmente producen oxígeno para nuestro consumo.

Su cuidado se da principalmente en dos funciones: ahorro y consumo eficiente. A continuación, se muestra textualmente cómo debe ser tratado el agua para su eficiente utilización (Guía de buenas prácticas ambientales, 2018):

- Implantación de sistemas de ahorro en el consumo de agua potable en grifos e inodoros, mediante la instalación de temporizadores, perlizadores, reductores de caudal, contrapesos o grifería termostática.
- Mejora del aislamiento de los sistemas de agua (en depósitos acumuladores y tuberías de distribución de agua caliente).
- Instalación de grifos monomandos en los espacios donde se requiera agua caliente y fría, ello conlleva ahorros del 50%.

- Solucionar de inmediato posibles goteos y fugas en grifos, puesto que una pérdida de una gota por segundo puede generar un derroche de 30 litros al día, es decir, 11 m<sup>3</sup> por año.
- Sensibilizar a los trabajadores para evitar desechar desperdicios en el inodoro de los aseos.
- En las oficinas de edificios compartidos, demandar contadores individualizados de agua para controlar los consumos propios.
- Elección de plantas interiores con criterios ambientales, que favorezcan el ahorro de agua y que estén adaptadas a la climatología del entorno.”

### **2.1.2.2. Residuos**

Naturalmente todos los alimentos procesados o no procesados tienen un envoltorio del cual conserva su estado y hermeticidad. Una vez consumido estos generan residuos de diferentes materiales, como pueden ser: plásticos, papel y orgánicos. Normalmente estos residuos considerados no tóxicos son necesarios que estén clasificados y no tengan ningún contacto con el medio ambiente. A continuación, se muestra textualmente cómo debe ser tratado los residuos para su eficiente utilización:

- Establecer un inventario de clases de residuos que son generados en la oficina, para consecutivamente diseñar un apropiado sistema de recogida selectiva.
- Instaurar un sistema de recogida selectiva en contenedores adecuados y compartimentalizados: envases de plástico y aluminio (papelera color amarillo), papel y cartón (color azul), vidrio (color verde) y desechos orgánicos (color marrón o negro).
- Promover la sensibilización de los empleados en la separación adecuada de residuos, especialmente producidos en el comedor de la oficina.

- Apostar por la compra de productos a granel o compactadas, y optar por materiales con embalajes mínimos para minimizar el volumen de residuos producidos.
- Comprar productos y otros artículos que cuenten con el símbolo identificativo de adhesión a algún Sistema Integrado de Gestión de Residuos, garantizando así garantice su reciclado y valorización posterior.
- Contratar entidades autorizadas para la gestión de residuos, dando preferencia a gestores que utilicen técnicas de reutilización, reciclado, valorización o transformación, siendo las últimas opciones la destrucción por incineración o depósito en vertedero.
- Reducción del consumo de suscripciones y publicaciones, minimizando el significativo impacto en el medio ambiente.
- Gestión adecuada de medicamentos y material sanitario del botiquín, depositándolos en puntos SIGRE de la farmacia más cercana.”

### **2.1.3. Videojuegos**

Los videojuegos son programas informáticos que se utilizan para jugar y que pueden ser diseñados para una amplia variedad de plataformas, desde computadoras hasta consolas de videojuegos y dispositivos móviles (Montagnana, 2008). En esencia, los videojuegos son una forma de entretenimiento interactivo en la que los usuarios pueden controlar lo que sucede en la pantalla y pueden interactuar con otros jugadores, ya sea en línea o en persona (Montagnana, 2008). Los videojuegos también se han utilizado en algunos entornos educativos y de capacitación para ayudar a los usuarios a adquirir habilidades y conocimientos específicos (Montagnana, 2008).

#### **2.1.3.1. Tipos de videojuegos**

Los videojuegos se clasifican en tres grupos grandes, de acuerdo a tres criterios (Montagnana, 2008):

- Por género: los videojuegos se pueden clasificar según su temática o género, como acción, aventura, deportes, estrategia, simulación, entre otros.
- Por nivel de dificultad: algunos videojuegos están diseñados para ser más desafiantes que otros, y se pueden clasificar en función de su nivel de dificultad, como fácil, medio o difícil.
- Por plataforma: los videojuegos se pueden clasificar según la plataforma en la que se juegan, como consolas de videojuegos, computadoras personales, dispositivos móviles, entre otros.

### **2.1.3.2. Problemas de los videojuegos**

A pesar de que los videojuegos son una forma de entretenimiento muy popular, también hay preocupaciones y problemas vigentes asociados con ellos. A continuación, se describe algunos de los problemas más comunes que se relacionan con los videojuegos (Montagnana, 2008):

- Adicción: algunos jugadores pueden desarrollar una adicción a los videojuegos, lo que puede afectar su salud mental, su vida social y su rendimiento académico o laboral.
- Contenido inapropiado: algunos videojuegos pueden contener contenido violento, sexual o de otro tipo que no es adecuado para ciertas edades o públicos.
- Impacto en la salud física: el uso excesivo de videojuegos puede llevar a problemas de salud física, como dolor de espalda, fatiga visual, obesidad y otros problemas asociados con el sedentarismo.
- Impacto en la salud mental: algunos estudios han sugerido que el uso excesivo de videojuegos puede tener un impacto negativo en la salud mental, como ansiedad, depresión y trastornos del sueño.

- Adquisición de habilidades sociales limitadas: los videojuegos pueden limitar la capacidad de los jóvenes para desarrollar habilidades sociales y emocionales, como la empatía y la resolución de conflictos.

### **2.1.3.3. Impacto positivo de los videojuegos educativos**

Los videojuegos se muestran en las aulas como una alternativa la cual se considera innovadora tanto por su llamativa forma de mostrarse como lo intrincado que puede resultar mover algún periférico de una computadora (Núñez-Barriopedro, 2020). La enseñanza tradicional siempre conlleva utilizar materiales tradicionales como lápiz, cuaderno o libros para tomar notas y apuntes. Todo ello visto desde un punto de vista muy simple si lo comparamos con un videojuego del cual primero que nada enciende las emociones de los estudiantes porque se trata de algo visualmente estimulante y también es algo que genera curiosidad de cómo mecánicamente se opera y se soluciona problemas o completa tareas con el mismo. Aunque parezca una idea nada novedosa, muchas instituciones no la implementan porque se sabe que el valor económico de instalar y configurar computadores, como mínimo para enseñanza de informática ya de por sí resulta costoso en países que se consideran en vías de desarrollo. Tomando como partida que un videojuego de carácter educativo tiene la necesidad de complementarse con usuarios, los mismos deben ser capaces de solventar los movimientos o acciones que a través de periféricos sean capaces de completar tareas. Una pregunta sería, entonces surge la cuál es: ¿A caso los niños ya aprendieron a través de sus hogares el manejo de dispositivos digitales?, la respuesta es sí por que en países que consideramos en vías de desarrollo, en aquellos se muestra la tendencia a obtener equipos, como mínimo celulares que facilitan la comunicación con otras personas. Estos equipos poseen tanta tecnología como software y hardware que son capaces de correr videojuegos en dos o incluso tres dimensiones.

Los dispositivos electrónicos que incluso se consideran de bajos recursos ya tienen tanta capacidad de procesamiento, almacenamiento y conectividad entre ellos que juegos de altas especificaciones que eran inimaginables que corran en celulares

de bajo costo lo pudieran hacer. Esto es otra realidad debido a la miniaturización de los componentes de hardware, así como la mejora de software en arquitectura y algoritmos (Ley de Moore – Ahora y en el Futuro, s. f.). Aunque no necesariamente un videojuego más sofisticado implique más impacto educativo, este concepto puede usarse para mejorar la atención de los niños que usarán el videojuego. Ante nuevas formas, texturas y colores que se presentan gracias al diseño gráfico, estos cada vez se innovan para recrear realidades y que éstas sean aprovechadas como juegos, educación y de manera profesional para un fin.

El impacto es positivo en los niños siempre y cuando se cuente con el equipo y software necesario. Los niños tendrán una retentiva más larga en cuanto a su aprendizaje, así como mejoras en sus habilidades para el uso de los periféricos computacionales, sin dejar de lado que el tópico el cual se pretende enseñar será aprendido y retenido para un futuro no muy lejano, el cual es su vida adulta y el que dará cuenta de sus valores y aprendizajes cuando fuere necesario aplicar. Lo que se logra con los videojuegos se puede listar a continuación (Montagnana, 2008):

- Mejoran la capacidad de respuesta.
- Fomentan el trabajo en equipo.
- Estimulan la creatividad, la atención y la memoria visual.
- Mejoran la estrategia y el liderazgo.
- Enseñan idiomas.
- Favorecen el pensamiento crítico.

En conclusión, los videojuegos tienen su aporte que va más allá del tema de diversión y ocio, pueden mejorar su rendimiento cognitivo, social y físico. Todo ello ha conllevado que los niños mejoren como alumnos y personas de la sociedad en la que la tecnología ha tenido bastante relevancia.

#### **2.1.4. Videojuegos y cuidado medioambiental**

Las generaciones que nacieron próximos a inicios del siglo veintiuno son aquellas personas que naturalmente son usuarios digitales de celulares, tablets o laptops como usuarios finales. Los videojuegos de cuidado medioambiental pueden penetrar la vida de los niños si logran que se diviertan o al menos interaccionen y se haya cultivado principios que serán necesarios para el cuidado del medio ambiente.

Los videojuegos han mostrado ser más efectivos a la hora de enseñar cualquier temática, porque requiere que el niño razone, comprenda y asimile lo que ve, caso contrario sucede con una charla de la cual la atención se puede esfumar rápidamente (EFEverde, 2022). Se pueden fomentar todo tipo de actividades sin siquiera tener presente las herramientas que hagan poner en práctica eso, como lo hemos recalado un mundo digital es zona para recrearse y fomentar prácticas saludables. A continuación, se presentan algunos videojuegos que han sido referentes en este tipo de temáticas.

##### **2.1.4.1. Beyond Blue**

Beyond Blue es un videojuego el cual un científico navega por los océanos y enseña sobre diferentes especies de animales acuáticos, así como tipo de plantas y sus alrededores. Estos son necesario como fundamento para que las generaciones entiendan de que se compone la naturaleza y la reacción que tiene la basura en los océanos.

##### **2.1.4.2. Bee Simulator**

Este videojuego refleja uno de los actores importantes que permiten que las plantas puedan ser mantenidas en buen estado y mejoren el ecosistema que rodea. El videojuego permite controlar a abejas que protegen el polen y además es multijugador lo cual lo hace más interesante para socializar e intercambiar ideas sobre este actor tan importante en la naturaleza.

## 2.2. Videojuegos y tecnología

### 2.2.1. Uso de la tecnología

A continuación, se presenta las tecnologías que se van a emplear para un análisis final del cual se podrá sacar conclusiones de qué tecnologías pueden resultar beneficiosos para desarrollar un buen producto en un corto período de tiempo.

#### 2.2.1.1. Motores para la creación de videojuegos

En la actualidad existen una gran variedad de frameworks que facilitan la creación de videojuegos en diferentes plataformas y lenguajes de programación. Realizar un análisis de ventajas y desventajas de diferentes frameworks será de mucha utilidad para considerar la elección de este.

##### 2.2.1.1.1. Unity

Unity es un motor gráfico para el desarrollo de videojuegos el cual está considerado como el más popular actualmente. Además de videojuegos también permite la creación de diversidad de animaciones en dimensiones 2D y 3D con lo cual se considera un motor gráfico de mucha utilidad tanto para el diseño como para el desarrollo de software a la par (Mula, 2022). En el análisis de ventajas y desventajas se tiene el siguiente cuadro:

**Tabla 1. Unity**

Ventajas	Desventajas
<b>Es gratuito sin embargo cualquier iniciativa creada en Unity deberá ser pagada.</b>	La gestión de memoria en las librerías de .NET es ineficiente en la administración de memoria.

<b>Contiene mucha documentación tanto en su página principal como libros y blogs.</b>	Unity requiere un rendimiento excepcional sobre el hardware que se va a desarrollar.
<b>Es multiplataforma, es decir permite la creación de videojuegos que puedan ser web, escritorio y móvil.</b>	Los parches que Unity ofrece muchas veces son intermedios con lo cual es inestable desarrollar sobre tal parche y se debe mantener en el anterior.

**Fuente:** Fausto Sebastián Borja Villacís

### 2.2.1.1.2. Unreal Engine

Unreal Engine es un motor gráfico para desarrollo de videojuegos que principalmente se caracteriza por poseer herramientas avanzadas para la creación de videojuegos en 3D. Es de uso gratuito, sin embargo, para uso comercial la casa comercial cobra una comisión dependiendo del valor del producto (Mula, 2022). En el análisis de ventajas y desventajas se tiene el siguiente cuadro:

**Tabla 2. Unreal Engine**

<b>Ventajas</b>	<b>Desventajas</b>
<b>Tiene la posibilidad de ser gratuito bajo ciertas condiciones.</b>	Comparado a Unity su comunidad es muy pequeña.
<b>Su motor gráfico es de acceso gratuito por lo cual se puede manipular su código sin concesiones.</b>	Su curva de aprendizaje es alta debido a que su lenguaje de programación es en C++.

<b>Tiene extensa documentación disponible.</b>	Ciertos conceptos resultan complicados de aprender dado su complejidad.
--	---

**Fuente:** Fausto Sebastián Borja Villacís

### 2.2.1.1.3. Godot Engine

Godot es un motor de videojuegos que de uso general que permite crear videojuegos en dimensiones 2D y 3D. Es multiplataforma, por lo que puede desarrollarse videojuegos tanto de escritorio, en la web o incluso para móviles (Introducción a Godot, s. f.). Tiene sus principales lenguajes de programación para scripting de los cuales se derivará la animación y diseño del videojuego. En un cuadro de ventajas y desventajas encontramos lo siguiente:

**Tabla 3. Godot Engine**

<b>Ventajas</b>	<b>Desventajas</b>
<b>Tiene una interfaz amigable que puede ser usada por principiantes en desarrollo.</b>	No tiene capacidades para creación de videojuegos con gráficas avanzadas.
<b>Tiene soporte para juegos en 2D y 3D</b>	No tiene algunos comportamientos físicos en la dimensión 2D
<b>Es multiplataforma.</b>	No es recomendable para usuarios con experiencia en desarrollo de videojuegos.

**Fuente:** Fausto Sebastián Borja Villacís

## **2.2.1.2. Lenguajes de programación**

### **2.2.1.2.1. Lenguaje de programación C++**

C++ es un lenguaje de programación que soporta los conceptos relacionados con objetos, los cuales entre muchos son: herencia, polimorfismo, encapsulación, entre otros. Su creador fue Bjarne Stroustrup de nacionalidad danesa en la década de los ochenta. El lenguaje C++ al estar basado en el lenguaje C, soporta el paradigma estructurado con lo cual es un lenguaje multiparadigma. Entre las principales ventajas de C++ encontramos las siguientes (Qué es C++: Características y aplicaciones, 2022):

- Alto rendimiento
- Lenguaje actualizado
- Multiplataforma
- Extendido

Como se podía esperar, todos los lenguajes de programación tienen sus desventajas, entre las que se destacan las siguientes:

- Su curva de aprendizaje es muy alta.
- Su depuración suele ser difícil en determinados casos.
- Su manejo de librerías es más complicado que otros lenguajes de programación como Java o .NET

### **2.2.1.2.2. Lenguaje de programación Java**

Java es un lenguaje de programación multiparadigma que fue creado por Sun Microsystems en el año 1995. A Java se le considera como un lenguaje de programación del cual la mayoría de los sistemas y aplicaciones están basados en él, esto empezaría entre la década de los noventa y a principios del siglo veintiuno. Su cultura se asocia a una taza de café, que sin lugar a duda es una práctica para crear ideas y revoluciones digitales. Su principal uso se lo da para emplear los paradigmas

de orientación a objetos, programación funcional y estructural. Las principales ventajas tenemos lo siguiente (Ventajas de Java, s. f.):

- Su curva de aprendizaje es baja.
- Es multiplataforma, por lo cual se dice codifica una vez, corre en muchas plataformas.
- Es orientado a objetos, por lo cual es útil para crear sistemas modulares y con código reutilizable.

Java no se excluye de tener algunos inconvenientes a la hora de planear utilizarlo o emplearlo en proyectos (Just a moment. . ., s. f.):

- Tiene pobre rendimiento en algunas circunstancias debido a que consume mucha memoria en aplicaciones extensas.
- La lectura de su código puede resultar complicada de leer debido a que es muy verboso en cada sentencia.
- El desarrollo de sus interfaces de usuario es simple y poco atractivos al usuario.

### **2.2.2. Lenguaje de programación Python**

El lenguaje de programación Python es un lenguaje multipropósito en el que se puede desarrollar aplicaciones web, software en general y también sirve para ciencia de datos y machine learning, en éstos últimos es más ampliamente usado debido a su facilidad de aprendizaje y por ende varios profesionales e investigadores ahorran tiempo en aprender un lenguaje e implementan lo más rápido algoritmos y estructuras programáticas (¿Qué es Python? - Explicación del lenguaje Python - AWS, s. f.). A continuación, se presenta una serie de ventajas y desventajas sobre Python:

**Tabla 4. Lenguaje de programación Python**

<b>Ventajas</b>	<b>Desventajas</b>
<b>Es un lenguaje de alto nivel y fácil de aprender debido a su fácil sintaxis y carencia de tipos.</b>	El consumo de memoria es alto comparado a otros lenguajes de programación.
<b>Es un lenguaje de alto nivel, por lo que no requiere manejo del hardware.</b>	No es muy apropiado para desarrollo móvil por carecer de características que hagan fácil el desarrollo.
<b>Es multiparadigma por lo que es fácil de usar en la mayoría de las aplicaciones que requieran conceptos de programación funcional, de objetos o estructurada.</b>	Es necesario en aplicaciones avanzadas la creación de perfiles para optimizar recursos de hardware.

Fuente: Fausto Sebastián Borja Villacís

### **2.2.3. Frameworks para la creación de videojuegos**

#### **2.2.3.1. LibGdx Framework**

LibGdx es un framework para desarrollo de videojuegos que utiliza Java como principal lenguaje para desarrollar. Puede por su puesto usar los principios de diseño orientado a objetos como reutilización de código entre otros. Es multiplataforma como los anteriores frameworks mencionados. A continuación, se puede observar una tabla de ventajas y desventajas de Libgdx (libGDX vs Godot, s. f.).

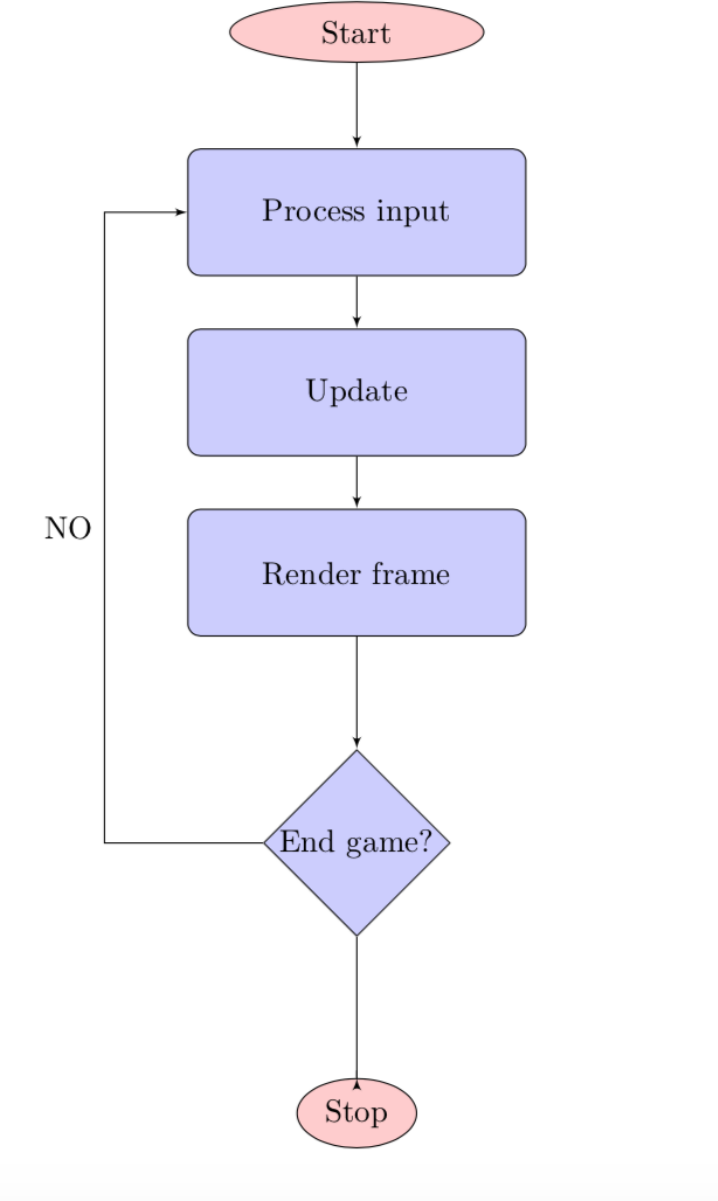
**Tabla 5. LibGdx Framework**

<b>Ventajas</b>	<b>Desventajas</b>
<b>Puede usarse desde Java 8</b>	No tiene soporte para el motor gráfico Vulkan
<b>Es muy liviano para dispositivos móviles</b>	La documentación no es basta
<b>Tiene acceso a la librería Box2D</b>	Su build system es Gradle, exclusivamente

**Fuente:** Fausto Sebastián Borja Villacís

LibGdx hace uso del bucle que procesa los videojuegos en sus etapas que comprenden desde la entrada de datos hasta la renderización, como se muestra en la siguiente figura:

**Figura 1. Bucle de renderización de un videojuego.**



Fuente: (G, 2022)

Este tipo de algoritmo es aplicable a cualquier tipo de framework o motor de gráfico el cual procese la entrada del usuario y actualice los gráficos conforme si la lógica del juego requiera movimientos o cambios de texturas y finalmente estos se actualizan y el bucle empieza de nuevo.

### 2.2.3.2. Phaser Framework

Phaser es un framework exclusivamente para navegadores que soporten Canvas y WebGL. Este framework en específico trabaja con el tag Canvas del cual se puede crear, actualizar y renderizar gráficos. Se necesita un navegador que soporte tales características y también es recomendable conocer y dominar los rudimentos básicos de Javascript, HTML y CSS. A continuación, se puede visualizar un cuadro de ventajas y desventajas de Phaser:

**Tabla 6. Phaser Framework**

<b>Ventajas</b>	<b>Desventajas</b>
<b>Su desarrollo es altamente eficiente, con lo cual se consiguen productos en menor tiempo comparado a otros frameworks.</b>	No tiene herramientas auxiliares, apoyar la producción de productos en línea a través de equipos de trabajo resulta complicado.
<b>Tiene excelentes capacidades de renderización gracias al script pixi.js.</b>	Suelen tener cuellos de botella cuando la memoria es insuficiente en el hardware al cual se está reproduciendo el juego.
<b>Tiene ejemplos de código muy vastos con lo cual aprender este framework es fácil.</b>	Sólo puede desarrollarse juegos en 2D.

Fuente: Fausto Sebastián Borja Villacís

## 2.2.4. Comparativa y elección de tecnologías para la creación del videojuego

De acuerdo con la teoría expuesta en la que se expone definición con ventajas y desventajas asociadas a cada tecnología tenemos la siguiente tabla comparativa que nos ayudará a discernir que tecnologías emplear en la realización del videojuego:

**Tabla 7. Comparativa y elección de tecnologías para la creación del videojuego**

	Lenguaje de programación	Documentación disponible	Editor gráfico o	Desarrollo multiplataforma	Curva de aprendizaje
Unity	C++	extensa	si	si	Larga y difícil
Unreal Engine	C++	extensa	si	si	Medianamente difícil
Godot	GScript (Python)	corto	si	si	Fácil de aprender
LibGdx	Java	corto	no	si	Difícil y larga
Phaser	HTML, CSS, Javascript	extensa	no	no	Difícil y larga

Fuente: Fausto Sebastián Borja Villacís

Como podemos ver Godot reúne las características que lo hacen ideal para un desarrollo ágil del cual saldrá un buen producto. Tenemos un editor gráfico, un lenguaje de programación fácil de emplear y sobre todo su documentación es corta, pero esto no resulta un inconveniente porque Godot tiene una infinidad de características, pero el software que se realizará no es extenso y no abarcará todas las funciones de Godot como es de esperarse.

## **2.3. Introducción al motor de videojuegos Godot**

Godot es un motor de gráficos que permite el desarrollo de videojuegos a través de su exclusivo IDE, el cual tiene todas sus características relacionadas uno a uno, es decir no tiene más ni menos de lo que Godot puede ofrecer para su desarrollo. Su principal enfoque es en videojuegos 2D por lo que su desarrollo es ágil y rápido. Tiene herramientas de diseño y animación que permite personalizar a la preferencia del usuario y según el contexto del juego los gráficos y la física de este (Engine, s. f.).

Godot tiene en su estructura para desarrollo la posibilidad de organizar las escenas en un esquema de grafos, en los que variables, actores con sus posiciones se colocan de manera jerárquica en el grafo, existen dos elementos fundamentales que este grafo contiene (Engine, s. f.):

- Nodos
- Escenas

### **2.3.1. Nodos**

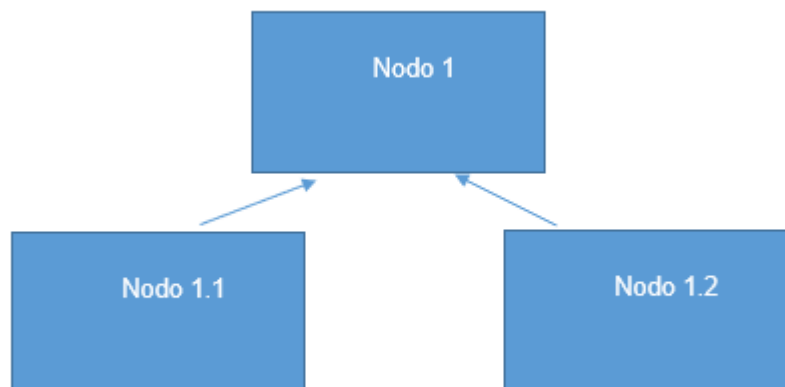
Un nodo es una entidad que se considera lo más elemental en Godot puesto que hace función de objeto que agrupa varias propiedades. Entre las cuales tenemos lo siguiente:

- Tiene un nombre específico

- Sus propiedades son editables
- Puede ser llamado mediante funciones que procesarán los fotogramas
- Puede aplicarse herencia para extender un nodo existente y crear otro a partir de este

Por lo que, si se considera a los nodos como clases que pueden contener objetos o instancias en una aproximación sumado a que la herencia o la extensión de un nodo puede aplicarse, sin problemas un gráfico como el siguiente lo puede representar:

**Figura 2. Grafo de nodos de Godot.**



**Fuente:** Fausto Sebastián Borja Villacís

Usualmente al crear nodos que sean específicos del juego, nodos padres de los cuales es necesario extender es obligatorio si lo que se requiere es tener funcionalidades especializadas. Estos nodos que contienen funciones especiales son (Engine, s. f.):

- Area2D: Permite crear eventos los cuales notifican cuando un objeto entra a una zona o sale de ella.
- Sprite: Registra la textura de un evento.

- **AnimatedSprite:** A partir de un conjunto de imágenes permite la creación de un objeto que tiene animación.
- **KinematicBody2D:** Utilidad para crear objetos 2D que son independientes.
- **CollisionShape2D:** Cualquier objeto 2D puede solidificarse.
- **Tilemap:** Permite la creación de una escena que toma objetos 2D almacenados en una cuadrícula.
- **Label:** Etiqueta que muestra texto.
- **CanvasLayers:** Permite la intercepción de un objeto 2D durante la ejecución del videojuego.
- **Tween:** Permite añadir animación a los nodos de manera simple.
- **Panel:** Permite contener gran variedad de nodos.
- **Timer:** Permite evaluar el tiempo en un intervalo dado.

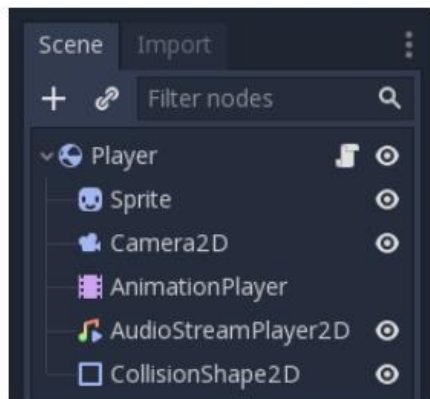
Puede observarse que los nodos tienen similitudes con lenguajes de programación que soportan el paradigma de objetos. Tienen clases o nodos que le son útiles para especializarse y obtener características y funciones que lo hacen único y personalizan el juego.

### **2.3.2. Escenas**

Las escenas en Godot son aquellas que pueden organizar y almacenar nodos que realizarán tareas en conjunto o de forma independiente. Las escenas además de almacenar nodos orquestan la ejecución de estos. Una escena tiene las siguientes características (Engine, s. f.):

- Parten de un nodo raíz.
- Se guardan y cargan en disco.
- Se puede instanciarles para dar paso a una escena.

**Figura 3. Estructura de una escena en Godot.**



**Fuente:** (*Engine, s. f.*)

Las escenas son en otras palabras, escenarios en los cuales los nodos que están en él interactúan y según la secuencia de fotogramas del videojuego, las escenas se instancian o llaman de acuerdo con el libreto. Cabe recordar que siempre la ejecución del videojuego empieza por una escena principal o ejecutoria que es el inicio y posiblemente pueda resultar el fin de la lógica del juego.

### 2.3.3. Señales

Las señales son mensajes que un evento de un nodo envía a otro nodo para que realice una acción en específico. Estos son análogos en un parecido a los lenguajes de programación como Java, C# o Python, que usan los eventos para notificar a bloques de código que realicen determinada tarea. Acá en Godot existen dos formas de conectar una señal con un nodo:

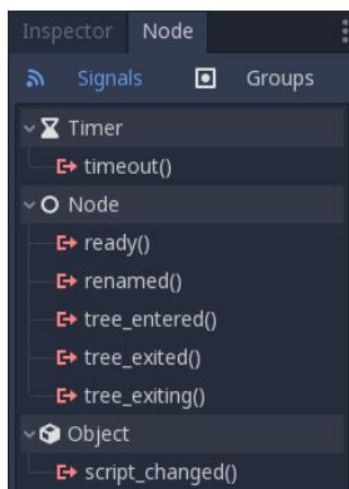
- Conexión de señal por código

Como su nombre indica, la conexión o envío de una señal de un nodo a otro por código no es más que la instanciación de un nodo cuando un determinado evento surge a través de un método o función que se implemente.

- Conexión de una señal por editor de Godot

Las señales que vienen incorporadas en el editor son las más fáciles de incorporar a los videojuegos porque son fáciles de implementar en el código. Es la manera más fácil de realizar implementaciones de señales, se lo hace cuando no se cuenta con alta experiencia en programación. Sin embargo, no por ser sencillo signifique que sea muy rudimentario, todo lo contrario, pueden ser esenciales en proyectos sofisticados.

**Figura 4. Ejemplo de señales en el editor de Godot.**



**Fuente:** (Engine, s. f.)

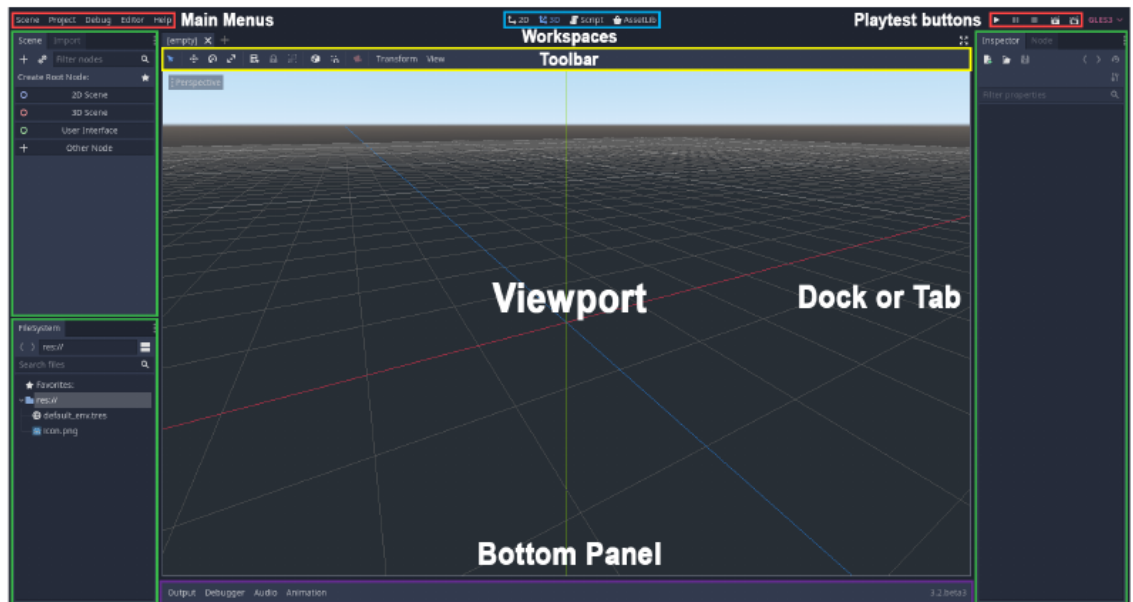
Las señales son de gran importancia cuando se requiera tener un código limpio, puesto que determinadas características sólo son lanzadas cuando algún suceso ocurre, por lo que no se fuerzan a los nodos a entrar en escenarios específicos. Se implementan señales como escuchas para realizar un conjunto de acciones que harán de solución en un contexto dado del videojuego.

### 2.3.4. Editor

El editor es el punto de partida de todo proyecto de Godot, porque en primera instancia el mismo incorpora todas las funcionalidades y características que permitirán la creación de un videojuego en 2D que tenga tanta sofisticación como se requiera

según los requerimientos. El editor se caracteriza por tener varios paneles distribuidos según lo que se requiera ocupar:

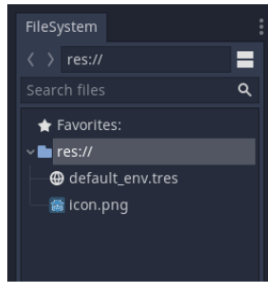
**Figura 5. Editor de Godot.**



**Fuente:** (Engine, s. f.)

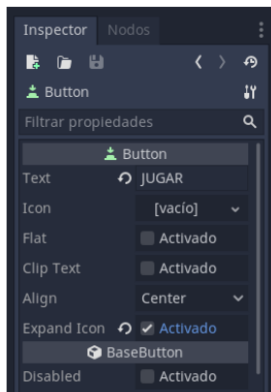
Las regiones enmarcadas con rojo son esenciales para interactuar con el menú principal, así como las zonas de trabajo, también pausa y ejecución del proyecto. La región enmarcada con verde y que se sitúa en la parte superior izquierda es aquella para configuración de escenas, nodos y características inherentes a ellos. La región de color verde y situada en la parte inferior izquierda del editor sirve para la visualización y gestión de los archivos del proyecto. La región verde situada en la parte derecha se le llama Inspector y sirve para la edición de propiedades, plegar y desplegar áreas entre otros. Las siguientes figuras ilustran las regiones antes mencionadas.

**Figura 6. Región de sistema archivos.**



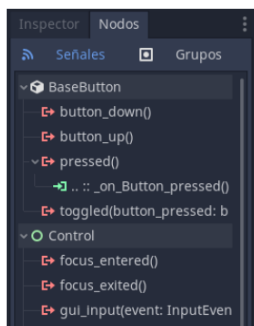
Fuente: (Engine, s. f.)

**Figura 7. Región de inspección**



Fuente: (Engine, s. f.)

**Figura 8. Región de escenas, nodos y señales.**



Fuente: (Engine, s. f.)

Como se ha visto, el editor además de componerse de características que lo hacen un IDE propio y completo también tiene funciones para edición de fotogramas

o escenarios gráficos que componen el juego ya sea en 2D o 3D. Todo ello traduce los esfuerzos del programador a la lógica propia del juego sin tener en cuenta detalles que están implementados bajo telón.

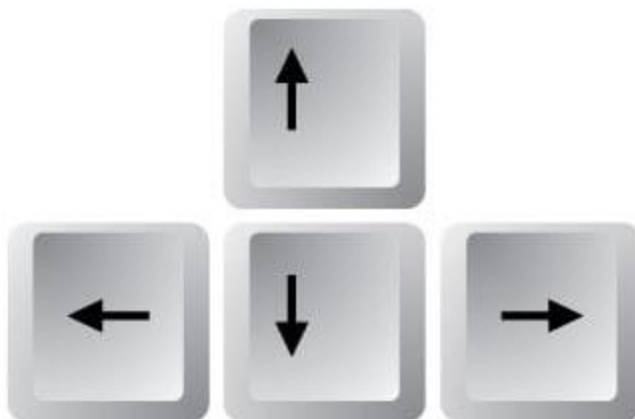
## 2.4. Análisis de requerimientos

### 2.4.1. Idea general del videojuego

El videojuego tiene el objetivo principal de enseñar sobre el cuidado medioambiental desde un punto de vista lúdica, es decir, sin hacerlo mediante conceptos o textos que de alguna forma resultan rudos para la audiencia a la que se dirige que es de 5 a 6 años. Todo ello surge bajo la necesidad de ayudar a generar conciencia que el planeta se encuentra con cambios climáticos bruscos debido a la alteración de ecosistemas como el mar, tierra y los polos antárticos.

El videojuego se enfoca en tres ejes principales, que son la: curiosidad, diversión, prueba y error. Consistirá en aprobar varios obstáculos que le surgen y en general son tres escenarios que el jugador tiene que afrontar o divertirse y poner a prueba sus habilidades con el uso de teclas muy rudimentarias que se encuentran en cualquier computador, estos son las flechas del teclado que usualmente sirven para mover o arrastrar objetos.

**Figura 9. Flechas de un teclado convencional**



Fuente: (C, 2021)

### **2.4.2. Arquitectura**

La arquitectura de Godot se basa principalmente en el diseño orientado a objetos, el cual cumple con todos los conceptos de dicho paradigma, así como la característica intrínseca de trabajar con nodos, los cuales, a diferencia de otros motores de juegos, no son componentes sino más bien son objetos.

### **2.4.3. Genero**

El videojuego corresponde con el género educativo, en el que prima la enseñanza de conceptos de cuidado medioambiental, de una forma divertida, sin ser aburrido o estresante dado que el uso del videojuego será usando lo más sencillo de un teclado que cualquier teclado incorpora: las flechas del teclado.

### **2.4.4. Audiencia**

La audiencia a la cual está dirigida es a niños de 5 a 6 años, los cuales dispongan de al menos un computador para que pueda ser instalado y usado el videojuego, con requerimientos del sistema que más adelante se especificarán en los requerimientos del sistema. Es un videojuego que no contiene contenido violento o para mayores de dieciocho años, por lo que tiene coherencia con un videojuego meramente educativo y formativo.

## **2.5. Requerimientos del usuario**

Los requerimientos del usuario son aquellos que se definen con casos de uso, el cual el jugador va a realizar con la aplicación o videojuego. Estos casos de uso serán verificados al final de este trabajo de titulación. Por tanto, primero se definen los actor o actores de ser el caso y para cada uno de ellos realizar casos de uso que no es más que una tabla a forma de formulario que necesita ser escrito con claridad.

### 2.5.1. Actores

El videojuego será accionado por un solo jugador que hará uso de las teclas esenciales navegar y seleccionar opciones. El rol de este jugador que tiene una edad comprendida entre 5 a 6 años es de explorar los diferentes escenarios e intuir la acciones que se deben completar. Cabe resaltar que la intuición está destinada a ese tipo de audiencia.

### 2.5.2. Definición de los casos de uso

A continuación, se muestra el formato específico en forma de tabla que permitirá detallar los casos de uso del actor que se mencionó en la sección anterior.

**Tabla 8. Definición de los casos de uso**

identificador	
Nombre	
Actor	
Descripción	
Precondición	
Post condición	

**Fuente:** Fausto Sebastián Borja Villacís

Los casos de uso se componen de los elementos que en la Tabla 9 se observan. Sus elementos se describen a continuación:

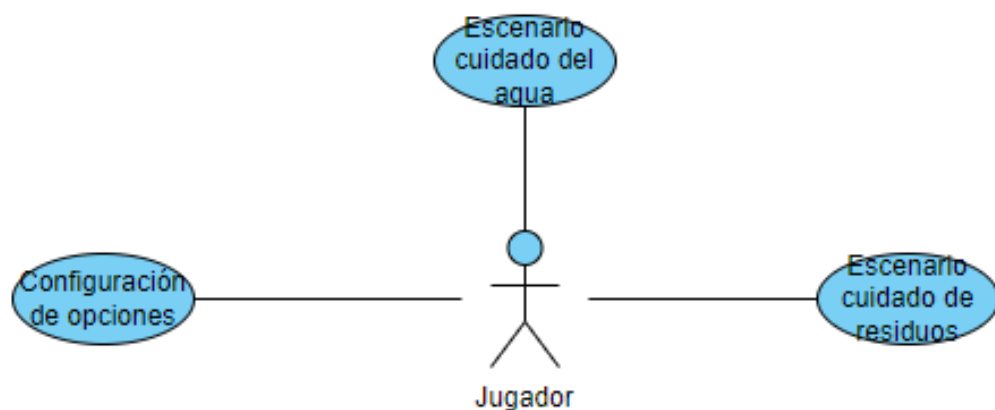
- Identificador: Identifica de manera única al caso de uso.
- Nombre: Es lo que realiza el caso de uso en pocas palabras.

- Actor: Los actores que intervienen en el caso de uso se listan aquí.
- Descripción: Es la descripción detallada del caso de uso.
- Precondición: Condición que debe cumplirse para que el caso de uso pueda darse.
- Postcondición: Estado en el que la aplicación resulta después de aplicar el caso de uso.

### 2.5.3. Casos de uso

Los casos de uso que se presentan a continuación pertenecen al jugador del videojuego el cual ya se lo ha definido en secciones anteriores. Los casos de uso tienen como fin dar a entender cómo se comportará el sistema cuando dado un escenario dado. El videojuego tendrá tres escenarios: principal, secundario, terciario y escenario final.

**Figura 10. Caso de uso general del videojuego.**



**Fuente:** Fausto Sebastián Borja Villacís

**Tabla 9. Caso de uso 1 – Configuración de opciones**

identificador	Caso de uso 1
<b>Nombre</b>	Configuración de opciones
<b>Actor</b>	Jugador
<b>Descripción</b>	El Jugador configura las opciones de audio y video que le permitirán sentirse cómodo en el juego.
<b>Precondición</b>	Ninguna
<b>Post condición</b>	Se inicializa el escenario opciones y el jugador tiene las opciones a elegir.

**Fuente:** Fausto Sebastián Borja Villacís

**Tabla 10. Caso de uso 2 – Escenario cuidado del agua**

identificador	Caso de uso 1
<b>Nombre</b>	Escenario cuidado del agua
<b>Actor</b>	Jugador
<b>Descripción</b>	El Jugador interacciona con flechas de teclado el personaje del escenario
<b>Precondición</b>	Ninguna
<b>Post condición</b>	Se inicializa el escenario cuidado del agua y el jugador interacciona con su personaje.

**Fuente:** Fausto Sebastián Borja Villacís

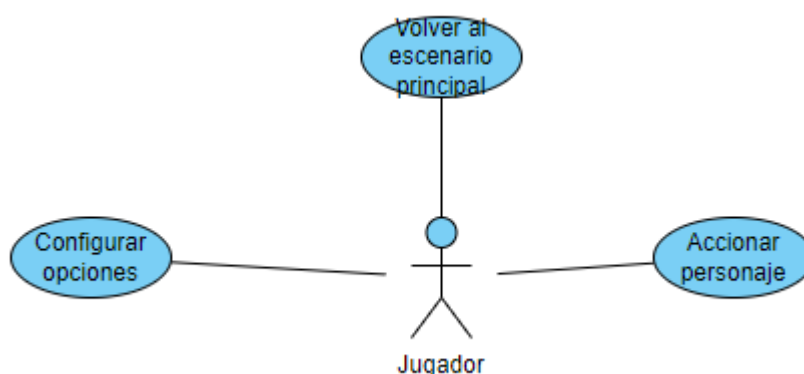
**Tabla 11. Caso de uso 3 – Escenario cuidado de residuos**

identificador	Caso de uso 3
<b>Nombre</b>	Escenario cuidado de residuos
<b>Actor</b>	Jugador
<b>Descripción</b>	El Jugador interacciona con flechas de teclado el personaje del escenario
<b>Precondición</b>	Ninguna
<b>Post condición</b>	Se inicializa el escenario cuidado de residuos y el jugador interacciona con su personaje.

**Fuente:** Fausto Sebastián Borja Villacís

La siguiente figura muestra el caso de uso que corresponde al escenario cuidado del agua. Se muestra una vez el usuario haya cogido tal escenario y en el cual están contenidos los siguientes casos de uso: volver a escenario principal, configurar opciones y accionar personaje.

**Tabla 12. Caso de uso escenario cuidado del agu**



**Fuente:** Fausto Sebastián Borja Villacís

**Tabla 13. Caso de uso 4 – Configurar opciones**

identificador	Caso de uso 4
Nombre	Configurar opciones
Actor	Jugador
Descripción	El Jugador tiene la posibilidad de configurar sus opciones de audio y video.
Precondición	Ninguna
Post condición	Se inicializa el escenario de configuración de opciones.

Fuente: Fausto Sebastián Borja Villacís

**Tabla 14. Caso de uso 5 – Volver al escenario principal**

identificador	Caso de uso 5
Nombre	Volver al escenario principal
Actor	Jugador
Descripción	El Jugador tiene la posibilidad de volver al escenario principal.
Precondición	Tiene que encontrarse en el escenario cuidado del agua.
Post condición	Se inicializa el escenario principal del juego.

Fuente: Fausto Sebastián Borja Villacís

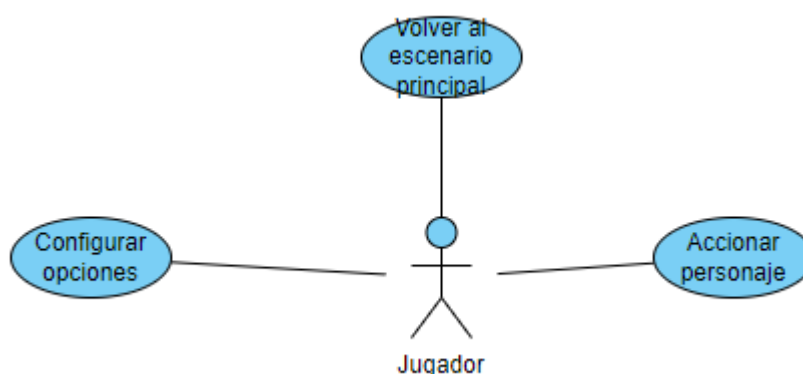
**Tabla 15. Caso de uso 6 – Accionar personaje**

identificador	Caso de uso 6
<b>Nombre</b>	Accionar personaje
<b>Actor</b>	Jugador
<b>Descripción</b>	El Jugador acciona su personaje mediante teclas que se encuentran en el teclado.
<b>Precondición</b>	Tiene que encontrarse en el escenario cuidado del agua.
<b>Post condición</b>	El personaje cambia de posición continua o discretamente.

**Fuente:** Fausto Sebastián Borja Villacís

A continuación, se muestra la figura que muestra el escenario cuidado de residuos con los casos de uso que le corresponden, los cuales son: volver a escenario principal, configurar opciones y accionar personaje.

**Tabla 16. Caso de uso escenario cuidado de residuos.**



**Fuente:** Fausto Sebastián Borja Villacís

**Tabla 17. Caso de uso 7 – Configurar opciones**

identificador	Caso de uso 4
Nombre	Configurar opciones
Actor	Jugador
Descripción	El Jugador tiene la posibilidad de configurar sus opciones de audio y video.
Precondición	Ninguna
Post condición	Se inicializa el escenario de configuración de opciones.

Fuente: Fausto Sebastián Borja Villacís

**Tabla 18. Caso de uso 8 – Volver al escenario principal**

identificador	Caso de uso 5
Nombre	Volver al escenario principal
Actor	Jugador
Descripción	El Jugador tiene la posibilidad de volver al escenario principal.
Precondición	Tiene que encontrarse en el escenario cuidado de residuos.
Post condición	Se inicializa el escenario principal del juego.

Fuente: Fausto Sebastián Borja Villacís

**Tabla 19. Caso de uso 9 – Accionar personaje**

<b>identificador</b>	<b>Caso de uso 6</b>
<b>Nombre</b>	Accionar personaje
<b>Actor</b>	Jugador
<b>Descripción</b>	El Jugador acciona su personaje mediante teclas que se encuentran en el teclado.
<b>Precondición</b>	Tiene que encontrarse en el escenario cuidado de residuos.
<b>Post condición</b>	El personaje cambia de posición continua o discretamente.

**Fuente:** Fausto Sebastián Borja Villacís

## **2.6. Requerimientos del sistema**

Los requerimientos del sistema se clasificarán en funcionales y no funcionales que se requieren para un óptimo funcionamiento. Éstos tienen como fin la definir su nombre, descripción, prioridad, claridad y necesidad.

### **2.6.1. Definición de requerimientos**

Los requerimientos tienen que seguir un formato en específico para representar sus características en este caso, es necesario que contenga un formato en específico. Una tabla que se muestra a continuación contiene un formato que resulta útil para registrar los requisitos del sistema.

**Tabla 20. Ejemplo de formato de requisitos funcionales y no funcionales**

Identificador			
Nombre			
Descripción			
Prioridad	Alta	Media	Baja
Claridad	Alta	Media	Baja
Necesidad	Esencial	Deseable	Opcional

**Fuente:** Fausto Sebastián Borja Villacís

Una tabla de requisitos funcionales o no funcionales contiene los siguientes elementos:

- Identificador: Identificador único del requerimiento
- Nombre: Descripción pequeña del requisito
- Descripción: La razón de ser del requerimiento
- Prioridad: Se explica como la urgencia de su implementación en sus diferentes niveles.
- Claridad: Clasifica al requisito como ambiguo en sus diferentes grados.

### **2.6.2. Requisitos funcionales**

Los requerimientos que se exponen a continuación son aquellos que el videojuego debe cumplir explícitamente.

**Tabla 21. Requisito funcional 1 – Intercambiar escenas**

Identificador	Requisito funcional 1
Nombre	Intercambiar escenas
Descripción	El jugador debe pasar de una escena a otra correctamente.
Prioridad	Alta x Media Baja
Claridad	Alta x Media Baja
Necesidad	Esencial x Deseable Opcional

**Fuente:** Fausto Sebastián Borja Villacís

**Tabla 22. Requisito funcional 2 – Cargar escenas**

Identificador	Requisito funcional 2
Nombre	Cargar escenas
Descripción	El videojuego debe cargar y renderizar correctamente las escenas según el tipo que sea
Prioridad	Alta x Media Baja
Claridad	Alta x Media Baja
Necesidad	Esencial x Deseable Opcional

**Fuente:** Fausto Sebastián Borja Villacís

**Tabla 23. Requisito funcional 3 – Mostrar configuraciones**

Identificador	Requisito funcional 3
Nombre	Mostrar configuraciones
Descripción	El videojuego debe mostrar el menú de configuraciones según el jugador lo seleccione o no.
Prioridad	Alta      Media x      Baja
Claridad	Alta x      Media      Baja
Necesidad	Esencial      Deseable      Opcional x

**Fuente:** Fausto Sebastián Borja Villacís

**Tabla 24. Requisito funcional 4 – Cambiar idioma**

Identificador	Requisito funcional 4
Nombre	Cambiar idioma
Descripción	La configuración de cambio de idioma debe estar disponible.
Prioridad	Alta x      Media      Baja
Claridad	Alta x      Media      Baja
Necesidad	Esencial      Deseable      Opcional x

**Fuente:** Fausto Sebastián Borja Villacís

**Tabla 25. Requisito funcional 5 – Activar o desactivar audio**

Identificador	Requisito funcional 5
Nombre	Activación o desactivación de audio
Descripción	La configuración de cambio de nivel de audio debe estar disponible
Prioridad	Alta      Media x      Baja
Claridad	Alta x      Media      Baja
Necesidad	Esencial      Deseable      Opcional x

**Fuente:** Fausto Sebastián Borja Villacís

**Tabla 26. Requisito funcional 6 – Validación de personaje en escenario**

Identificador	Requisito funcional 6
Nombre	Validación de personaje en escenario
Descripción	El videojuego muestra si el jugador completo la tarea con éxito o no
Prioridad	Alta x      Media      Baja
Claridad	Alta x      Media      Baja
Necesidad	Esencial x      Deseable      Opcional

**Fuente:** Fausto Sebastián Borja Villacís

**Tabla 27. Requisito funcional 7 – Sugerencias en escenario**

Identificador	Requisito funcional 7
Nombre	Sugerencias en escenario
Descripción	El videojuego muestra sugerencias cuando el tiempo de realización de una tarea tarda demasiado
Prioridad	Alta x Media Baja
Claridad	Alta x Media Baja
Necesidad	Esencial x Deseable Opcional

**Fuente:** Fausto Sebastián Borja Villacís

**Tabla 28. Requisito funcional 8 – Mostrar el menú de ganar o perder**

Identificador	Requisito funcional 8
Nombre	Mostrar el menú de ganar o perder
Descripción	El videojuego muestra un menú que da opciones para regresar al escenario principal, configurar opciones o realizar otra vez la tarea
Prioridad	Alta x Media Baja
Claridad	Alta x Media Baja
Necesidad	Esencial x Deseable Opcional

**Fuente:** Fausto Sebastián Borja Villacís

### 3.6.3. Requisitos no funcionales

En el videojuego se extrajeron los siguientes requisitos no funcionales:

**Tabla 29. Requisito no funcional 1 – Eficiencia al procesar múltiples entradas de teclado**

Identificador	Requisito no funcional 1
Nombre	Eficiencia
Descripción	El videojuego debe responder únicamente a entradas las cuales fue programado
Prioridad	Alta x Media Baja
Claridad	Alta x Media Baja
Necesidad	Esencial x Deseable Opcional

**Fuente:** Fausto Sebastián Borja Villacís

**Tabla 30. Requisito no funcional 2 – Fiabilidad en los gráficos que se muestran**

Identificador	Requisito no funcional 2
Nombre	Fiabilidad
Descripción	El videojuego debe mostrar coherencia en sus gráficos en el contexto que se encuentren
Prioridad	Alta x Media Baja
Claridad	Alta x Media Baja
Necesidad	Esencial x Deseable Opcional

**Fuente:** Fausto Sebastián Borja Villacís

**Tabla 31. Requisito no funcional 3 – Disponibilidad de navegabilidad**

Identificador	Requisito no funcional 3
Nombre	Disponibilidad
Descripción	El videojuego debe mostrar navegabilidad para cambiar de escenas en todo momento
Prioridad	Alta x Media Baja
Claridad	Alta x Media Baja
Necesidad	Esencial x Deseable Opcional

**Fuente:** Fausto Sebastián Borja Villacís

**Tabla 32. Requisito no funcional 4 – Mantenibilidad de configuraciones**

Identificador	Requisito no funcional 4
Nombre	Mantenibilidad
Descripción	El videojuego debe reflejar las configuraciones que el usuario haga en el menú de configuraciones
Prioridad	Alta x Media Baja
Claridad	Alta x Media Baja
Necesidad	Esencial x Deseable Opcional

**Fuente:** Fausto Sebastián Borja Villacís

## **CAPÍTULO III**

### **METODOLOGÍA**

#### **3. Metodología**

Metodología es el conjunto de métodos y técnicas que se utilizan para realizar una investigación o estudio (Sampieri et al., 2014). Es el proceso seguido para recopilar y analizar datos y producir resultados que respondan a las preguntas de investigación (Sampieri et al., 2014). El método debe ser riguroso y sistemático y permitir que otros investigadores repliquen los resultados. Es importante seguir el método correcto para garantizar la validez y fiabilidad de los resultados obtenidos (Sampieri et al., 2014).

#### **3.1. Enfoque de la Investigación**

##### **3.1.1. Enfoque de investigación cualitativa**

La metodología de la investigación cualitativa se basa en la idea de que el conocimiento se construye socialmente a través de la interacción entre el investigador y los participantes (Sampieri et al., 2014). Por lo tanto, este enfoque se enfoca en comprender las experiencias, percepciones y comportamientos de los participantes a través de datos no numéricos y contextualizados en su entorno natural.

Para recopilar datos, se utilizan técnicas como entrevistas, observación participante y análisis de documentos. Estos datos se analizan mediante técnicas de análisis de contenido, análisis temático y análisis de narrativas, entre otros. En general, la metodología de la investigación cualitativa se enfoca en obtener información detallada y rica sobre un fenómeno, más que en generalizar los resultados a una población más amplia (Sampieri et al., 2014).

Es importante mencionar que la investigación cualitativa es un enfoque flexible y adaptable. A menudo, los investigadores deben ajustar su metodología a medida

que avanzan en la investigación, para adaptarse a los nuevos hallazgos o descubrimientos. Por lo tanto, la metodología de la investigación cualitativa puede ser una tarea desafiante, pero también es una oportunidad para que los investigadores se involucren profundamente en su tema de estudio y desarrollen un conocimiento detallado y comprensivo sobre el mismo (Sampieri et al., 2014).

En conclusión, la metodología de la investigación cualitativa es crucial para una investigación de este tipo porque se necesita de recolección de datos o información no cuantitativos que permitan determinar la viabilidad y efectividad que supone el desarrollo de la aplicación-videojuego que enseña sobre el cuidado del medio ambiente aplicado a una determinada población.

## **3.2. Diseño de investigación**

Se pueden definir como investigaciones realizadas sin manipular deliberadamente las variables, es decir, investigaciones en las que no se modifican deliberadamente las variables independientes (Sampieri, 2014). El propósito de la investigación no experimental es observar los fenómenos que ocurren en el medio natural y luego analizarlos. En este sentido, el estudio se considera no experimental porque no observó cómo el proceso en curso llegó a un diseño que pudiera dar una solución al problema actual sin cambiar las variables independientes (Sampieri, 2014).

### **3.2.1. Diseño no experimental**

La investigación se ha diseñado de forma que cumple con los ítems de la siguiente lista:

- **Objetivo de la investigación:** El objetivo principal es comprender las necesidades y perspectivas de la docente sobre la enseñanza del cuidado medioambiental mediante un videojuego.
- **Participante:** Selecciona a una docente que tenga experiencia en la enseñanza relacionada con el cuidado medioambiental o tenga interés en utilizar nuevas tecnologías en su metodología educativa.

- Metodología: Utiliza una entrevista dirigida como método de recolección de datos. Una entrevista dirigida implica formular una serie de preguntas específicas para obtener información detallada sobre las experiencias y opiniones de la docente.
- Preguntas de la entrevista:
  - a. Experiencia y enfoque actual de la docente en la enseñanza del cuidado medioambiental.
  - b. Percepción de la importancia de los videojuegos como herramienta educativa.
  - c. Opiniones sobre la incorporación de un videojuego para enseñar el cuidado medioambiental.
  - d. Necesidades específicas y recomendaciones para el diseño y contenido del videojuego.
- Procedimiento: Establece una cita con la docente y realiza la entrevista en un entorno cómodo y propicio para la comunicación.
- Análisis de datos: Transcribe y analiza las respuestas de la entrevista de manera sistemática. Busca patrones, temas recurrentes y puntos clave relevantes para los objetivos de tu investigación.
- Resultados y conclusiones: Resume y presenta los hallazgos más relevantes de la entrevista. Utiliza citas o ejemplos concretos para respaldar tus conclusiones.

### **3.3. Nivel de investigación**

#### **3.3.1. Investigación exploratoria**

La investigación exploratoria es un tipo de investigación que se realiza cuando el tema o fenómeno objeto de estudio es poco conocido o no ha sido estudiado en profundidad (Sampieri, 2014). Su objetivo principal es explorar y familiarizarse con el tema, descubrir nuevas ideas, generar hipótesis iniciales y obtener una comprensión general del fenómeno dado (Sampieri, 2014). En la investigación exploratoria, los métodos suelen ser flexibles y abiertos, lo que permite recopilar datos cualitativos y

cuantitativos de una variedad de fuentes, como entrevistas, observaciones, revisiones de literatura, encuestas o análisis de datos existentes (Sampieri, 2014). La investigación exploratoria también puede implicar la realización de estudios piloto o pequeños estudios preliminares antes de realizar estudios más amplios.

### **3.4. Tipo de investigación**

#### **3.4.1. Investigación bibliográfica**

La investigación bibliográfica es una técnica de investigación que consiste en la búsqueda, selección, análisis, interpretación y sistematización de información contenida en fuentes bibliográficas, es decir, en libros, revistas, tesis, informes, artículos, entre otros documentos escritos (García et al., 2001). Esta técnica de investigación es muy utilizada en el ámbito académico y científico para recopilar información previa a la realización de un estudio o para profundizar en una temática específica (García et al., 2001).

Las características principales de la investigación bibliográfica son (García et al., 2001):

1. Utiliza fuentes secundarias: la investigación bibliográfica se basa en el uso de fuentes secundarias, es decir, en información que ha sido recopilada previamente por otros autores.
2. Requiere selección y evaluación de fuentes: para realizar una investigación bibliográfica es necesario seleccionar y evaluar cuidadosamente las fuentes bibliográficas a utilizar, con el fin de garantizar su calidad y pertinencia.
3. Se realiza sin contacto directo con el objeto de estudio: a diferencia de otras técnicas de investigación, en la investigación bibliográfica no es necesario el contacto directo con el objeto de estudio.
4. Permite la exploración de temas complejos: esta técnica permite explorar temas complejos en profundidad, a través de la revisión exhaustiva de la literatura existente.

5. Se pueden obtener datos de diversas fuentes: gracias a la amplia variedad de fuentes bibliográficas disponibles, se pueden obtener datos de diversas fuentes para complementar la investigación.

La investigación bibliográfica es una técnica de investigación que consiste en la búsqueda, selección y análisis de información contenida en fuentes bibliográficas, con el fin de profundizar en una temática específica o recopilar información previa a la realización de un estudio.

### **3.5. Descripción de la propuesta**

Esta investigación tiene como objetivo principal comprender las necesidades y perspectivas de los docentes con respecto a la enseñanza del cuidado medioambiental mediante un videojuego educativo. A través de entrevistas dirigidas a una docente con experiencia en la materia, se explorarán sus opiniones sobre la importancia de los videojuegos como herramienta educativa, así como sus recomendaciones y necesidades específicas para el diseño y contenido del videojuego. Los hallazgos de esta investigación servirán como base para el desarrollo de un videojuego educativo efectivo y adaptado a las necesidades de los docentes y los estudiantes

### **3.6. Población**

Docente de la Unidad Educativa Franciscana San Andrés que es docente de dicha institución y que mostró interés en un videojuego para la enseñanza del cuidado medio ambiental. Dicha docente tiene experiencia en enseñanza de prácticas de cuidado medio ambiental mediante la asignatura ciencias naturales.

### 3.7. Técnica e instrumentos

En la recolección de datos se hará uso de la herramienta llamada entrevista, la cual estará formulada por cinco preguntas.

1. ¿Por qué cree que los videojuegos podrían ser una herramienta efectiva para enseñar a los niños sobre el cuidado del medio ambiente?
2. ¿Cree que los videojuegos podrían complementar o reemplazar la enseñanza tradicional en el aula en lo que respecta al cuidado medioambiental? ¿Por qué?
3. ¿Qué tipo de videojuegos cree que serían los más efectivos para enseñar sobre el cuidado medioambiental?
4. ¿Cree que los videojuegos podrían motivar a los estudiantes a involucrarse más en la protección del medio ambiente? ¿Por qué?
5. ¿Cuáles serían las principales preocupaciones o desafíos que tendríamos que enfrentar si decidiéramos incorporar videojuegos en el currículo de enseñanza de cuidado medioambiental en la escuela?

Estas preguntas fueron respondidas por Ruth Céspedes Docente de primaria de la UEFSA, de forma oral de la siguiente manera:

1. ¿Por qué cree que los videojuegos podrían ser una herramienta efectiva para enseñar a los niños sobre el cuidado del medio ambiente?

“Personalmente, creo que los videojuegos podrían ser una herramienta efectiva para enseñar a los niños sobre el cuidado del medio ambiente porque la mayoría de los niños disfrutan jugando videojuegos y los videojuegos tienen la capacidad de hacer que los niños se sientan involucrados en el aprendizaje. Además, los videojuegos pueden ofrecer un enfoque más práctico y atractivo para la enseñanza de conceptos abstractos como la sostenibilidad y el cuidado medioambiental.” (Céspedes, 2023)

2. ¿Cree que los videojuegos podrían complementar o reemplazar la enseñanza tradicional en el aula en lo que respecta al cuidado medioambiental? ¿Por qué?

“En mi opinión, los videojuegos podrían complementar la enseñanza tradicional en el aula en lo que respecta al cuidado medioambiental. Aunque la enseñanza tradicional sigue siendo importante, los videojuegos pueden ofrecer un enfoque más interactivo y atractivo que puede ayudar a los estudiantes a involucrarse más en el aprendizaje. Además, los videojuegos pueden ser una herramienta efectiva para llegar a los estudiantes que pueden tener dificultades para aprender a través de métodos de enseñanza tradicionales.” (Céspedes, 2023)

3. ¿Qué tipo de videojuegos cree que serían los más efectivos para enseñar sobre el cuidado medioambiental?

“Creo que los videojuegos que se centran en la simulación y la toma de decisiones pueden ser los más efectivos para enseñar sobre el cuidado medioambiental. Por ejemplo, un videojuego que simule la creación y gestión de una granja sostenible o un parque natural puede ayudar a los estudiantes a comprender los impactos ambientales de las decisiones que toman. Los videojuegos también podrían incorporar desafíos y problemas relacionados con la conservación y el cuidado del medio ambiente para ayudar a los estudiantes a desarrollar habilidades y conocimientos prácticos.” (Céspedes, 2023)

4. ¿Cree que los videojuegos podrían motivar a los estudiantes a involucrarse más en la protección del medio ambiente? ¿Por qué?

“Definitivamente creo que los videojuegos podrían motivar a los estudiantes a involucrarse más en la protección del medio ambiente. Los videojuegos pueden ser una forma divertida y atractiva de aprender y pueden ayudar a

los estudiantes a sentirse más empoderados para tomar medidas para proteger el medio ambiente. Además, los videojuegos pueden ofrecer una forma de medir y rastrear el progreso de los estudiantes en la comprensión de los conceptos ambientales clave, lo que puede motivar a los estudiantes a seguir aprendiendo y mejorando.” (Céspedes, 2023)

5. ¿Cuáles serían las principales preocupaciones o desafíos que tendríamos que enfrentar si decidiéramos incorporar videojuegos en el currículo de enseñanza de cuidado medioambiental en la escuela?

“Uno de los principales desafíos que enfrentaríamos si decidiéramos incorporar videojuegos en el currículo de enseñanza de cuidado medioambiental en la escuela sería garantizar que los videojuegos sean educativos y precisos desde el punto de vista ambiental. Además, tendríamos que asegurarnos de que los videojuegos sean accesibles y atractivos para una amplia gama de estudiantes, incluidos aquellos con discapacidades o barreras de aprendizaje. También tendríamos que tener en cuenta los costos asociados con la incorporación de videojuegos en el currículo escolar y garantizar que los recursos estén disponibles para implementar esta tecnología de manera efectiva.” (Céspedes, 2023)

### **3.8. Metodología de desarrollo del videojuego**

La metodología de desarrollo de software se refiere a un conjunto de técnicas y prácticas que se utilizan para planificar, diseñar, construir, probar y mantener software de alta calidad. La metodología es un proceso sistemático que guía el desarrollo de software desde el inicio hasta la entrega del producto final. Existen varias metodologías de desarrollo de software, entre ellas se encuentran las metodologías tradicionales y las metodologías ágiles (Blankenship et al., 2011).

Las metodologías tradicionales, como el modelo en cascada, se caracterizan por un enfoque secuencial y lineal, en el que cada etapa del proceso de desarrollo se

lleva a cabo en orden. El modelo en cascada comienza con la definición de requisitos, seguido por el diseño, la implementación, la verificación y la validación, y finalmente, la operación y el mantenimiento (Blankenship et al., 2011).

Por otro lado, las metodologías ágiles, como Scrum o Kanban, se basan en la iteración y la colaboración. Los equipos trabajan en sprints cortos para entregar funcionalidades en pequeñas iteraciones. Las metodologías ágiles también se centran en la comunicación frecuente entre el equipo de desarrollo y el cliente para garantizar que se satisfagan las necesidades del usuario y que el producto final sea de alta calidad (Blankenship et al., 2011).

### **3.8.1. Metodología ágil SCRUM**

SCRUM es un marco de trabajo ágil utilizado en la gestión de proyectos de software. Fue desarrollado en la década de 1990 por Ken Schwaber y Jeff Sutherland, y ha sido adoptado por numerosas organizaciones en todo el mundo. SCRUM se basa en la entrega continua de software a través de iteraciones cortas y enfocadas, conocidas como sprints (Rubin, 2012).

SCRUM se centra en la colaboración, la transparencia y la adaptabilidad. El equipo de desarrollo trabaja en estrecha colaboración con el cliente para entender sus necesidades y objetivos, y para asegurarse de que el software entregado cumpla con sus expectativas. La transparencia es fundamental en SCRUM, y se logra a través de reuniones y artefactos que permiten a todos los miembros del equipo conocer el estado del proyecto en todo momento. La adaptabilidad es una característica clave de SCRUM, ya que el equipo de desarrollo debe ser capaz de ajustar su enfoque en función de los cambios en los requisitos del cliente o en las condiciones del mercado (Rubin, 2012).

Los principales roles en SCRUM son el Scrum Master, el Product Owner y el Equipo de Desarrollo. El Scrum Master es el facilitador del proceso y se asegura de que el equipo de desarrollo siga el marco de trabajo de SCRUM de manera efectiva. El Product Owner es el representante del cliente y es responsable de definir el backlog

de producto, que es una lista ordenada de características y funcionalidades que el software debe tener. El Equipo de Desarrollo es responsable de desarrollar y entregar el software (Rubin, 2012).

SCRUM se divide en sprints, que son iteraciones de trabajo de una duración fija, generalmente de una a cuatro semanas. Al comienzo de cada sprint, el equipo de desarrollo se reúne en una reunión de planificación del sprint para definir los objetivos y el alcance del sprint. Durante el sprint, el equipo de desarrollo se reúne diariamente en una reunión de seguimiento diario para informar sobre el progreso y coordinar el trabajo. Al final del sprint, el equipo de desarrollo se reúne en una reunión de revisión del sprint para demostrar el software entregado y recibir comentarios del cliente. Finalmente, el equipo de desarrollo se reúne en una reunión de retrospectiva del sprint para analizar lo que funcionó bien y lo que se puede mejorar en el siguiente sprint (Rubin, 2012).

SCRUM utiliza una serie de artefactos para garantizar la transparencia y la comunicación efectiva en el proceso de desarrollo (Rubin, 2012). El backlog de producto es una lista ordenada de características y funcionalidades que el software debe tener. El backlog de sprint es una lista de tareas específicas que el equipo de desarrollo se compromete a completar durante el sprint. Los gráficos de quemado son herramientas visuales que muestran el progreso del equipo de desarrollo durante el sprint y el proyecto en su conjunto.

SCRUM tiene varias ventajas en comparación con otros enfoques de gestión de proyectos. En primer lugar, SCRUM es altamente adaptable y puede manejar cambios en los requisitos del cliente o en las condiciones del mercado de manera efectiva. En segundo lugar, SCRUM es altamente transparente y fomenta la comunicación efectiva entre los miembros del equipo. En tercer lugar, SCRUM promueve una cultura de colaboración y empoderamiento, lo que puede mejorar la motivación y la satisfacción del equipo de desarrollo (Rubin, 2012).

Sin embargo, SCRUM también tiene algunas desventajas potenciales (Rubin, 2012):

1. Requiere un alto grado de compromiso del equipo: SCRUM requiere que el equipo de desarrollo se comprometa plenamente con el proceso y que esté dispuesto a trabajar juntos de manera intensiva durante cada sprint. Si un miembro del equipo no está comprometido o no puede cumplir con sus responsabilidades, puede afectar negativamente el progreso del proyecto.
2. Requiere una buena planificación: El éxito de SCRUM depende en gran medida de una buena planificación, que incluye definir los objetivos y el alcance de cada sprint y del proyecto en su conjunto. Si la planificación no es sólida, puede conducir a problemas durante el desarrollo y a retrasos en el cronograma del proyecto.
3. Puede ser difícil de implementar en grandes equipos: SCRUM funciona mejor en equipos pequeños y altamente colaborativos. En equipos grandes, puede ser difícil coordinar y comunicarse de manera efectiva, lo que puede afectar la calidad del software entregado.
4. Puede ser desafiante para equipos inexperimentados: SCRUM requiere una comprensión sólida del proceso y una buena gestión del tiempo. Los equipos que son nuevos en SCRUM pueden enfrentar una curva de aprendizaje empinada, lo que puede afectar negativamente la calidad del software entregado y el cronograma del proyecto.
5. Puede haber dificultades en la gestión de riesgos: SCRUM se centra en la entrega continua de software, lo que significa que se entregan pequeñas partes del producto en cada sprint. Esto puede dificultar la gestión de riesgos en el proyecto, ya que los riesgos pueden no ser evidentes hasta que se haya entregado una parte del producto.

### **3.8.1.1. Ciclo de vida de SCRUM**

El ciclo de vida de SCRUM se divide en varios procesos iterativos e incrementales, que se conocen como sprints, y consta de los siguientes elementos (Rubin, 2012):

1. Planificación del sprint: El equipo de SCRUM se reúne para planificar el trabajo que se realizará durante el próximo sprint. Se establecen los objetivos y las metas para el sprint, se seleccionan las historias de usuario que se deben completar y se asignan las tareas a los miembros del equipo.
2. Sprint: El equipo trabaja en la lista de tareas y entregables definidos durante la planificación del sprint. El sprint tiene una duración fija, generalmente de dos a cuatro semanas, y al final se debe entregar un incremento del producto potencialmente entregable.
3. Reuniones diarias: Durante el sprint, el equipo de SCRUM se reúne diariamente para revisar el progreso del trabajo y hacer ajustes necesarios para alcanzar los objetivos del sprint. Estas reuniones son llamadas "stand-ups" y duran de 10 a 15 minutos.
4. Revisión del sprint: Al final del sprint, el equipo de SCRUM presenta el incremento del producto potencialmente entregable al dueño del producto y a otros interesados. En esta reunión se analizan los logros y se identifican las áreas para mejorar.
5. Retrospectiva del sprint: El equipo de SCRUM se reúne para revisar el sprint, evaluar el proceso y definir mejoras para el próximo sprint. En esta reunión se identifican los puntos fuertes y débiles del sprint, se discuten las oportunidades de mejora y se toman decisiones para el próximo sprint.

En la siguiente figura se ilustra de manera didáctica el ciclo de vida de SCRUM.

**Figura 11. Ciclo de vida SCRUM.**



Fuente: (Canive, 2020)

## **CAPÍTULO IV**

### **DISEÑO Y DESARROLLO DE LA APLICACIÓN**

#### **4. Diseño y desarrollo de la aplicación**

##### **4.1. Recolección de audio y animaciones**

La aplicación se compondrá de imágenes, textos y animaciones (conjunto de imágenes forman una animación). La elaboración de arte desde base puede suponer un gran gasto de tiempo, energía y creatividad, por lo que se ha dado como elección la decisión de recolectar recursos 2D de varias páginas web que tienen varios elementos que compondrán la aplicación. Estos son los Sprites, Sprite Sheets, imágenes de fondo y audio.

##### **4.1.1. Sprites**

En Godot los Sprites son objetos que se utilizan para representar imágenes o gráficos 2D en un juego. Un Sprite es un tipo de nodo (Node) que se utiliza para agregar una imagen o textura a un escenario. En un juego 2D, un Sprite puede representar personajes, objetos, fondos, efectos visuales, entre otros elementos. Los Sprites en Godot se pueden crear de diferentes maneras, como a través de la carga de una imagen desde un archivo, la creación de una textura dinámica o la generación procedural de gráficos mediante código (Engine, s. f.). Los Sprites tienen una serie de propiedades que se pueden ajustar, como la posición, la escala, la rotación, la opacidad, la textura y la animación. También es posible agregar efectos visuales a los Sprites, como sombras, reflejos y desenfocados, mediante el uso de shaders. Los Sprites en Godot son objetos que se utilizan para renderizar gráficos 2D en un juego, y son una parte importante de la creación de la estética visual y la jugabilidad de los juegos 2D en esta plataforma.

#### 4.1.2. Muestra de Sprites de la aplicación

La muestra de Sprites empleados van desde imágenes de fondo, así como los personajes y objetos que se usarán como mini mundo, todos ellos tomados de OpenGameArt (OpenGameArt.org, n.d.) y de FreePik (Freepik, s. f.):

**Figura 12. Tortuga animada**



Fuente: (OpenGameArt.org, s.f.)

**Figura 13. Fondo marino**



Fuente: (OpenGameArt.org, s.f.)

## Figura 14. Variedad de objetos



**Fuente:** (*Sprites Pack 16px\*16px Over 150 Sprites! By RRFreelance - PixelBurner, s.f.*)

### 4.1.3. Sprite Sheets

En Godot, los spritesheets son imágenes que contienen múltiples cuadros o fotogramas de animación, que se utilizan para crear animaciones en 2D (Engine, s. f.). En lugar de utilizar múltiples imágenes para cada cuadro de animación, los spritesheets permiten agrupar todos los cuadros en una sola imagen, lo que facilita la gestión y optimización de recursos. En Godot, para utilizar un spritesheet, se debe crear una nueva animación 2D y cargar la imagen del spritesheet. A continuación, se deben definir los cuadros de la animación, especificando las coordenadas de los cuadros en la imagen del spritesheet. Esto se hace mediante la definición de "rectángulos de recorte" que indican la posición y tamaño de cada cuadro en la imagen. Una vez que se han definido los cuadros, se pueden crear secuencias de animación utilizando los cuadros del spritesheet. Esto se puede hacer mediante la configuración de los parámetros de la animación, como la velocidad, la dirección y la repetición. Los spritesheets son una forma eficiente y organizada de crear

animaciones en 2D en Godot. Permiten agrupar múltiples cuadros en una sola imagen y facilitan la creación y gestión de animaciones.

#### 4.1.4. Muestra de Sprite Sheets de la aplicación

Los spritesheet al ser una imagen que contiene un personaje con su animación, en Godot se lo procesa de forma que enlazado al un cuerpo con colisiones este pueda interactuar, la siguiente figura muestra una tortuga que una vez procesado su movimiento simulará la acción de andar o caminar.

**Figura 15. Tortuga caminando**



Fuente: *(OpenGameArt.org, n.d.)*

**Figura 16. Día y noche**



Fuente: *(OpenGameArt.org, n.d.)*

#### 4.1.5. Audio

El audio de la aplicación se corresponde con audio de fondo y de animaciones de cada personaje o evento que suceda. Todo este material será recolectado de

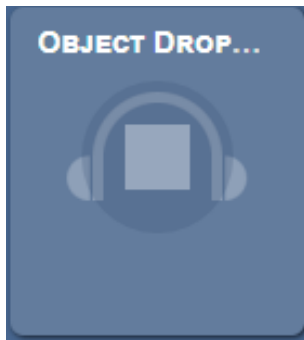
(OpenGameArt.org, s.f.) el cual tiene varios audios que son de licencia libre con créditos al autor de forma opcional. Las siguientes figuras ejemplifican las muestras de audio que se usarán en la aplicación:

**Figura 17. Sonido de río y océano con efectos de agua recorriendo**



**Fuente:** (OpenGameArt.org, n.d.)

**Figura 18. Sonido de caída de objetos en agua**



**Fuente:** (OpenGameArt.org, n.d.)

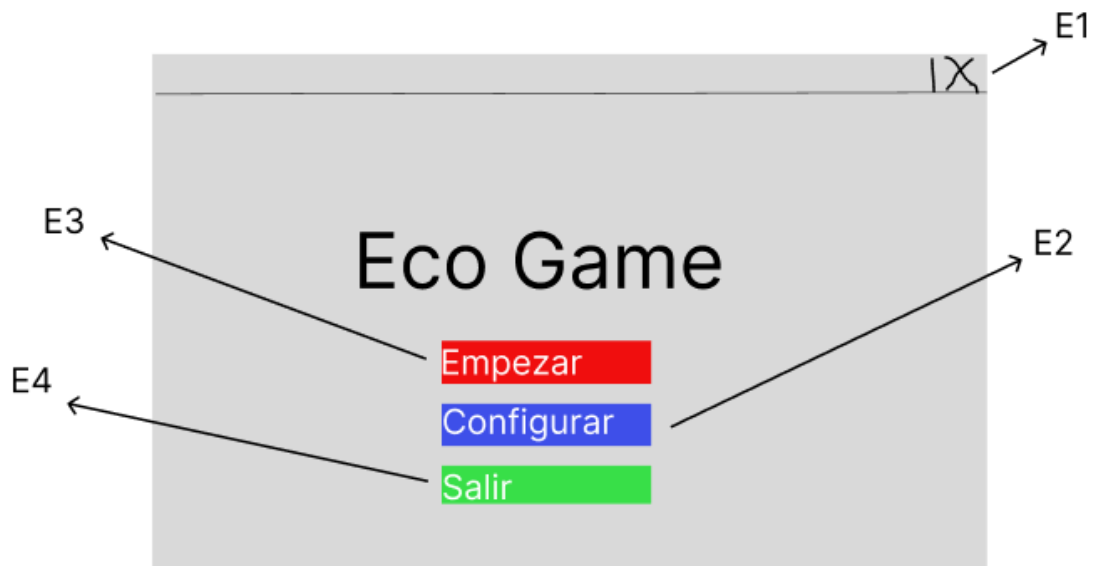
## 4.2. Diseño de la interfaz de usuario

El videojuego al encontrarse en una etapa temprana, interfaces definidas y completas aún no son factibles. Por tanto, bosquejos de este son más asequibles en la etapa que se encuentra actualmente el proyecto. Encontraremos las siguientes escenas:

1. Escena principal
2. Escena cuidado del agua
3. Escena cuidado de residuos

#### 4.2.1. Escena principal

Figura 19. Escena principal con sus cuatro elementos



**Fuente:** Fausto Sebastián Borja Villacís

Los elementos que componen esta escena se describen en la siguiente tabla:

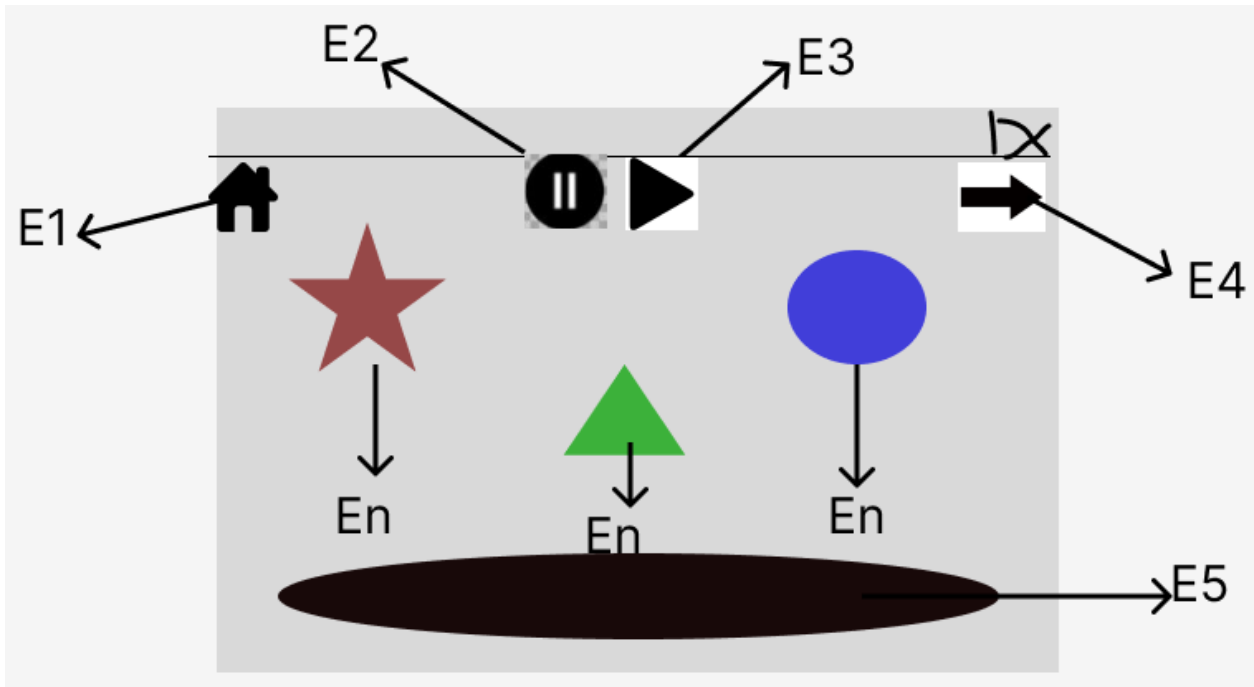
**Tabla 33. Elementos de la escena principal**

Elemento	Representación
E1	Es la salida opcional de la aplicación, nativa de Windows
E2	Al pulsar el botón Configurar, se muestra una pequeña región que permite configurar volumen y brillo de la pantalla
E3	Al pulsar el botón Empezar, se inicia la escena 1
E4	Es la salida opcional proporcionada por la GUI desarrollada en Godot

Fuente: Fausto Sebastián Borja Villacís

#### 4.2.2 Escena cuidado del agua

**Figura 20. Escena cuidado del agua con sus elementos**



Fuente: Fausto Sebastián Borja Villacís

Los elementos que componen esta escena se describen en la siguiente tabla:

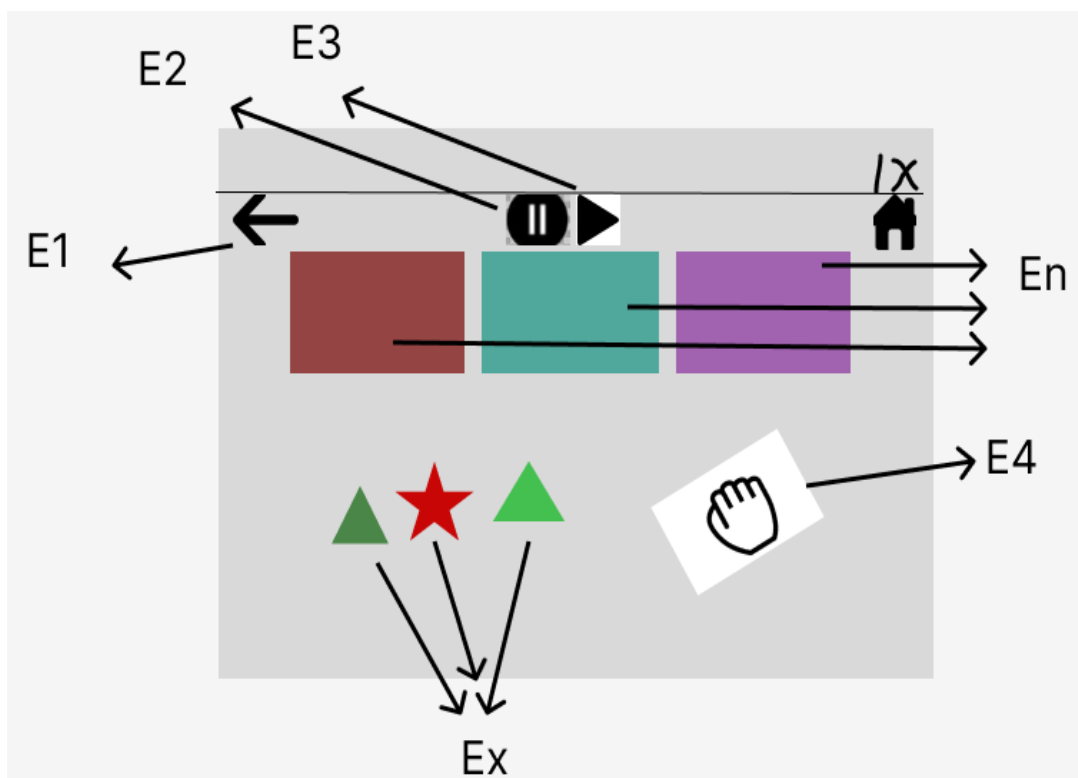
**Tabla 34. Elementos de la escena cuidado del agua**

Elemento	Representación
E1	Es la salida opcional a la escena principal
E2	Es la pausa del juego
E3	Se reanuda el juego después de una pausa
E4	Opción para ir a la escena cuidado de residuos
E5	Área que el jugador debe evitar que los elementos 'En' colisionen

Fuente: Fausto Sebastián Borja Villacís

### 4.2.3. Escena cuidado de residuos

**Figura 21. Escena cuidado de los residuos con sus elementos**



Fuente: Fausto Sebastián Borja Villacís

Los elementos que componen esta escena se describen en la siguiente tabla:

**Tabla 35. Elementos de la escena cuidado del agua**

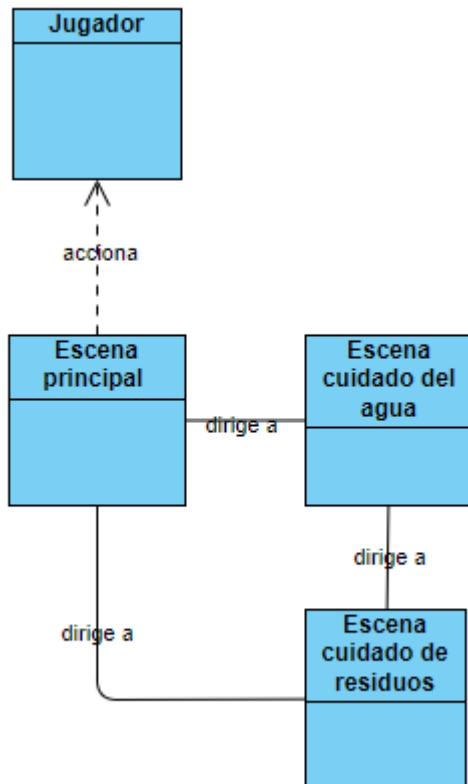
<b>Elemento</b>	<b>Representación</b>
<b>E1</b>	Es la salida opcional a la escena cuidado del agua
<b>E2</b>	Es la pausa del juego
<b>E3</b>	Se reanuda el juego después de una pausa
<b>E4</b>	Cursor de mouse que incita el arrastre de objetos
<b>Ex</b>	Objetos que serán arrastrados a los tachos 'En' según el tipo que sea

**Fuente:** Fausto Sebastián Borja Villacís

### **4.3. Diseño de la aplicación en UML**

El diseño de la aplicación fue diseñado mediante un diagrama de clases. A grandes rasgos no muestra atributos ni métodos puesto que estos están sujetos a cambios cuando el desarrollo de este esté en ejecución. Se cuenta con cuatro clases principales que ilustran los componentes generales que la aplicación estará usando. Cabe recordar que la clase jugador se implementa a través de los personajes que se mostrarán en el juego. La siguiente figura muestra el diagrama de clases propuesto y que se utilizará como base para desarrollar el videojuego.

**Figura 22. Escenas y jugador como clases**



**Fuente:** Fausto Sebastián Borja Villacís

## 4.4. Desarrollo de la aplicación

### 4.4.1. Análisis

Las aplicaciones se perfilan para comprender requerimientos solicitados por el cliente, resultando en: organización, clasificación, priorización y validación. Hay varios puntos de vista en el análisis donde puede basarse en alguna fuente de datos o sumidero, de modo que, al utilizar diferentes métodos de análisis de requisitos, es de vital importancia un modelo-ejemplo del proceso, siendo éste de forma visual.

El propósito de este documento es resumir todos los requisitos del videojuego. Así tendrán que hacerlo los dueños de restaurantes, así el docente de le UEFSA Ruth

Céspedes debe coordinar reuniones con el desarrollador para que dé a entender la finalidad del videojuego.

#### **4.4.2. Historias de usuario**

El videojuego tiene por finalidad enseñar acerca del medio ambiente a niños escolares de 5 a 6 años de la unidad educativa UEFA, todas las necesidades del docente serán capturadas a través de las historias de los usuarios, al igual que el método Scrum. Para ello se realizó una reunión con la docente de la unidad educativa, y la docente comentó que necesita lo siguiente:

1. “En el videojuego necesito que haya un menú principal y dos escenas extras” (Céspedes, 2023)
2. “Necesito que haya una escena la cual enseñe sobre clasifica sobre lo que es basura de la que no es” (Céspedes, 2023)
3. “Necesito que haya otra escena en la que se clasifique basura que sea vidrio, papel y plástico” (Céspedes, 2023)
4. “Necesito que se explique en varios videos como se juega cada escena” (Céspedes, 2023)
5. “Necesito un menú de configuraciones que permita cambiar la configuración de pantalla y de sonido” (Céspedes, 2023)

#### **4.4.3. Product Backlog**

Product Backlog enumera todas las funciones que necesita el cliente mediante el historial de usuario, se detallan los siguientes campos que la tabla necesita para ser nombrada como Product Backlog:

- Número: número de identificación de/los requerimientos.
- Descripción: contiene una breve descripción del requerimiento.
- Story Points: Pequeños cálculos que te permiten dedicar esfuerzo y tiempo al trabajo de desarrollo del videojuego es de 1 a 5, siendo 1 poco esfuerzo y 5 gran esfuerzo.

- Rol: El usuario que utilizará el requerimiento.
- Prioridad: La prioridad con la que se procesan los reclamos clasificados en una escala de 1 a 5, siendo 1 de baja prioridad y 5 de alta prioridad.
- Sprint: El número en el que el requerimiento del cliente se va a satisfacer.
- Tiempo: Tiempo estimado para desarrollar la característica (en horas).
- Criterios de aceptación: Pequeños detalles a implementar, para que el mismo sea aceptado como íntegro y funcional.

#### 4.4.4. Especificación de roles

La siguiente tabla detallará la semilla de trabajo que siguen todos los principios del método Scrum.

**Tabla 36. Especificación de roles**

Rol	Responsable
Product Owner	<b>Fausto Borja</b>
Scrum Master	<b>Fausto Borja</b>
Equipo de desarrollo	<b>Fausto Borja</b>
Cliente	<b>Ruth Céspedes</b>

**Fuente:** Fausto Sebastián Borja Villacís

#### 4.4.5. Sprint 1

##### 4.4.5.1. Planificación del Sprint

La planificación del Sprint 1 hará toda la creación de la interfaz de usuario, y los videos que contienen material para utilizar o jugar el videojuego.

#### 4.4.5.2. Sprint Backlog

**Tabla 37. Sprint Backlog**

Número	Descripción	Punto de historia	Rol	Prioridad	Sprint	Tiempo	Criterio de aceptación
1	En el menú principal debe haber opciones de jugar, cómo jugar, configuraciones y salir de la aplicación	1	Jugador	5	1	15	Los componentes gráficos son intuitivos y fácil de visualizar
2	En la escena de clasificación de basura y no basura debe haber opciones de pausar, continuar, cambiar de escenas y cómo jugar la respectiva escena	1	Jugador	5	1	18	Los componentes gráficos son intuitivos y fácil de visualizar
3	En la escena de clasificación de basura y no basura debe haber opciones de pausar, continuar, cambiar de escenas y cómo jugar la	1	Jugador	5	1	18	Los componentes gráficos son intuitivos y fácil de visualizar

	respectiva escena						
4	Creación de videos para escenas principal, uno y dos	4	Jugador	5	1	15	Los vídeos generan buena atención visual, son llamativos y explicativos

Fuente: Fausto Sebastián Borja Villacís

#### 4.4.5.3. Duración del Sprint

El tiempo que tardará en desarrollarse este Sprint es de 66 horas, los cuales se emplearán de lunes a viernes con 2 horas de carga diaria.

#### 4.4.5.4. Revisión del Sprint

Después de completar el desarrollo del Sprint 1, hubo una reunión con el docente donde se presentaron las siguientes características: interfaces gráficas de escenas uno, dos y vídeos de cada escena respectiva. Después de la revisión, el cliente está satisfecho con todas estas características.

#### 4.4.6. Sprint 2

Planificación del Sprint

La planificación del Sprint 2 hará toda la funcionalidad de las interfaces de usuario, es decir el juego en sí funcionando e íntegro.

#### 4.4.6.1.Sprint Backlog

**Tabla 38. Sprint Backlog**

Número	Descripción	Punto de historia	Rol	Prioridad	Sprint	Tiempo	Criterio de aceptación
5	En el menú principal debe haber funcionalidad de cada una de sus opciones	5	Jugador	4	2	10	El menú de configuración permite expandir o comprimir la pantalla y activar/desactivar el sonido general
6	En la escena uno el personaje es animado y puede recolectar desechos y criaturas marinas	2	Jugador	5	2	30	El personaje principal se mueve e interacciona de forma positiva o negativa con cada tipo de objeto que cae al suelo marino
7	En la escena dos el usuario puede clasificar desechos	3	Jugador	5	2	42	El usuario puede arrastrar los objetos y clasificarlos según su tipo.

	como plástico, vidrio y papel						Admite o no interacciona según sea el caso
--	-------------------------------	--	--	--	--	--	--

Fuente: Fausto Sebastián Borja Villacís

#### 4.4.6.2. Duración del Sprint

El tiempo que tardará en desarrollarse este Sprint es de 82 horas, los cuales se emplearán de lunes a viernes con 6 horas de carga diaria.

#### 4.4.6.3. Revisión del Sprint

Después de completar el desarrollo del Sprint 2, hubo una reunión con el docente donde se presentaron las siguientes características: escenas uno y dos funcionando, menú principal funcional. Después de la revisión, el cliente está satisfecho con todas estas características.

#### 4.4.7. Pruebas

Esta sección realizará todas las pruebas que el docente ha solicitado en los Product Backlog y que deben ser aceptados de acuerdo con su criterio de aceptación.

##### 4.4.7.1. Pruebas Sprint 1

Requerimiento número 1: En el menú principal debe haber opciones de jugar, cómo jugar, configuraciones y salir de la aplicación.

**Criterio de aceptación:** Los componentes gráficos son intuitivos y fáciles de visualizar.

**Figura 23. Criterio de aceptación:**

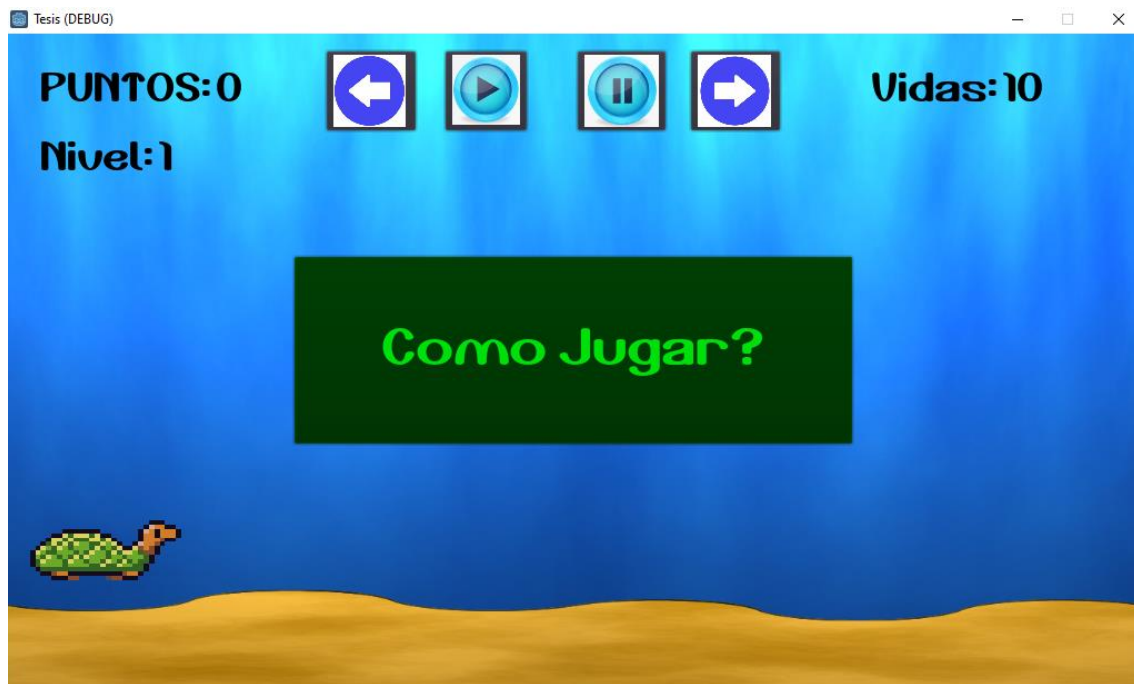


**Descripción de prueba:** Se puede observar que la interfaz es intuitiva, en español nativa y se visualiza de manera clara.

**Requerimiento número 2:** En la escena de clasificación de basura y no basura debe haber opciones de pausar, continuar, cambiar de escenas y cómo jugar la respectiva escena.

**Criterio de aceptación:** Los componentes gráficos son intuitivos y fáciles de visualizar.

**Figura 24. Criterio de aceptación**



**Descripción de prueba:** Se puede observar que la interfaz es intuitiva con sus botones de pausa, continuar, cómo jugar y también el cambio de escenas.

**Requerimiento número 3:** En la escena de clasificación de basura y no basura debe haber opciones de pausar, continuar, cambiar de escenas y cómo jugar la respectiva escena.

**Criterio de aceptación:** Los componentes gráficos son intuitivos y fáciles de visualizar.

**Figura 25. Criterio de aceptación:**



**Descripción de prueba:** Se puede observar que la interfaz es intuitiva con sus botones de pausa, continuar, cómo jugar y también el cambio de escenas.

**Requerimiento número 4:** Creación de videos para escenas principal, uno y dos

**Criterio de aceptación:** Los vídeos generan buena atención visual, son llamativos y explicativos

**Figura 26. Criterio de aceptación:**



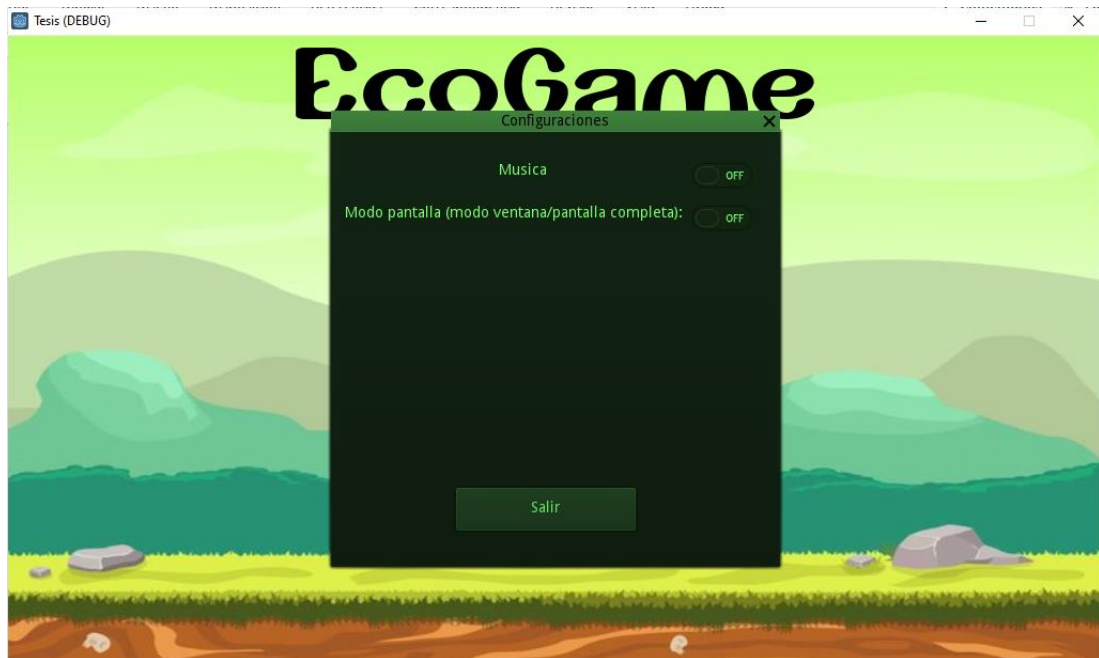
**Descripción de prueba:** Se tiene un personaje creado por inteligencia artificial que explica cómo jugar, de la manera más clara y concisa posible.

#### **4.4.7.2. Pruebas Sprint 2**

**Requerimiento número 5:** En el menú principal debe haber funcionalidad de cada una de sus opciones.

**Criterio de aceptación:** El menú de configuraciones permite expandir o comprimir la pantalla y activar/desactivar el sonido general.

**Figura 27. Criterio de aceptación**



**Descripción de prueba:** El menú de configuraciones permite acceder a opciones de audio y video sencillas.

**Requerimiento número 6:** En la escena uno el personaje es animado y puede recolectar desechos y criaturas marinas

**Criterio de aceptación:** El personaje principal se mueve e interacciona de forma positiva o negativa con cada tipo de objeto que cae al suelo marino.

**Figura 28. Criterio de aceptación**

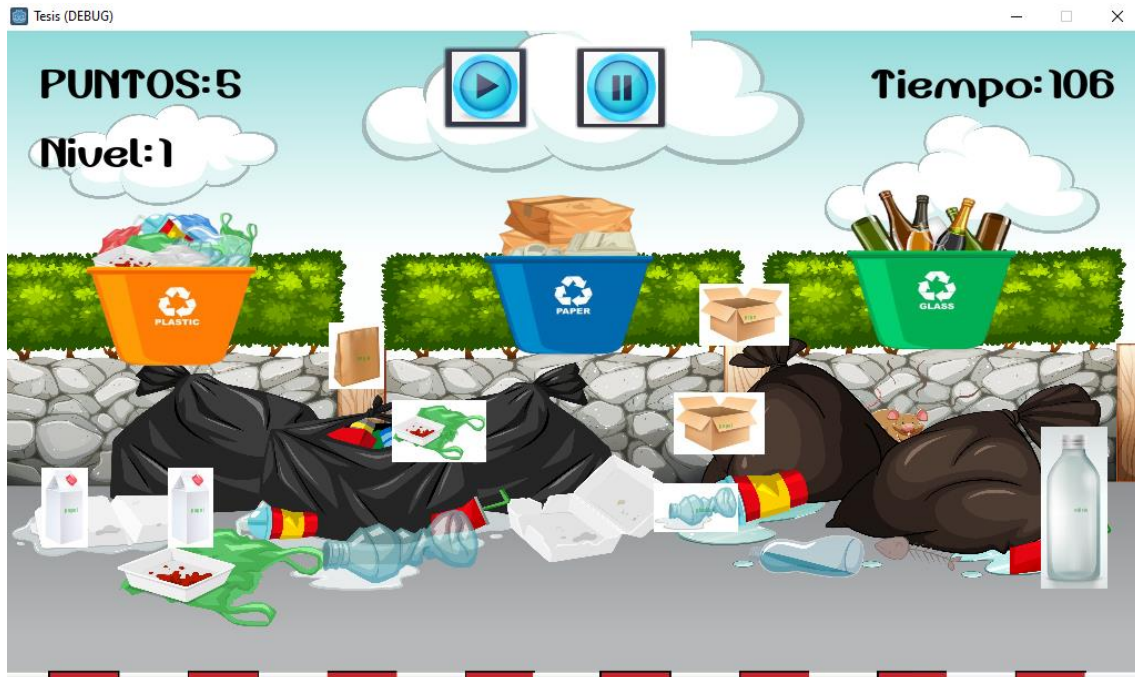


**Descripción de prueba:** el personaje principal interacción con los objetos que caen al suelo marino, sea forma negativa o positiva.

**Requerimiento número 7:** En la escena dos el usuario puede clasificar desechos como plástico, vidrio y papel

**Criterio de aceptación:** El usuario puede arrastrar los objetos y clasificarlos según su tipo. Admite o no interacciona según sea el caso.

Figura 29. Criterio de aceptación



**Descripción de prueba:** los objetos de estar en orden horizontal pasan a moverse en la escena y suma puntos cuando el tacho que corresponde es el correcto.

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1. Conclusiones:**

- El videojuego es una herramienta tecnológica digital que permite a escolares o infantes la familiarización con el tema del cuidado medioambiental.
- El desarrollo del videojuego permitió familiarizarse con términos de la metodología Scrum, la cual es esencial para obtener productos de software de buena calidad.
- El videojuego es esencial para grupos de personas que requieran practicar y aprender el uso de periféricos de computador y familiarizarse con medios visuales.

#### **5.2. Recomendaciones:**

- Se recomienda interaccionar al usuario final con el videojuego para encontrar inconformidades o sugerencias.
- Se recomienda formar un lazo amigable con el usuario para intercambio de ideas sin presión ni exigencias.
- Se recomienda usar terminología adecuada con el usuario no técnico puesto que el uso de este puede confundirlo y desanimarlo.

## BIBLIOGRFÍA

- Mula, J. (2022, 23 diciembre). *Pros y contras de programar con Unity vs. Unreal Engine*. Obtenido de <https://www.deustoformacion.com/blog/disenio-produccion-audiovisual/pros-contras-programar-unity-vs-unreal-engine>
- Huéspedes. (s. f.). *Cambio climático*. Obtenido de [https://huespedes.cica.es/gimadus/33/03\\_el\\_cambio\\_climatico.html](https://huespedes.cica.es/gimadus/33/03_el_cambio_climatico.html)
- Papa Francisco. (2015). *Laudato si'*. Obtenido de [https://www.vatican.va/content/francesco/es/encyclicals/documents/papa-francesco\\_20150524\\_enciclica-laudato-si.html](https://www.vatican.va/content/francesco/es/encyclicals/documents/papa-francesco_20150524_enciclica-laudato-si.html)
- Guía de buenas prácticas ambientales*. (2018). Obtenido de [https://promocionsocial.org/wp-content/uploads/2018/04/Gu%C3%ADa-Buenas-Pr%C3%A1cticas-Ambientales\\_Fundaci%C3%B3n-Promoci%C3%B3n-Social-1.pdf](https://promocionsocial.org/wp-content/uploads/2018/04/Gu%C3%ADa-Buenas-Pr%C3%A1cticas-Ambientales_Fundaci%C3%B3n-Promoci%C3%B3n-Social-1.pdf)
- EFEverde, R. (2022, 28 julio). *Los videojuegos, una herramienta para concienciar sobre el medioambiente*. Obtenido de <https://efeverde.com/los-videojuegos-una-herramienta-para-concienciar-sobre-el-medioambiente/>
- libGDX vs Godot*. (s. f.). Obtenido de <https://glusoft.com/tutorials/libGDX-vs-Godot>
- Qué es C++: Características y aplicaciones*. (2022, 2 noviembre). Obtenido de <https://openwebinars.net/blog/que-es-cpp/>
- Ventajas de Java*. (s. f.). Obtenido de <https://www.ibm.com/docs/es/aix/7.3?topic=monitoring-advantages-java>
- Just a moment*. (s. f.). Obtenido de <https://techvidvan.com/tutorials/pros-and-cons-of-java/>

Núñez-Barriopedro, E. (2020). *Los videojuegos en la educación: Beneficios y perjuicios*. Obtenido de

<https://www.redalyc.org/journal/1941/194163269012/html/>

*Ley de Moore – Ahora y en el Futuro*. (s. f.). Obtenido de

<https://www.intel.la/content/www/xl/es/newsroom/opinion/moore-law-now-and-in-the-future.html>

*¿Por qué tus hijos sí deberían jugar a videojuegos?* (2021, 22 abril). Obtenido de

<https://www.iberdrola.com/talento/beneficios-videojuegos-aprendizaje>

*Introducción a Godot*. (s. f.). Obtenido de

[https://docs.godotengine.org/es/stable/getting\\_started/introduction/introduction\\_to\\_godot.html](https://docs.godotengine.org/es/stable/getting_started/introduction/introduction_to_godot.html)

G. (2022, 11 noviembre). *The Game Loop*. Obtenido de

<https://www.gameludere.com/2019/11/01/game-loop/>

*¿Qué es Python? - Explicación del lenguaje Python - AWS*. (s. f.). Obtenido de

<https://aws.amazon.com/es/what-is/python/>

Engine, G. (s. f.). *Features*. Obtenido de <https://godotengine.org/features/>

C. (2021, 22 diciembre). *¿Cuáles son las teclas de direcciones?*. Obtenido de

<https://www.cavsi.com/preguntasrespuestas/cuales-son-las-teclas-de-direcciones/>

*OpenGameArt.org*. (s.f.). Obtenido de <https://opengameart.org/>

*Sprites pack 16px\*16px over 150 sprites! by RRFreelance - PixelBurner*. (s.f.).

Obtenido de <https://rrfreelance-pixelburner.itch.io/sprites-pack-16px-over-150>

Freepik. (s. f.). *Freepik Recursos gráficos para todos*. Obtenido de

<https://www.freepik.es/>

- Sampieri, R. H., Collado, C. F., Lucio, P. B., Valencia, S. M., & Torres, C. P. M. (2014). *Metodología de la investigación*. Obtenido de <https://www.uca.ac.cr/wp-content/uploads/2017/10/Investigacion.pdf>
- Blaxter, L., Hughes, C., & Tight, M. (2000). *Cómo se hace una investigación*. Gedisa Editorial S A.
- García, J. a. C., Lucas, J. L., & Pulido, J. R. V. (2001). *Manual de investigación bibliográfica y documental: teoría y práctica*.
- Blankenship, J., Bussa, M., & Millett, S. (2011). *Managing Agile Projects with Scrum*. In *Apress eBooks* (pp. 13–27). Apress. [https://doi.org/10.1007/978-1-4302-3534-7\\_2](https://doi.org/10.1007/978-1-4302-3534-7_2)
- Rubin, K. S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley Professional.
- Canive, T. (2020, September 25). *Metodología SCRUM: ¿qué es y cómo aplicarlo en tu trabajo?*. Obtenido de <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-scrum>
- Piaget, J., & Inhelder, B. (1997). *Psicología del niño*. Ediciones Morata.
- Montagnana, V. (2008). *Videojuegos*. Ediciones Robinbook.
- Céspedes, Ruth. (2023, 20 de marzo). Requisito para la creación del videojuego [Comunicación Personal].
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2014). *Metodología de la investigación*. McGraw-Hill Education. Obtenido de [http://caterina.udlap.mx/u\\_dl\\_a/tales/documentos/lcp/texson\\_a\\_gg/capitulo4.pdf](http://caterina.udlap.mx/u_dl_a/tales/documentos/lcp/texson_a_gg/capitulo4.pdf)

## **ANEXOS**

### **Anexos 1. Formato de la entrevista para recolección de historias de usuario Pontificia Universidad Católica del Ecuador**

#### **Formato para la Entrevista de Levantamiento de Historias de Usuario**

1. ¿Por qué cree que los videojuegos podrían ser una herramienta efectiva para enseñar a los niños sobre el cuidado del medio ambiente?
2. ¿Cree que los videojuegos podrían complementar o reemplazar la enseñanza tradicional en el aula en lo que respecta al cuidado medioambiental? ¿Por qué?
3. ¿Qué tipo de videojuegos cree que serían los más efectivos para enseñar sobre el cuidado medioambiental?
4. ¿Cree que los videojuegos podrían motivar a los estudiantes a involucrarse más en la protección del medio ambiente? ¿Por qué?
5. ¿Cuáles serían las principales preocupaciones o desafíos que tendríamos que enfrentar si decidiéramos incorporar videojuegos en el currículo de enseñanza de cuidado medioambiental en la escuela?