

z



# **Diseño de experiencias en la implementación de un juego RPG con mecánicas de combate estratégico**

Autor: Gaona Pineda, Jheyson Steven

Tutor: Duch Gavaldà, Jordi.

Profesor: Arnedo, Joan Moreno

Máster Universitario en Diseño y Programación de Videojuegos

Diseño de experiencias

14/01/2024

# Créditos/Copyright



Esta obra está sujeta a una licencia de Reconocimiento- NoComercial-SinObraDerivada [3.0 España de Creative Commons.](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>Diseño de experiencias en la implementación de un juego RPG con mecánicas de combate estratégico.</i>
<b>Nombre del autor:</b>	<i>Jheyson Gaona Pineda</i>
<b>Nombre del colaborador/a docente:</b>	<i>Jordi Duch Gavaldà</i>
<b>Nombre del PRA:</b>	<i>Joan Arnedo Moreno</i>
<b>Fecha de entrega:</b>	<i>01/2024</i>
<b>Titulación o programa:</b>	<i>Diseño de experiencias</i>
<b>Área del Trabajo Final:</b>	<i>Máster Universitario en Diseño y Programación de Videojuegos</i>
<b>Idioma del trabajo:</b>	<i>Español</i>
<b>Palabras clave</b>	<i>RPG, Sistema de Combate, Unity</i>
<b>Resumen del Trabajo (máximo 250 palabras):</b>	
<p>El presente trabajo se enfoca en el diseño y desarrollo de un videojuego de rol RPG que se destaca por la implementación de un sistema de combate estratégico en tercera persona, dicho sistema se distingue por la clasificación de los personajes en diversas clases, cada uno con habilidades únicas. El videojuego además combina elementos de aventura y exploración, ambientado en un mundo de fantasía que brinda al jugador una experiencia inmersiva en la que pueda embarcarse en una emocionante hazaña y enfrentar desafiantes retos.</p> <p>El proyecto se centra en la creación de un sistema de combate inspirado en los héroes y jefes de mazmorras de World of Warcraft, donde se requiere que el jugador tome decisiones tácticas durante los enfrentamientos con personajes no jugables que disponen de mecánicas únicas. Para llevar a cabo el desarrollo se utiliza el motor gráfico de Unity 3D donde se aborda aspectos claves del diseño de juegos, la programación, la narrativa y la jugabilidad.</p> <p>Para el desarrollo y planificación efectiva del proyecto se opta por hacer uso de la metodología ágil Scrum, debido a que se elabora un producto que aborda el diseño de experiencia de un juego y para ello es necesario obtener el mejor resultado posible. Scrum facilita un seguimiento continuo del proyecto, permitiendo gestionar y organizar los tiempos de manera eficiente.</p>	

**Abstract (in English, 250 words or less):**

The present work focuses on the design and development of an RPG role-playing video game that stands out for the implementation of a strategic third-person combat system. This system is characterized by the classification of characters into various classes, each with unique abilities. The video game also combines elements of adventure and exploration, set in a fantasy world that provides players with an immersive experience in which they can embark on exciting quests and face challenging challenges.

The project revolves around the creation of a combat system inspired by the heroes and dungeon bosses of World of Warcraft. It requires the player to make tactical decisions during encounters with non-playable characters that have unique mechanics. Unity 3D graphic engine is employed to address key aspects of game design, programming, narrative, and gameplay.

For the effective development and planning of the project, the agile Scrum methodology is chosen. This decision is based on the need to create a product that addresses the design of a gaming experience, requiring the best possible outcome. Scrum enables continuous project monitoring, allowing for efficient time management and organization.

## Dedicatoria

Dedico este Trabajo de Fin de Máster con profundo agradecimiento a mis padres, José Gaona y Teresa Pineda, cuyo amor y apoyo constante han sido la fuerza impulsora detrás de cada paso que he dado.

A mis queridas hermanas Jhuliana, Lady y Estefanía, agradezco de todo corazón su inquebrantable apoyo y las fuerzas que me han brindado para seguir adelante, incluso cuando no han podido estar físicamente a mi lado.

A mi compañera de vida, Jazmín Andrade, que me alentó con ánimos y valentía para poder alcanzar y cumplir mi sueño de ser un experto en el área. Su apoyo incondicional ha sido mi mayor fortaleza y motivación a lo largo de este camino, gracias por ser mi fuente constante de inspiración y por compartir este viaje a mi lado.

A mi jefe y coordinador de laboratorio, Ing. Sebastián Quevedo, quien me supo orientar en mi trabajo gracias a su vasto conocimiento en el área de la programación, además de su inmenso apoyo y por brindarme la oportunidad de enfrentar este reto. Su liderazgo ha sido esencial en mi trayectoria académica y profesional.

## Agradecimientos

Quiero expresar mi profundo agradecimiento a Dios, fuente de fortaleza y guía en cada paso de mi vida. Agradezco la bendición de contar con el amor y apoyo incondicional de mis padres, José Gaona y Teresa Pineda, quienes siempre han sido mi faro en la oscuridad y mis mayores impulsores hacia el éxito.

Quisiera expresar mi sincero agradecimiento a Sebastián Quevedo, mi jefe y coordinador de laboratorio, por su invaluable apoyo y orientación en el desarrollo de este proyecto. Agradezco sinceramente la oportunidad que me brindó para priorizar la elaboración de mi TFM, así como su disposición constante para brindarme valiosos consejos y guía en cuestiones relacionadas con la programación. Su contribución ha sido fundamental para el éxito de este trabajo.

## Resumen

El presente trabajo se enfoca en el diseño y desarrollo de un videojuego de rol RPG que se destaca por la implementación de un sistema de combate estratégico en tercera persona, dicho sistema se distingue por la clasificación de los personajes en diversas clases, cada uno con habilidades únicas. El videojuego además combina elementos de aventura y exploración, ambientado en un mundo de fantasía que brinda al jugador una experiencia inmersiva en la que pueda embarcarse en una emocionante hazaña y enfrentar desafiantes retos.

El proyecto se centra en la creación de un sistema de combate inspirado en los héroes y jefes de mazmorras de World of Warcraft, donde se requiere que el jugador tome decisiones tácticas durante los enfrentamientos con personajes no jugables que disponen de mecánicas únicas. Para llevar a cabo el desarrollo se utiliza el motor gráfico de Unity 3D donde se aborda aspectos claves del diseño de juegos, la programación, la narrativa y la jugabilidad.

Para el desarrollo y planificación efectiva del proyecto se opta por hacer uso de la metodología ágil Scrum, debido a que se elabora un producto que aborda el diseño de experiencia de un juego y para ello es necesario obtener el mejor resultado posible. Scrum facilita un seguimiento continuo del proyecto, permitiendo gestionar y organizar los tiempos de manera eficiente.

### Palabras clave

*RPG, Sistema de Combate, Unity*

## Abstract

The present work focuses on the design and development of an RPG role-playing video game that stands out for the implementation of a strategic third-person combat system. This system is characterized by the classification of characters into various classes, each with unique abilities. The video game also combines elements of adventure and exploration, set in a fantasy world that provides players with an immersive experience in which they can embark on exciting quests and face challenging challenges.

The project revolves around the creation of a combat system inspired by the heroes and dungeon bosses of World of Warcraft. It requires the player to make tactical decisions during encounters with non-playable characters that have unique mechanics. Unity 3D graphic engine is employed to address key aspects of game design, programming, narrative, and gameplay.

For the effective development and planning of the project, the agile Scrum methodology is chosen. This decision is based on the need to create a product that addresses the design of a gaming experience, requiring the best possible outcome. Scrum enables continuous project monitoring, allowing for efficient time management and organization.

### Keywords

*RPG, Combat System, Unity*



# Índice

<b>1. Introducción</b>	<b>13</b>
<b>1.1. Contexto y justificación</b>	<b>13</b>
<b>1.2. Descripción</b>	<b>13</b>
<b>1.3. Objetivos generales</b>	<b>14</b>
1.3.1. Objetivos principales	14
1.3.2. Objetivos secundarios	14
<b>1.4. Metodología y proceso de trabajo</b>	<b>15</b>
1.4.1. Historias de usuario	16
1.4.2. Product Backlog	17
1.4.3. Plan de lanzamiento (Sprints)	20
<b>1.5. Planificación</b>	<b>21</b>
<b>1.6. Presupuesto</b>	<b>22</b>
<b>1.7. Estructura del resto del documento</b>	<b>24</b>
<b>2. Análisis de mercado</b>	<b>25</b>
<b>2.1. Público objetivo y perfiles de usuario</b>	<b>25</b>
<b>2.2. Marco teórico</b>	<b>26</b>
2.2.1. Historia de los videojuegos RPG	26
2.2.2. Diseño y desarrollo de RPG	27
2.2.3. Aspectos técnicos de los RPG	28
2.2.4. Trabajos relacionados	29
2.2.5. Comparación de trabajos relacionados	35
2.2.6. Motores gráficos para desarrollo de videojuegos	37
2.2.7. Comparación de motores gráficos para desarrollo de videojuegos	40
<b>3. Propuesta</b>	<b>41</b>
<b>3.1. Problemática</b>	<b>41</b>
<b>3.2. Definición de objetivos/especificaciones del producto</b>	<b>41</b>
3.2.1. Plan de niveles	42
3.2.2. Plan de personajes jugables	42
3.2.3. Plan de habilidades	43

3.2.4.	Plan para tipos de enemigos.....	44
3.2.5.	Requisitos funcionales .....	45
3.2.6.	Requisitos no funcionales.....	45
<b>4.</b>	<b>Diseño .....</b>	<b>46</b>
4.1.	Herramientas y entorno de desarrollo .....	46
4.2.	Inventario de recursos - Assets.....	47
4.3.	Enfoque de juego .....	49
4.3.1.	Mecánicas, Dinámicas y Estética .....	49
4.4.	Arquitectura general de la aplicación .....	51
4.5.	Arquitectura de la información y diagramas de navegación .....	52
4.5.1.	Entidad relación sistema de habilidades mediante Scriptable Object.....	52
4.5.2.	Entidad relación sistema de combate .....	53
4.5.3.	Árboles de decisiones y máquinas de estado de la IA .....	54
4.6.	Diseño gráfico e interfaces .....	56
<b>5.</b>	<b>Implementación .....</b>	<b>58</b>
5.1.	Sprint 1.....	59
5.2.	Sprint 2.....	67
5.3.	Sprint 3.....	70
5.4.	Sprint 4.....	78
5.5.	Sprint 5.....	81
<b>6.</b>	<b>Demostración.....</b>	<b>85</b>
6.1.	Guía de usuario .....	85
	<b>Conclusiones y líneas de futuro .....</b>	<b>87</b>
6.2.	Conclusiones .....	87
6.3.	Líneas de futuro.....	87
	<b>Bibliografía .....</b>	<b>89</b>

# Figuras y tablas

## Índice de figuras

Figura 1 Flujo de la metodología Scrum [2] .....	16
Figura 2 Formato de las historias de usuario para el proyecto .....	17
Figura 3 Diagrama de Gantt que describe el flujo de trabajo del presente proyecto.....	21
Figura 4 Representación visual del juego Dungeons and Dragons.....	26
Figura 5 Presentación visual de la jugabilidad en World of Warcraft .....	31
Figura 6 Presentación visual de la jugabilidad en Genshin Impact .....	33
Figura 7 Presentación visual de la jugabilidad en Elden Ring .....	35
Figura 8 Principales interfaces del motor gráfico de Unity .....	38
Figura 9 Principales interfaces del motor gráfico de Unreal Engine.....	39
Figura 10 Propuesta del diseño de nivel a integrar en el proyecto.....	42
Figura 11 Propuesta de personajes jugables a integrar en el proyecto.....	43
Figura 12 Propuesta de habilidades a integrar en el proyecto. ....	44
Figura 13 Propuesta de enemigos a integrar en el proyecto. ....	44
Figura 14 Arquitectura general del videojuego RPG .....	51
Figura 15 Entidad relación del sistema de creación de habilidades y manejo de datos de los mismos.....	52
Figura 16 Entidad relación del sistema de combate de personajes jugables y no jugables .....	53
Figura 17 Árbol de decisión de la IA para gestión de movimiento y objetivo.....	55
Figura 18 FMS – IA Gestor de combate para personajes no jugables que poseen o no habilidades.....	56
Figura 19 Diseño de la UI que se integra al desarrollo del RPG.....	57
Figura 20 Asignación de teclado para el control de movimiento y cámara del personaje jugable .....	60
Figura 21 Funcionalidad para el movimiento del personaje jugable.....	60
Figura 22 Creación y diseño del nivel base para desarrollar las actividades del juego RPG .....	62
Figura 23 Funcionamiento para el movimiento del personaje no jugable.....	63
Figura 24 Controlador de animaciones para personajes jugables – clase guerrero .....	65
Figura 25 Controlador de animaciones para personajes no jugables – Monarca Fungico.....	66
Figura 26 Ruta del sistema de habilidades para la creación de una nueva habilidad.....	68
Figura 27 Scriptable Object de habilidad de daño directo que integra efecto temporal .....	68
Figura 28 Ruta del sistema de máquina de estados que controla la AI de habilidades del personaje no jugable.....	69
Figura 29 Scriptable Object de gestor de AI de habilidades para los personajes no jugables .....	70
Figura 30 Diagrama Entidad-Relación del sistema de estadísticas de los personajes .....	72
Figura 31 Script del controlador de estadísticas para cada tipo de personaje.....	73
Figura 32 Diagrama Entidad-Relación del sistema de combate de los personajes .....	74
Figura 33 Sistema de control de combate de los personajes desde la perspectiva del editor de Unity.....	75
Figura 34 Sistema de combate para el personaje jugable .....	76
Figura 35 Sistema de combate mediante ataques y uso de habilidades del personaje jugable.....	77
Figura 36 Sistema de combate para el personaje no jugable .....	77
Figura 37 Sistema de combate y habilidades del personaje no jugable.....	78
Figura 38 Diseño final de UI que integra el juego RPG, basada en el diseño propuesto.....	80

Figura 39 Funcionalidad del gestor de spawn de NPCs .....	81
Figura 40 Sistema de gestión de recompensas por parte de los personajes .....	82
Figura 41 Sistema de despojo de recompensa de los NPC.....	83
Figura 42 Ambiente creador para el videojuego RPG estilo Stylized .....	84
Figura 43 Teclas de acceso para jugar.....	85

## Índice de tablas

Tabla 1 Artefactos de Scrum .....	15
Tabla 2 Listado de las historias de usuario presentes en el Product Backlog .....	19
Tabla 3 Plan de lanzamiento de los Sprints para la asignación del flujo de trabajo.....	20
Tabla 4: Organización de entregables de las PEC, fechas clave.....	21
Tabla 5 Presupuesto de hardware.....	22
Tabla 6 Presupuesto de software .....	22
Tabla 7 Presupuesto de recursos humanos.....	23
Tabla 8 Estimación total de presupuesto para el proyecto .....	23
Tabla 9 Resumen comparativo de los trabajos relacionados.....	36
Tabla 10 Comparación de motores gráficos para el desarrollo de videojuegos.....	40
Tabla 11 Requisitos técnicos mínimos y recomendados para que funcione la herramienta Unity .....	46
Tabla 12 Inventario de Assets para la construcción del proyecto. ....	48
Tabla 13 Planificación de tareas del sprint 1.....	59
Tabla 14 Planificación de tareas del sprint 2.....	67
Tabla 15 Planificación de tareas del sprint 3.....	71
Tabla 16 Planificación de tareas del sprint 4.....	79
Tabla 17 Planificación de tareas del sprint 5.....	82

# 1.Introducción

## 1.1. Contexto y justificación

Los videojuegos de rol RPG<sup>1</sup> que incorporan sistemas de combate estratégico no solo atraen a los usuarios por sus mecánicas, diseño de niveles y técnicas de inteligencia artificial, estos videojuegos trascienden el entretenimiento, debido a que representan la culminación de una experiencia minuciosamente programada, donde cada elemento se encuentra diseñado y configurado con la intención de sumergir al jugador en mundo completamente imaginario.

El presente trabajo se enfoca en desarrollar una experiencia de juego RPG con mecánicas de combate estratégico, es decir, se construye una experiencia única que cautive al jugador y para conseguirlo se elabora una arquitectura de juego basada en sistema de clases de personajes y habilidades, mismas que permitan escalar con el tiempo, en otros términos, optimizar el trabajo para la integración de nuevo contenido, lo que resulta en una mejora notable que fomente la experiencia de juego.

En un RPG, la diversidad de personajes y habilidades proporciona al jugador múltiples estilos de juego y estrategia que aporta variedad en la resolución de desafíos, permitiendo fomentar la rejugabilidad, dicho de otra manera, se puede probar diversas combinaciones y experimentar nuevas formas de jugar, donde la inmersión del jugador con respecto al mundo del juego puede enriquecerse en profundidad, estrategia, diversidad y sensación de progreso.

El deseo de elaborar este tipo de experiencias proviene de mi gran pasión por los juegos de rol MMORPG<sup>2</sup>, además de la gran motivación de aprender y superar desafíos al momento de desarrollar juegos complejos.

## 1.2. Descripción

Durante el transcurso del máster, he adquirido un vasto conocimiento en el ámbito del diseño de videojuegos y la programación. Este aprendizaje me ha permitido descubrir aspectos que suelen pasar desapercibidos al consumir grandes títulos como World of Warcraft, un juego de rol MMORPG, mismo que en particular ha sido el principal incentivo que me ha impulsado a adentrarme en este fascinante mundo del desarrollo de videojuegos. El presente trabajo me brinda la oportunidad de aplicar y poner en práctica todas las habilidades, conocimientos y conceptos adquiridos en programación, con la finalidad de entregar un producto único y distintivo en este campo.

Los juegos de rol RPG son actualmente un tema de suma relevancia debido a su evolución en la jugabilidad y la experiencia que el jugador puede llegar a percibir, es decir, el usuario por medio de un avatar que controla en tiempo real tiene la oportunidad de vivir una vida de fantasía o asumir el papel del protagonista, este tipo de videojuegos se encuentran centrados en el combate y acción [1].

---

<sup>1</sup> Role Playing Game “Género de videojuego donde el jugador asume roles de personajes ficticios”.

<sup>2</sup> Massively Multiplayer Online Role-Playing Game “Juego de rol multijugador masivo en línea”.

### 1.3. Objetivos generales

El presente TFM establece los siguientes objetivos a alcanzar durante su desarrollo.

#### 1.3.1. Objetivos principales

Objetivos de la aplicación/producto/servicio:

- Crear un videojuego RPG en tercera persona con clases de personajes distintivos que fomenten la diversidad y complejidad estratégica.
- Desarrollar un sistema de habilidades mediante Scriptable Objects, mismas que sean variadas, únicas y que se alineen a cada clase de personaje, incluyendo beneficios, curaciones y daño de diversos tipos (directos, proyectiles, AoE) con la posibilidad de integrar efectos temporales.
- Desarrollar un sistema de combate que habilite la utilización de poderes y una amplia gama de armas, asegurando la sincronización fluida entre las animaciones de ataques y habilidades.
- Implementar técnicas de inteligencia artificial que permitan a los personajes no jugables adaptarse dinámicamente al entorno del juego y presentar desafíos apropiados mediante el empleo de máquinas de estados o árboles de decisiones.

Objetivos para el cliente/usuario:

- Proporcionar un videojuego con una experiencia inmersiva y emocionante que permita al jugador embarcarse en una hazaña llena de desafíos.
- Brindar un videojuego con objetivos claros, mismos que garanticen una experiencia de usuario satisfactoria de inicio a fin.

Objetivos personales del autor del TF:

- Demostrar habilidades excepcionales en el ámbito de la programación, al ofrecer un producto final sin presencia de errores que causen bugs o glitches
- Trabajar relativamente con metodologías ágiles para ofrecer un producto organizado y funcional que represente la calidad del trabajo del autor.

#### 1.3.2. Objetivos secundarios

Objetivos adicionales que enriquecen el TF.

- Diseñar un sistema de recompensas y progresión de niveles que motive al jugador a seguir explorando y enfrentando desafíos dentro del juego.
- Integrar clases de personajes (guerrero, mago, arquero, curandero) con habilidades distintivas.
- Implementar mecánicas de combate específicas para cada clase que se adapten a su estilo de juego, permitiendo estrategias variadas para enfrentar desafíos.

## 1.4. Metodología y proceso de trabajo

Para la construcción del presente proyecto se opta por emplear la metodología ágil Scrum, esta elección es fundamental debido a que se elabora un videojuego que prioriza la experiencia de usuario y para ello se busca obtener el mejor resultado posible en la construcción del mismo. Scrum brinda un enfoque de trabajo interactivo-incremental, dicho de otra manera, se define el término interactivo como referencia a periodos cortos de tiempo, donde se diseña y construye una porción del producto el cual toma nombre de incremento, por otro lado, el término incremental, hace referencia a la unión de cada incremento para generar un producto completo.

Scrum es una metodología que se caracteriza por su flexibilidad y cortos ciclos de vida, donde se puede asignar y desarrollar tareas a medida que avance el proyecto, el ciclo de vida de Scrum está conformado por cuatro fases: planeación, puesta en escena, desarrollo y entrega [2]. Esta metodología a diferencias de otras metodologías ágiles como Extreme Programming (XP) no requiere de entregables precipitados y cambios constantes o como la metodología Kanban que requiere de la asignación de todas las tareas desde la fase de planificación. Scrum por otro lado es estricto con los entregables, caso requerido para la ejecución correcta del presente proyecto.

Según Trigas Gallego menciona que, “Scrum al ser una metodología de desarrollo ágil tiene como base la idea de creación de ciclos breves para el desarrollo, que comúnmente se llaman iteraciones y que en Scrum se llaman ‘Sprints’” [3].

Scrum dispone de artefactos que ofrecen dirección y transparencia apropiada al equipo de trabajo, mismos que se encuentran organizados tal como se aprecia en la Tabla 1.

Nombre del artefacto	Descripción
Product Backlog	Es la lista de actividades planificadas que se pretenden desarrollar dentro del proyecto.
Sprint Backlog	Es la lista de tareas que se realizan en cada sprint.
Incremento	Es la parte añadida y desarrollada en un sprint, por lo general es una porción del proyecto completamente funcional y operante.

Tabla 1 Artefactos de Scrum

Como se evidencia en la Tabla 1, los artefactos desempeñan un papel importante para la gestión de productividad y calidad del proyecto, garantizando un completo entendimiento al grupo de trabajo con el fin de minimizar la probabilidad de tomar decisiones erróneas.

A continuación, se presenta en la Figura 1 el flujo de desarrollo de la metodología ágil Scrum, donde un Sprint suele tener una duración máxima de un mes y se definen las actividades del trabajo a realizar, mientras que en el Daily Scrum se especifica el trabajo realizado, los obstáculos presentados y que debe realizarse a continuación, por lo general esta actividad se realiza diariamente con el fin de evitar

atrasos en el proyecto. Una vez finalizado el sprint este pasa por su respectiva revisión para ser asignado como incremento del proyecto [2,3].

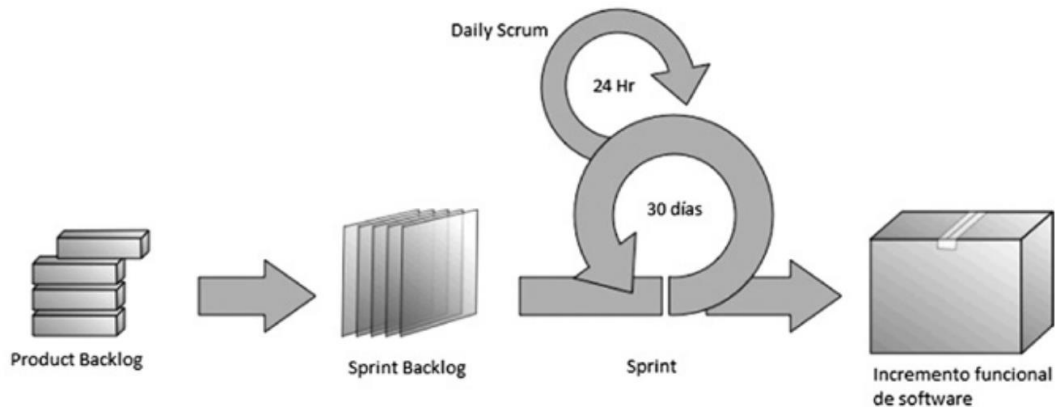


Figura 1 Flujo de la metodología Scrum [2]

En la Figura 1 se observa el proceso de la metodología Scrum, misma que requiere de una organización y estructuración minuciosa, con el fin de garantizar que los Sprints a entregar se agrupen de tal manera que formen parte de un producto final, evitando realizar tareas redundantes y para lograrlo de la mejor manera en el presente proyecto se hace uso de la herramienta Trello<sup>3</sup>.

Esta aplicación visual permite administrar cualquier proyecto y además facilita la supervisión y seguimiento de las actividades [4]. Se utilizan los espacios de trabajo de la herramienta Trello para cada artefacto según la Tabla 1, con el fin de generar un proyecto sólido y poder facilitar la planificación de las actividades a poner en práctica.

### 1.4.1. Historias de usuario

Las historias de usuario son esenciales en el desarrollo de software, ya que constituyen un método para describir requisitos y funcionalidades desde la perspectiva del usuario. Estas narrativas contienen elementos clave que son:

- **Quién:** Describe el usuario o al tipo de usuario que se beneficia de la funcionalidad.
- **Qué:** Describen con precisión la necesidad específica del usuario dentro del sistema o producto, identificando sus requerimientos detallados.
- **Por qué:** Explican el propósito o el valor comercial que satisface esa necesidad para el usuario y/o el negocio.

Estas historias son herramientas efectivas para mantener el enfoque en las necesidades reales de los usuarios. Facilitan una comunicación clara y eficiente entre el equipo de desarrollo y los interesados en el producto, permitiendo una comprensión profunda de los requisitos y expectativas de los usuarios finales.

<sup>3</sup> <https://trello.com>



Como punto de partida del proyecto, es necesario detallar las historias de usuario<sup>4</sup> y para lograrlo de manera efectiva se hace uso del siguiente formato, tal como se aprecia en la Figura 2.

**Anverso**

<b>ID</b>	<b>Título</b>
<b>Descripción:</b>	
<b>Estimación:</b>	
<b>Prioridad:</b>	<b>Dependiente de:</b>

**Reverso**

<b>Pruebas de Aceptación:</b> Criterio 1 Criterio 2 Criterio n...
---

Figura 2 Formato de las historias de usuario para el proyecto

Como se observa en la Figura 2 el formato a emplear dispone de dos caras, que son:

Cara adversa

- **ID:** Identificador único de la historia de usuario
- **Título:** nombre específico de la historia de usuario.
- **Descripción:** explicación resumida de la historia de usuario.
- **Estimación:** tiempo estimado para dar cumplimiento a la historia de usuario.
- **Prioridad:** nivel de importancia para desarrollar la historia de usuario.
- **Dependencia:** si la historia de usuario depende de otra para su correcto funcionamiento.

Cada reversa

- **Pruebas de aceptación:** criterios de validación para el correcto funcionamiento de la misma.

### 1.4.2. Product Backlog

Siguiendo las directrices fundamentales de la gestión de proyectos según THE PMBOK GUIDE<sup>5</sup>, se utilizará el formato predefinido para elaborar el Product Backlog. Este incluye un resumen detallado de las historias de usuario, como se ilustra en la Tabla 2, a fin de establecer una estructura clara y completa para la planificación y el desarrollo del proyecto. Además, se considerará la adaptación de este formato para satisfacer las necesidades específicas del juego RPG en términos de características y funcionalidades requeridas.

<sup>4</sup> Descripción sencilla y amigable de una función, vista desde la perspectiva del usuario final

<sup>5</sup> Project Manager Body of Knowledge

<b>PRODUCT BACKLOG</b>				
<b>Título del proyecto:</b> Diseño de experiencias en la implementación de un juego RPG con mecánicas de combate estratégico.			<b>Fecha de preparación:</b> 01/10/2023	
<b>ID</b>	<b>Descripción resumida</b>	<b>Prioridad</b>	<b>Historia</b>	<b>Estado</b>
<b>US-01</b>	Como usuario del videojuego, es fundamental tener completa libertad para controlar el personaje, lo que implica la capacidad de moverse autónomamente.	Alta	Programar el movimiento de los personajes jugables.	Completado
<b>US-02</b>	Se pretende incorporar movimientos autónomos a los personajes no jugables, tales como patrullar puntos específicos, recorrer áreas de manera aleatoria, vigilar zonas estratégicas y perseguir a otros personajes de acuerdo a la situación.	Alta	Programar el movimiento de los personajes no jugables.	Completado
<b>US-03</b>	La integración de animaciones en los personajes es esencial para dotarlos de realismo, mejorar el atractivo visual y asegurar una experiencia de juego emocionante y cautivadora.	Alta	Animación de personajes jugables y no jugables.	Completado
<b>US-04</b>	Se necesita un sistema de habilidades que permita la creación de diversos poderes, accesibles tanto para personajes jugables como para aquellos que no lo son.	Alta	Programar sistema de habilidades.	Completado
<b>US-05</b>	Se pretende desarrollar una inteligencia artificial que capacite a los personajes no jugables para utilizar sus habilidades en momentos específicos, con el objetivo de brindar mecánicas de juego y estrategias de combate diferenciadas y distintivas.	Alta	Programar IA para controlar el sistema de habilidades de los personajes no jugables.	Completado
<b>US-06</b>	Es fundamental integrar un sistema de estadísticas para los personajes, permitiendo ajustar sus atributos, gestionar la salud y administrar el recurso principal de manera efectiva.	Alta	Programar sistema de estadísticas de los personajes.	Completado
<b>US-07</b>	Es esencial desarrollar un sistema de combate que permita a los personajes ejecutar ataques, usar habilidades y	Alta	Programar sistema de	Completado

	adoptar mecánicas de combate únicas según su clase.		combate de los personajes.	
<b>US-08</b>	Es necesario diseñar una interfaz de usuario que exhiba las estadísticas de vida y recursos tanto del personaje del jugador como del objetivo de combate actual. Esta función posibilita seguir y controlar el daño infligido tanto a los objetivos como al propio personaje durante el juego.	Media	Diseñar y programar la interfaz de usuario para visualizar las estadísticas de los personajes.	Completado
<b>US-09</b>	Incorporar un sistema para regenerar NPCs estándar después de ser derrotados, asegurando la disponibilidad de desafiantes batallas repetibles, excluyendo a los jefes.	Baja	Programar sistema de Respawn de NPC	Completado
<b>US-10</b>	La entrega de recompensas a los jugadores es fundamental, ya que les permite progresar, superar desafíos y acceder a armamento mejorado, lo que garantiza y estimula su motivación.	Media	Programar sistema de recompensas.	Completado
<b>US-11</b>	Se requiere crear un escenario de mundo abierto que fomente la interacción entre los personajes. Este entorno debe ser diseñado con una estética estilizada que se adapte a un mundo de fantasía.	Media	Creación del escenario.	Completado
<b>US-12</b>	Incorporar música y efectos de sonido al juego para crear una atmósfera inmersiva y sumergir al jugador en la experiencia del juego.	Media	Integración de sonidos.	Completado

Tabla 2 Listado de las historias de usuario presentes en el Product Backlog

Como se aprecia en la Tabla 2, el Product Backlog se encuentra formado por doce historias de usuario.

### 1.4.3. Plan de lanzamiento (Sprints)

En esta sección se organizan los Sprints, mismas que son iteraciones que dividen el proyecto en partes más manejables, es decir, se ha planeado dividir y desarrollar el proyecto en cinco componentes principales que son: personajes, habilidades, sistema de combate, sistema de recompensa y escenario, como se ilustra en la Tabla 3.

Componente	ID Sprint	Historia
Personajes	Sprint 1	• US-01: Programar el movimiento de los personajes jugables.
		• US-02: Programar el movimiento de los personajes no jugables.
		• US-03: Animación de personajes jugables y no jugables.
Habilidades	Sprint 2	• US-04: Programar sistema de habilidades.
		• US-05: Programar IA para controlar el sistema de habilidades de los personajes no jugables.
Sistema de combate	Sprint 3	• US-06: Programar sistema de estadísticas de los personajes.
		• US-07: Programar sistema de combate de los personajes.
	Sprint 4	• US-08: Diseñar y programar una UI para visualizar las barras de salud de los personajes.
		• US-09: Programar sistema de Respawn de NPC.
Sistema de recompensa y Escenario	Sprint 5	• US-10: Programar sistema de recompensas.
		• US-11: Creación del escenario.
		• US-12: Integración de sonidos.

Tabla 3 Plan de lanzamiento de los Sprints para la asignación del flujo de trabajo

La Tabla 3 detalla la planificación de los Sprints para el correcto desarrollo del proyecto, donde se asigna cada componente a un sprint en específico, lo que suma un total de cinco Sprints. Es importante destacar que el cumplimiento de cada sprint implica un progreso que debe ser validado y aprobado para proceder con el siguiente sprint.

## 1.5. Planificación

En esta sección se muestra la detallada planificación del TFM, que cubre todo el proceso desde su inicio hasta su culminación, permitiendo establecer los cimientos esenciales que sustentan la investigación, diseño e implementación del proyecto, buscando asegurar una ejecución efectiva y alcanzar exitosamente los objetivos propuestos. A continuación, la Tabla 4 muestra las fechas claves correspondientes a los entregables para cada PEC<sup>6</sup> propuestos dentro de la asignatura del TFM.

Planificación – Fechas clave				
		Inicio	Fin	Descripción
PEC	1	27/09/2023	08/10/2023	PEC 1 - Plan del proyecto.
	2	09/10/2023	12/11/2023	PEC 2 - Estado del arte y primera versión del proyecto.
	3	13/11/2023	17/12/2023	PEC 3 - Implementación de versión jugable.
	4	18/12/2023	14/01/2023	PEC 4 - Memoria y productos finales.
	5	15/01/2023	31/01/2023	PEC 5 - Defensa virtual – tribunal de TFM.

Tabla 4: Organización de entregables de las PEC, fechas clave

La Tabla 4 detalla los entregables de cada una de las PEC, mismos que están limitadas por un tiempo preestablecido y no pueden prolongarse debido a sus características específicas, por ello, se dispone de aproximadamente cuatro meses para completar el desarrollo del juego RPG propuesto y alcanzar con los objetivos establecidos. Es importante tener en cuenta que se va a desarrollar el proyecto con el esfuerzo de una sola persona, el autor del presente trabajo. Para administrar eficazmente el tiempo asignado en el plan, que incluye fechas clave para cada entrega y la aplicación usando la metodología ágil de Scrum, se presenta en la Figura 3 el Diagrama de Gantt con su respectivo control de tareas.

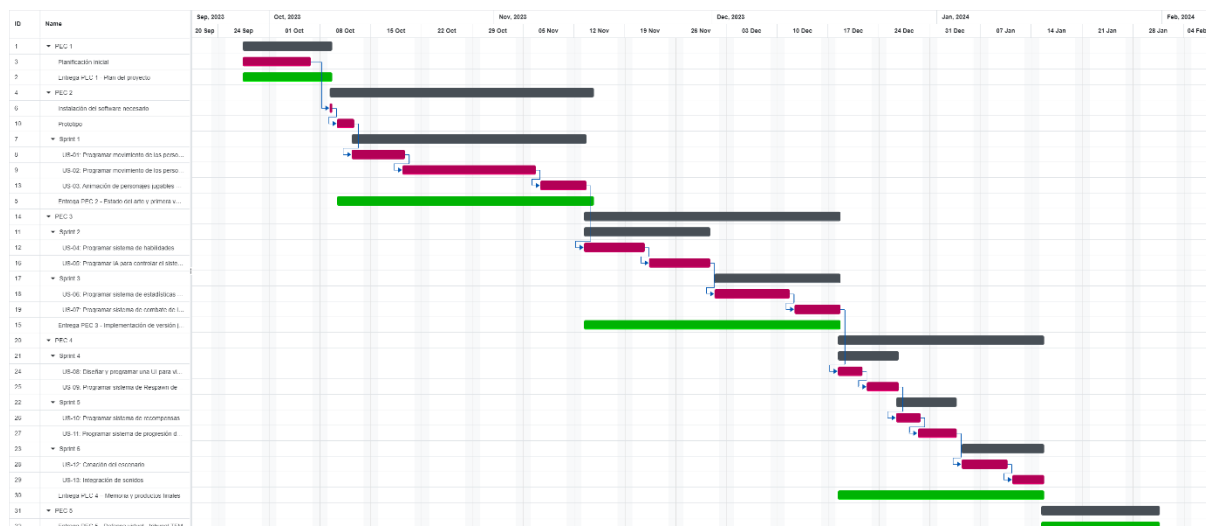


Figura 3 Diagrama de Gantt que describe el flujo de trabajo del presente proyecto

<sup>6</sup> Prueba de Evaluación Continua

## 1.6. Presupuesto

En esta sección se detalla el presupuesto destinado al desarrollo del videojuego RPG, considerando el uso de equipos, activos pagados y otros recursos que, por lo general, demandan tiempo para ser implementados por una sola persona. Los precios mencionados en esta sección pueden variar según la región y la marca específica. Se ha tomado como referencia el costo en el país de Ecuador

La Tabla 5 detalla el presupuesto asignado al hardware utilizado durante la creación del videojuego, este recurso es crucial, ya que impacta directamente en el rendimiento y la capacidad para llevar a cabo el proceso de desarrollo del videojuego.

Presupuesto de Hardware	
Componente	Precio unitario (USD)
Computador Gaming: Core i9, video RTX 4070, 16 GB RAM, Disco SSD de 1 TB, Windows 11.	2000,00
Mouse Gaming Logitech	34,00
Teclado Gaming Genius k8	25,00
Monitor Asus TUF Gaming	350,00
<b>TOTAL:</b>	<b>2409,00</b>

Tabla 5 Presupuesto de hardware

Además del hardware, se necesita software para crear el producto, mismo que puede llegar a ser gratuito o de pago. En la Tabla 6 se detalla el presupuesto destinado al software principal, que es necesario para el proyecto, cabe resaltar que se ha asignado un pequeño margen adicional para posibles imprevistos con respecto a tiempos o la necesidad de comprar recursos adicionales.

Presupuesto de Software	
Componente	Precio unitario (USD)
Unity	00,00
Visual Studio Code	00,00
Recursos de la Asset Store <sup>7</sup> o de HumbleBundle <sup>8</sup>	900,00
Herramientas de productividad (Word, Excel, Diagramas, Trello, etc.)	00,00
Educación y formación profesional (Cursos online, tutoriales, etc.)	150,00
<b>SUBTOTAL:</b>	<b>1050,00</b>
Fondo de reserva (10% del costo subtotal del presupuesto software)	105,00
<b>TOTAL:</b>	<b>1155,00</b>

Tabla 6 Presupuesto de software

<sup>7</sup> <https://assetstore.unity.com/>

<sup>8</sup> <https://www.humblebundle.com/>

La Tabla 7 presenta el presupuesto asignado para los recursos humanos incluidos en el desarrollo y ejecución del proyecto, donde se detallan los costos estimados correspondientes a: roles, salarios y otros gastos relacionados con el equipo humano necesario para realizar las distintas fases y tareas del proyecto. Conforme al apartado 1.5, se estableció un plazo de cuatro meses para el desarrollo, por esa razón la información que se muestra a continuación corresponde al presupuesto de tiempo estipulado.

<b>Presupuesto de Recursos humanos</b>						
<b>Rol</b>	<b>Cantidad</b>	<b>Horas laborales / mes</b>	<b>Precio por hora</b>	<b>Meses requeridos</b>	<b>Salario mensual (USD)</b>	<b>Costo final (USD)</b>
Project Manager	1	160	7,5	4	1200,00	4800,00
Game Designer	1	160	5	4	800,00	3200,00
Desarrollador	2	160	5	3	800,00	2400,00
Diseñador 3D	2	160	4,375	3	700,00	2100,00
Artista conceptual	1	160	3,75	1	600,00	600,00
Animador	1	160	4,375	2	700,00	1400,00
Tester	2	80	3,75	1	300,00	300,00
<b>SUBTOTAL:</b>						<b>14800, 00</b>
Reserva de emergencia (10% del costo subtotal del presupuesto recursos humanos)						1480,00
<b>TOTAL:</b>						<b>16280, 00</b>

Tabla 7 Presupuesto de recursos humanos

El presupuesto final del proyecto para el desarrollo del videojuego RPG comprende los recursos esenciales, como hardware, software y costos relacionados con el equipo humano. Esta estimación financiera se fundamenta en una evaluación detallada de los elementos críticos para la creación del juego. A continuación, en la Tabla 8 se detalla la estimación total del proyecto.

<b>Estimación total</b>	
<b>Componente</b>	<b>Costo Final (USD)</b>
Presupuesto de hardware	2409,00
Presupuesto de software	1155,00
Presupuesto de recursos humanos	16280,00
<b>TOTAL:</b>	<b>19844,00</b>

Tabla 8 Estimación total de presupuesto para el proyecto

La planificación financiera refleja la inversión necesaria para asegurar el éxito y la calidad en el desarrollo del videojuego RPG propuesto, necesitando un total de 19844,00 dólares americanos.

## 1.7. Estructura del resto del documento

- **Análisis de mercado:** Esta sección detalla los pilares esenciales para crear un videojuego RPG, abordando la comprensión del público objetivo, perfiles de usuario y el marco teórico del género. Explora grupos demográficos, preferencias de jugadores y proporciona información valiosa sobre diseño, desarrollo y aspectos técnicos, incluyendo estudios de proyectos previos.
- **Propuesta:** El objetivo es crear un RPG en tercera persona con clases y habilidades únicas utilizando Unity 3D. Se destaca la implementación de un sistema de combate estratégico basado en habilidades y estilos únicos para cada clase, con énfasis en código eficiente mediante programación orientada a objetos (POO) para escalabilidad. Se propone el uso de Scriptable Objects para gestionar habilidades y se busca un estilo visualmente impactante inspirado en un mundo de fantasía.
- **Diseño:** La sección de diseño es el núcleo creativo del proyecto, detallando elementos esenciales como personajes, mecánicas y entornos. Se destaca el proceso de creación de una experiencia cautivadora, fusionando imaginación y funcionalidad para dar vida a una aventura única, con énfasis en la herramienta utilizada, recursos, arquitectura del proyecto y funcionalidad del juego.
- **Implementación:** Este apartado destaca la implementación de un proyecto de videojuego RPG, resaltando la integración exitosa entre la metodología ágil Scrum y la plataforma Unity. Describe desafíos, soluciones y lecciones aprendidas en la creación del juego, detallando el progreso sprint a sprint para comprender cómo contribuyen al logro de objetivos y adaptarse a necesidades emergentes.
- **Demostración:** Esta sección expone el funcionamiento del juego y proporciona una guía de usuario para facilitar la comprensión del videojuego desarrollado e implementado.
- **Conclusiones y líneas futuras:** Se presentan las conclusiones individuales derivadas de los resultados del proyecto, junto con información sobre las futuras mejoras que se planean implementar.



## 2. Análisis de mercado

En esta sección se proporciona un detallado panorama de los pilares fundamentales que influyen en la creación de un videojuego RPG, mismo que se estructura en subsecciones que exploran desde la comprensión minuciosa del público objetivo y los perfiles de usuario hasta una inmersión profunda en el marco teórico que define el género RPG.

En la primera subsección, 2.1, Público objetivo y perfiles de usuario, se introduce en los diversos grupos demográficos y perfiles de jugadores a los que se orienta el juego, profundizando en sus preferencias, expectativas y patrones de juego.

La subsección 2.2, Marco teórico, describe el mundo del RPG desde distintas perspectivas, proceso de diseño y desarrollo, los aspectos técnicos involucrados, además se detallan estudios de proyectos ya realizados lo que permite obtener información valiosa para implementar en el motor gráfico los requisitos solicitados en el aplicativo.

### 2.1. Público objetivo y perfiles de usuario

En esta sección se estudia a detalle las distintas audiencias a las que va dirigido el juego, junto con los perfiles específicos de los jugadores, donde se lleva a cabo un cuidadoso análisis de las métricas, el objetivo es comprender no sólo las preferencias y expectativas del público, sino también a ajustar las mecánicas de juego, la narrativa y los elementos visuales para garantizar una experiencia cautivadora y atractiva para los usuarios finales, seguidamente se enlista los grupos demográficos a considerar:

- **Jugadores RPG tradicionales:** Personas que disfrutan de la profundidad narrativa y el progreso de los personajes, aspectos típicamente asociados con los RPG clásicos.
- **Amantes de combate estratégicos:** Personas a las que les gusta la estrategia en los juegos y disfrutan planificando tácticas para enfrentamientos desafiantes.
- **Audiencia interesada en el estilo artístico único:** Personas atraídas por el arte visual distintivo, el diseño de personajes y escenarios que ofrecen algo diferente y llamativo.
- **Aventureros de mundo abierto:** Jugadores que disfrutan explorando mundos extensos y descubriendo secretos mientras avanzan en la trama del juego.
- **Exploradores coleccionistas:** Estos jugadores se deleitan descubriendo y recolectando elementos ocultos, secretos o coleccionables dispersos a lo largo del mundo del juego.
- **Público general de videojuegos:** Este segmento, aunque amplio, puede ser atraído por una narrativa atractiva, mecánicas de juego interesantes y un estilo visual único.

Estos perfiles demográficos se convierten en una guía esencial para el diseño y desarrollo, asegurando que cada aspecto del juego este cuidadosamente adaptado a las expectativas y deseos de los jugadores.

## 2.2. Marco teórico

En esta sección se exploran aspectos fundamentales que van desde la apasionante historia de los videojuegos RPG hasta el detallado proceso de diseño y desarrollo, junto con los elementos técnicos esenciales que dan forma a su funcionamiento, asimismo, se lleva a cabo una revisión exhaustiva de trabajos relacionados que enriquecen y contextualizan el estudio, acompañada de un análisis detallado de los motores gráficos más relevantes utilizados en el desarrollo de videojuegos.

### 2.2.1. Historia de los videojuegos RPG

Antes de profundizar en los detalles de esta sección, es importante comprender un poco de la historia que dio origen a los juegos de rol. Estos encuentran sus orígenes en los juegos de mesa, especialmente con Dungeons and Dragons, presentado en 1974 y reconocido como el pionero de los juegos de rol modernos, mismo que se inspira en un mundo de la fantasía, lo que ha llevado a que los juegos de rol estén estrechamente asociados con este género hasta el día de hoy, Dungeons and Dragons introdujo muchas mecánicas clásicas en los juegos de rol, como la personalización de personajes, el sistema de razas y clases, la progresión por niveles, la adquisición de puntos de experiencia, entre otros [5]. Su influencia en la historia del diseño de juegos ha sido monumental, dejando una huella perdurable en el mundo de los juegos. La Figura 4 muestra el juego de rol citado.



Figura 4 Representación visual del juego Dungeons and Dragons

Los juegos de rol RPG abren las puertas a un universo interactivo donde los jugadores se sumergen en roles o personajes específicos, participando activamente en narrativas ficticias, donde la toma de decisiones de los participantes moldea el desarrollo de la historia, creando una experiencia envolvente y única que se adapta a las elecciones individuales de cada jugador [5–7].

El mundo del RPG se basa en una historia de innovación y creatividad que ha dejado una marca en la industria del entretenimiento, mismos que combinan la fantasía con la participación activa de los jugadores, logrando conservar su atractivo y capacidad para cautivar a todo tipo de audiencias.

## 2.2.2. Diseño y desarrollo de RPG

En esta sección, se explora a fondo los pilares fundamentales que dan forma a estos juegos: la creación y evolución de personajes, sistemas de combate, progresión del jugador, construcción de mundos narrativos, mecánicas únicas y las elecciones del jugador. Además, se aborda el sistema de recompensas que es un aspecto que agrega profundidad y motivación a la experiencia de juego.

Estos videojuegos atraen a los jugadores al mezclar una diversidad de elementos cautivadores, donde los sumergen en mundos envolventes y experiencias interactivas, proporcionando aspectos que pueden llegar desde la personalización de personajes hasta narrativas inmersivas y mecánicas distintivas, los RPG ofrecen una diversidad de características que realzan su jugabilidad.

- **Creación de personajes:** La creación de personajes en los juegos de rol RPG es crucial para sumergir a los jugadores en mundos ficticios, donde se puede personalizar la apariencia, habilidades y atributos de los protagonistas, llegando a tener la libertad de elegir entre diversas clases y razas para definir las características únicas de cada personaje. Este proceso no solo es estético sino que impacta directamente en la experiencia de juego al definir las fortalezas y debilidades del personaje, de esta manera se genera una experiencia más personalizada y satisfactoria para los jugadores [8].
- **Sistema de combate:** Los RPG se distinguen por su sistema de combate en tiempo real, distinto de los sistemas tácticos o por turnos, donde su perspectiva puede ser en primera o tercera persona, ofreciendo diversas formas de jugar [5].
- **Progresión y desarrollo:** La progresión se fundamenta en la adquisición gradual de experiencia, reflejando los logros y acciones del jugador en el crecimiento de su personaje, donde el aumento de niveles es fundamental en este proceso, lo que fortalece las habilidades, destrezas y capacidades del personaje a medida que avanza en la trama.
- **Narrativa y mundo de juego:** La narrativa no sólo guía el juego, sino que también se convierte en un elemento esencial para sumergir al jugador en la experiencia, permitiendo que los mundos del juego se creen meticulosamente, fusionando una estética atractiva con una narrativa profunda y conexiones significativas entre lugares, personajes y eventos [9]. Actualmente la narrativa de los RPG contemporáneos tiende a ser no lineal, brindando a los jugadores la capacidad de tomar decisiones que moldean activamente la dirección y el desarrollo de la historia.
- **Mecánicas específicas:** Las mecánicas específicas constituyen los rasgos distintivos que dotan de singularidad a un RPG, entre ellas se puede destacar la exploración de mundos abiertos que emerge como una función principal permitiendo que los jugadores se adentren en entornos extensos, descubrir secretos ocultos y desenvolverse en paisajes variados y expansivos [10,11], asimismo, permite la interacción con personajes no jugables (NPCs) destacando como un elemento crucial; estos personajes ofrecen información relevante, misiones secundarias e incluso plantean desafíos que enriquecen la experiencia del jugador.

- **Sistema de recompensas:** Las recompensas pueden tomar diversas formas, desde la obtención de valiosos botines y equipamiento mejorado, hasta la adquisición de habilidades adicionales o mejoras permanentes para el personaje, asimismo, se puede incluir premios especiales por la culminación exitosa de misiones o desafíos específicos dentro del juego.

Estos componentes se combinan y generan enriquecimiento sobre la experiencia de juego, lo que proporciona un escenario excepcional para la imaginación y la inmersión del jugador.

### 2.2.3. Aspectos técnicos de los RPG

En esta sección se estudia los aspectos técnicos detrás de los RPG, revelando elementos como: los gráficos, la inteligencia artificial, el diseño de niveles, los sistemas de combate, la optimización y otros aspectos que al combinarse dan vida a estos mundos virtuales.

- **Gráficos y motor de juego:** Los motores modernos posibilitan entornos detallados, realistas y dinámicos, impactando la narrativa y la interacción del jugador, estos no solo proporcionan una calidad visual impresionante, sino que permiten efectos de iluminación avanzados, texturas más realistas y animaciones fluidas, contribuyendo enormemente a la inmersión del jugador en la experiencia del juego [12].
- **Inteligencia artificial (IA):** La IA se utiliza para simular comportamientos humanos, lo que implica que los NPCs puedan reaccionar ante las acciones del jugador, tomar decisiones contextualizadas y adaptarse al entorno cambiante del juego [8,13].
- **Diseño de niveles:** El diseño de niveles implica crear entornos visualmente atractivos que contengan progresión lógica y brinden desafíos bien equilibrados, de igual manera, deben presentar un camino fluido que genere retos estimulantes, pero no abrumadores, lo que permite la exploración y la inmersión sin comprometer el ritmo narrativo. Un diseño efectivo garantiza que los jugadores se sientan desafiados y recompensados, manteniendo un sentido de logro sin generar frustración.
- **Mecánica de combate y animación:** La mecánica de combate debe equilibrar la profundidad estratégica con la accesibilidad, ofreciendo encuentros tácticos que no abrumen a los jugadores y es aquí donde las animaciones desempeñan un papel crucial al aportar realismo y dinamismo a estas interacciones, sumergiendo al jugador en la experiencia [10]. El desarrollo de estos sistemas implica una cuidadosa iteración y refinamiento, combinando mecánicas innovadoras con animaciones cautivadoras para lograr una experiencia de combate emocionante y satisfactoria [14].
- **Optimización y rendimiento:** La optimización y rendimiento es clave en el desarrollo de videojuegos ya que permite mejorar el código y ajustar los recursos para garantizar una experiencia fluida y óptima para los jugadores, donde se busca encontrar un equilibrio entre la calidad gráfica y el rendimiento para asegurar que los juegos funcionen sin problemas en consolas, PC o dispositivos móviles, satisfaciendo así las expectativas de los jugadores en cada plataforma.

- **Banda sonora y sonidos:** Una banda sonora que integre una cuidadosa composición, es capaz de evocar emociones, resaltar momentos épicos y sumergir al jugador en diversas situaciones del juego, mientras que los efectos de sonido que pueden llegar a ser desde simples pasos hasta poderosos hechizos, son los que añaden realismo y eficacia al entorno del juego, haciendo que la experiencia sea más vibrante y emocionante [15,16].
- **Interfaz de usuario (UI) y la experiencia del usuario (UX):** La UI se debe desarrollar con la intención de proporcionar información clara y accesible al jugador, mientras que la UX se centra en la experiencia global del jugador, garantizando una navegación intuitiva y atractiva, se debe brindar retroalimentación y orientación cuando sea necesario con el fin de mejorar la inmersión en el juego.

Como se ha apreciado la literatura de esta sección, los aspectos técnicos abren una brecha fascinante hacia el minucioso mundo de los videojuegos, que muestran la combinación entre el arte y la ciencia, misma que desafía constantemente los límites de la creatividad y la innovación dentro de los RPG.

#### 2.2.4. Trabajos relacionados

Los RPG disponen de una amplia red de títulos icónicos que han sido protagonistas a lo largo de la historia del género, sin embargo, esta sección se enfoca exclusivamente en aquellos juegos que han dejado una huella significativa en la industria, marcando un hito o alcanzando un estatus emblemático en el género, es por ello que se destacan trabajos que convergen con el tema central de este trabajo, con la finalidad de descubrir las diversas influencias que han dado forma a este género y poder comprender cómo la creatividad y la innovación continúan impulsando la evolución actual de los RPG. Seguidamente se exponen trabajos relacionados que integran sistemas de combate estratégico.

**World of Warcraft:** Conocido también como WoW es un videojuego MMORPG creado por Blizzard Entertainment<sup>9</sup>, se lanzó al mercado en 2004 y su propósito es sumergir a los jugadores en el vasto mundo de Azeroth que es un lugar lleno de fantasía, magia y conflictos épicos, dejando una huella significativa en la cultura de los videojuegos y sigue siendo uno de los MMORPG más influyentes y populares hasta la fecha.

Seguidamente se enlistan temas de interés con respecto a este gran título, con la intención de proporcionar una visión detallada y completa de la vasta experiencia que ofrece World of Warcraft.

- **Historia del mundo de Warcraft:** Sumerge al jugador en una fascinante línea temporal de Warcraft, explorando sus orígenes, eventos cruciales, la evolución de las distintas razas y facciones, y cómo estas han progresado a lo largo de las expansiones del juego.
- **Razas y clases:** Dispone de varias razas jugables, mismas que se caracterizan por sus aspectos únicos como habilidades raciales y trasfondos culturales, del mismo modo, presenta una amplia variedad de clases que se adaptan a cada una de las razas.

---

<sup>9</sup> <https://www.blizzard.com/>

- **Mapas y regiones:** Este videojuego dispone de continentes y territorios en Azeroth, que van desde los exuberantes bosques hasta desoladas tierras, donde cada región posee su propia historia y desafíos únicos.
- **Mazmorras y raids:** Son elementos fundamentales en Warcraft que ofrecen a los jugadores desafíos grupales y se destacan por sus desafiantes jefes, mecánicas de juego y las recompensas que aguardan a los valientes aventureros.
- **Expansiones:** La riqueza del videojuego se encuentra organizada por expansiones, cada una presentando nuevos mundos, razas, mazmorras y raids, lo que permite seguir ampliando el universo de Warcraft con emocionantes contenidos.
- **PvP y PvE:** Comprende las diferencias entre el combate jugador contra jugador (PvP *en español se conoce como JcJ*) y jugador contra entorno (PvE).
- **Profesiones:** Son habilidades especializadas que los jugadores pueden aprender para recolectar, crear o mejorar objetos útiles para el personaje o para otros jugadores.
- **Comunidad y eventos especiales:** Estos términos hacen alusión a la dinámica social entre los jugadores y a las actividades planificadas en el juego que promueven la participación y fortalece la comunidad entre los jugadores.
- **Música y ambientación:** El juego cuenta con una increíble banda sonora en la que cada zona, ciudad, mazmorra o evento tiene su propia música, lo que enriquece la atmósfera y la narrativa del entorno. En términos de ambientación, presenta un mundo de fantasía amplio y diverso que incluye desde criaturas míticas hasta razas jugables como humanos, orcos, elfos, enanos, tauren, trolls, entre otras, así como entidades mágicas como dragones, elementales y seres de otros mundos.

El sistema de combate de WoW se destaca por su dinamismo, estrategia y fundamento en habilidades clave. A continuación, se presenta un resumen detallado de su respectivo funcionamiento:

- **Selección de objetivos:** Los jugadores seleccionan objetivos haciendo clic en ellos o utilizando atajos de teclado, se puede realizar ataques a enemigos individuales o en grupo, dependiendo de la situación.
- **Habilidades y hechizos:** Cada clase tiene un conjunto único de habilidades y hechizos que pueden utilizar durante el combate, mismas que pueden ser de ataque, defensa, curación, control de masas o utilidad.
- **Rotación de habilidades:** Se refiere al orden en que los jugadores utilizan sus habilidades durante el combate para maximizar el daño, la supervivencia o el apoyo al grupo, aspecto que varía según la clase y la especialización que el jugador haya elegido.
- **Recursos:** Cada clase tiene su propio recurso para usar habilidades, es decir, los guerreros utilizan ira, los magos maná, los cazadores enfoque, etc. Cada uno de estos recursos es crucial para optimizar el rendimiento durante el combate.
- **Movimiento y posicionamiento:** El movimiento es importante en el combate, donde algunas habilidades requieren que el jugador esté en una posición específica o que evite ciertos

ataques al moverse de manera estratégica, asimismo, algunos jefes en mazmorras o raids tienen poderes que los jugadores deben evitar moviéndose o posicionándose correctamente.

- **Combate PvE y PvP:** En combate PvE, los jugadores luchan contra monstruos controlados por el juego, mientras que en combate PvP se enfrentan a otros jugadores, las tácticas y estrategias pueden variar significativamente en cada tipo de combate.

El sistema de combate de WoW es dinámico y requiere tanto habilidad como estrategia, por ello se requiere una correcta coordinación entre habilidades, la adaptabilidad a diferentes situaciones y la comprensión de las fortalezas y debilidades de la clase elegida. Seguidamente, la Figura 5 presenta un fragmento de interacción con respecto al combate de World of Warcraft.



Figura 5 Presentación visual de la jugabilidad en World of Warcraft

**Genshin Impact:** Es un videojuego de acción de mundo abierto desarrollado por MiHoYo<sup>10</sup> y se lanza al mercado en septiembre de 2020, ofrece a los jugadores la oportunidad de explorar un vasto y detallado mundo de fantasía llamado Teyvat, donde la magia y los elementos son fundamentales en la trama y jugabilidad. Los jugadores pueden explorar ciudades, pueblos, paisajes naturales y diversos, y mazmorras intrigantes mientras desentrañan secretos y enfrentan desafíos.

Enseguida se enumeran temas de interés con respecto a este videojuego, con el propósito de ofrecer una visión exhaustiva y amplia de la extensa gama de vivencias que proporciona Genshin Impact.

- **Historia del mundo de Genshin Impact:** La trama se centra en un personaje conocido como el Viajero que está en busca de su hermano perdido en Teyvat que es un mundo lleno de maravillas y peligros.

<sup>10</sup> <https://genshin.hoyoverse.com/es/>

- **Elementos y magia:** El juego presenta siete elementos: Anemo, Geo, Electro, Dendro, Hydro, Pyro y Cryo, (viento, tierra, relámpago, naturaleza, agua, fuego y hielo respectivamente) los cuales impactan tanto en el mundo como en los combates, poder dominar estos elementos permite al jugador combinarlos con sus habilidades, desbloqueando así nuevas formas de interactuar con el entorno y los enemigos.
- **Combate estratégico en tiempo real:** El juego cuenta con un sistema de combate dinámico y fluido que permite cambiar entre personajes para aprovechar sus habilidades y elementos únicos, lo que posibilita la creación de poderosos combos y la desencadenación de reacciones elementales devastadoras.
- **Personajes y compañeros:** Existen diversos tipos de personajes únicos, cada uno con su historia, habilidades y estilo de combate distintivos, el jugador puede formar equipos estratégicos, mejora sus habilidades y desbloquea su potencial para afrontar desafíos cada vez mayores.
- **Eventos y actualizaciones constantes:** El juego integra eventos especiales, colaboraciones emocionantes y actualizaciones frecuentes que enriquecen la experiencia de juego, proporcionando nuevos desafíos, personajes, misiones y recompensas.

El sistema de combate en Genshin Impact es dinámico y versátil, lo que permite a los jugadores alternar entre personajes para aprovechar al máximo las habilidades y elementos únicos de cada uno, lo que facilita la interacción entre ellos y genera reacciones elementales. Esta mecánica promueve la experimentación y la creatividad al enfrentar enemigos y jefes, enseguida se presenta un resumen de su funcionamiento.

- **Interacción con objetivos:** Los jugadores interactúan con objetivos haciendo clic en ellos o utilizando un sistema de apuntado, se puede realizar ataques a enemigos individuales, en grupo o enfocarse en otros elementos del entorno para resolver puzzles y desafíos.
- **Habilidades y hechizos:** Cada personaje tiene habilidades elementales y habilidades definitivas (Ultimates), cada una con su tiempo de reutilización.
- **Rotaciones de habilidades:** Los jugadores planean y ejecutan secuencias de habilidades y ataques para maximizar el daño o aprovechar las debilidades del enemigo. Esto se logra mediante el uso estratégico de habilidades elementales que generan reacciones elementales.
- **Rotación de personajes:** La capacidad de cambiar entre personajes en tiempo real permite aprovechar las habilidades elementales únicas de cada uno, creando combos y reacciones elementales para obtener una ventaja táctica.
- **Recursos:** Cada personaje tiene un recurso asociado a sus habilidades, como energía elemental o puntos de habilidad que se consumen al utilizar habilidades especiales y se regeneran con el tiempo o mediante objetos.
- **Movimiento y posicionamiento:** Los jugadores deben moverse estratégicamente para evitar ataques enemigos y posicionar a sus personajes para maximizar el daño.



- **Combate PvE y PvP:** Los jugadores pueden enfrentarse a enemigos controlados por el juego en ciertas áreas específicas, a combatir con otros jugadores. Los enemigos pueden variar desde criaturas comunes hasta jefes desafiantes.

Una de las características más distintivas de Genshin Impact es su sistema de combate en tiempo real, donde los jugadores pueden alternar entre varios personajes, cada uno con habilidades únicas y elementos asociados, como fuego, agua, tierra, viento, electro y hielo. Estos elementos pueden combinarse estratégicamente para desatar potentes habilidades y ataques. A continuación, en la Figura 6 se presenta la interacción del sistema de combate de Genshin Impact.



Figura 6 Presentación visual de la jugabilidad en Genshin Impact

**Elden Ring:** Es un videojuego de acción y rol RPG desarrollado por FromSoftware<sup>11</sup> en colaboración con George R.R. Martin, reconocido autor de la serie de novelas 'Canción de hielo y fuego' (Game of Thrones). Esta asociación ha despertado un gran interés entre los aficionados a los videojuegos y la literatura fantástica, el videojuego se lanza al mercado en el año de 2022, mismo que permite al jugador la capacidad de explorar y además posibilita el descubrimiento de secretos ocultos que hacen de Elden Ring una aventura envolvente y llena de posibilidades.

Seguidamente se enlistan temas de interés con respecto a este gran título, con la intención de proporcionar una visión detallada y completa de la vasta experiencia que ofrece Elden Ring.

- **Historia del mundo de Elden Ring:** Elden Ring se ambienta en las Tierras Intermedias, un continente ficticio bajo el dominio de varios semidioses, cuyo pasado era regido por la Reina

<sup>11</sup> <https://www.fromsoftware.jp/jp/>

Marika la Eterna, quien actuaba como la guardiana del Círculo de Elden, una formidable fuerza que encarnaba el concepto físico del orden.

- **Criaturas y enemigos misteriosos:** El jugador se enfrenta a una variedad de criaturas y enemigos únicos que deambulan por los extensos territorios de Elden Ring, mismos que van desde bestias imponentes hasta seres místicos que disponen de habilidades y estrategias únicas.
- **Personalización y progresión de personajes:** Cuenta con un sistema de personalización que permite a los jugadores adaptar sus personajes a estilos de juego preferidos. Esto incluye: selección de equipamiento, mejora de habilidades y capacidad de forjar alianzas estratégicas.
- **Armas y magia únicas:** Dispone de una amplia gama de armas y magia, cada una con su propio estilo de combate y habilidades especiales, lo que además permite que el jugador pueda dominar el arte de la espada y el uso estratégico de objetos poderosos.

Esta sección ofrece información detallada sobre las tácticas de combate disponibles en Elden Ring. Se destaca lo más relevante de estas mecánicas para evitar redundancias con respecto a otros trabajos anteriores sobre el tema.

- **Interacción con objetivos:** Se puede apuntar a un objetivo simplemente moviendo la cámara hacia él, o preferiblemente el juego integra un sistema de bloqueo de cámara que permite la interacción mucho más dinámica sin necesidad de estar siempre en posición del objetivo.
- **Sistema de bloqueo, parada y esquivar:** Los jugadores necesitan sincronizar sus acciones para evitar daños y esto se realiza bloqueando o esquivando los ataques del enemigo en el momento exacto, también, se puede ejecutar paradas precisas, lo que abre grandes oportunidades para contraatacar. Este sistema resulta fundamental para la supervivencia en los enfrentamientos más complicados.
- **Recursos:** Un recurso primordial del juego es el vigor, misma que se agota al esquivar, bloquear, atacar, realizar ataques, correr y nadar, y se restablece gradualmente con el paso del tiempo, por otro lado, las habilidades especiales demandan energía mágica u otro recurso específico, y pueden resultar devastadoras si se emplean de manera efectiva.
- **Sistema de habilidades:** Ofrece una variedad que va desde habilidades de combate hasta destrezas de supervivencia, magia y exploración y cada habilidad se divide en varios niveles donde el jugador puede incrementarlos al subir de nivel.
- **Sistema de reputación e influencia:** La reputación del jugador se construye a partir de sus decisiones durante la aventura, influenciando la manera en que los personajes no jugables (NPC) reaccionan y se relacionan con él personaje del jugador.
- **Estilo de combate a distancia, cuerpo a cuerpo, mágico y en montura**
  - **Distancia:** Se basa en emplear arcos, ballestas y conjuros que alcanzan objetivos a distancia, lo que resulta óptimo para aquellos que eligen atacar desde lejos, procurando mantener una distancia segura entre ellos y sus oponentes.

- **Cuerpo a cuerpo:** Esta táctica se enfoca en emplear armas cuerpo a cuerpo como espadas, hachas y martillos, que es una elección ideal para aquellos jugadores que buscan una estrategia de combate más directa y decididamente agresiva.
- **Mágico:** Esta táctica se basa en el uso de hechizos y habilidades mágicas para derrotar a los oponentes, perfecta para aquellos que prefieren un enfoque estratégico y versátil en el combate.
- **Montura:** Esta experiencia sin igual, los jugadores pueden liberar toda la fuerza del arma y la montura en intensos y épicos combates desafiantes.

A continuación, se muestra en la Figura 7 la jugabilidad del Elden Ring con el objetivo de complementar lo descrito previamente.



Figura 7 Presentación visual de la jugabilidad en Elden Ring

## 2.2.5. Comparación de trabajos relacionados

En esta sección, se examinan y comparan múltiples enfoques de análisis y estudios previos relacionados con el tema central, dicho de otra manera, este apartado busca identificar: similitudes, diferencias, puntos fuertes y áreas de oportunidad entre las investigaciones anteriores, proporcionando una visión integral y crítica que enriquecerá la comprensión del tema abordado. Por tal motivo, en la Tabla 9 se presenta de manera resumida las comparaciones de los trabajos que se citaron con anterioridad en la sección 2.2.4, con el objetivo de esclarecer las discrepancias, semejanzas y aspectos entre los videojuegos RPG de: World of Warcraft, Genshin Impact y Elden Ring.

<b>Aspecto</b>	<b>World of Warcraft</b>	<b>Genshin Impact</b>	<b>Elden Ring</b>
Sistema de combate	Basado en habilidades y roles, combate estratégico con opciones de magia, cuerpo a cuerpo y distancia.	Combate en tiempo real, con enfoque en habilidades elementales y cambio de personajes en equipo.	Combate centrado en la acción, con variedad de armas y movimientos con énfasis en la esquivas y el uso de habilidades.
Mecánicas	Misiones, mazmorras, JcJ, profesiones, sistema de hermandad y actividades grupales.	Exploración libre, misiones, JcJ, mazmorras cooperativas, sistema de banner para personajes y sistema de rangos de aventura.	Mundo abierto, mazmorras, jefes, sistema de habilidades y magias, sistema de creación de personajes.
Dinámicas	Basado en grupos y roles definidos, con énfasis en la cooperación y estrategia.	Enfoque en el jugador individual con elementos multijugador.	Enfoque individual en un mundo desafiante y de exploración.
Estética	Estilo de fantasía con gráficos coloridos y variados entornos.	Estética anime, con diseño de personajes vibrantes y entornos detallados.	Ambientación de fantasía oscura con gráficos realistas y atmósfera inmersiva.
Jugabilidad	Enfocado en el juego en equipo, estrategia y colaboración.	Combina exploración, acción y roles específicos de personajes.	Aventura individualizada y personal, pero con oportunidades de cooperación.
Habilidades	Basadas en roles, magias, habilidades y talentos específicos de clases.	Cada personaje tiene habilidades elementales y conjuntos de ataques únicos.	Variedad de habilidades basadas en armas, magias y movimientos específicos.

Tabla 9 Resumen comparativo de los trabajos relacionados

La comparación presentada en la Tabla 9 busca destacar la diversidad de posibilidades y estilos en el género de los RPG, por ello no se centra en elegir opción alguna o específica para el desarrollo del trabajo, sino en ofrecer una visión amplia de las múltiples opciones disponibles que brindan una amplia gama de combinaciones de preferencias o aspectos específicos que se ajusten de mejor manera a la planificación y propuesta del proyecto.

## 2.2.6. Motores gráficos para desarrollo de videojuegos

En esta sección se describen los motores gráficos para el desarrollo de videojuegos, mismos que no solo actúan como cimientos técnicos, sino que también permiten la creación de mundos virtuales llenos de detalles, inmersión y creatividad, además, brindan a los desarrolladores los implementos necesarios para dar vida a narraciones épicas, paisajes impresionantes y personajes icónicos, de igual manera los motores facilitan la optimización y la adaptabilidad, permitiendo que los videojuegos se ejecuten en una variedad de plataformas y dispositivos, que posibilita una calidad visual y jugabilidad excepcional. A continuación, se describen los motores gráficos, mismos que, según su grado de popularidad y preferencia de la comunidad, son bien conocidos en el desarrollo de videojuegos 3D.

**Unity:** Es un motor gráfico desarrollado por Unity Technologies<sup>12</sup> y se emplea para crear videojuegos, aplicaciones y experiencias en entornos tanto en 2D como en 3D [12], lo que posibilita la generación de simulaciones para una amplia gama de dispositivos que van desde: consolas, dispositivos móviles, incluyendo aplicaciones para computadoras, realidad virtual y otros dispositivos [17].

Unity se usa de manera profesional para crear videojuegos Indie, además trabaja en conjunto con el lenguaje de programación C-Sharp Script (C#<sup>13</sup>) lo que permite una integración y desarrollo de secuencias de comandos estables [18].

El programa incorpora demos descargables al realizar la instalación, los cuales contienen scripts y modelos predefinidos conocidos por el editor como 'Assets'<sup>14</sup>, estos demos proporcionan una guía y una base fundamental para quienes deseen adentrarse en el desarrollo de videojuegos.

El aprendizaje en Unity se desarrolla como una experiencia gradual y accesible para los desarrolladores, teniendo como punto inicial una interfaz intuitiva del editor, donde se aprecia las diversas herramientas básicas para manipular objetos y crear escenas simples, esto con la ayuda de una gran comunidad activa y recursos educativos como: tutoriales en línea, extensa documentación y foros, que son pilares fundamentales para este proceso de aprendizaje.

Unity dispone de las siguientes ventanas:

- **Ventana del proyecto:** aquí se presentan todos los "Assets" disponibles para la construcción del proyecto.
- **Vista de la escena:** permite la visualización y edición de la escena en curso.
- **Ventana de jerarquía:** muestra la estructura jerárquica, organización y agrupación de los diferentes objetos 3D dentro de la escena.
- **Ventana del inspector:** revela las diversas propiedades del objeto seleccionado, brindando la posibilidad de editar y observar estas características.
- **Barra de herramientas:** ofrece acceso a una amplia gama de componentes y funciones para facilitar el desarrollo en Unity.

---

<sup>12</sup> <https://unity.com/es>

<sup>13</sup> Lenguaje de programación multiparadigma desarrollado y estandarizado por Microsoft

<sup>14</sup> Representación de cualquier elemento que se puede utilizar dentro del proyecto

Seguidamente se muestra en la Figura 8 la herramienta de Unity con sus respectivas interfaces.

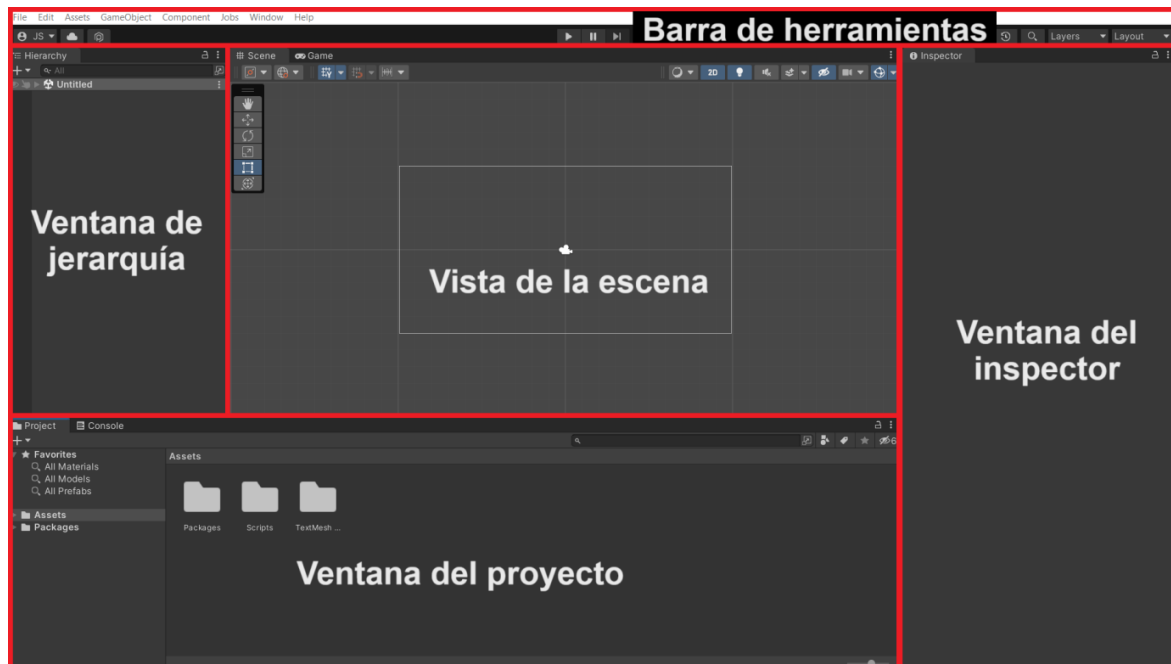


Figura 8 Principales interfaces del motor gráfico de Unity

**Unreal Engine:** Es un conjunto de herramientas y software de desarrollo de videojuegos creado por Epic Games<sup>15</sup> se destaca en la industria del desarrollo y el entretenimiento interactivo, para producir contenido tanto en 2D o 3D, como videojuegos AAA<sup>16</sup>, simulaciones, películas y otros medios [12].

Ofrece una amplia gama de funciones, desde gráficos avanzados y física hasta animación, inteligencia artificial y herramientas de creación de contenido, es por ello que se ha convertido en una opción popular entre desarrolladores de videojuegos y profesionales del entretenimiento [18].

El impacto de Unreal Engine en la industria del entretenimiento va más allá de su éxito en juegos; ha transformado cómo se idean, desarrollan y presentan los videojuegos actualmente, todo esto gracias a su sólida estructura y su capacidad para crear entornos visuales excepcionales, por lo que ha elevado el estándar de calidad, permitiendo a los desarrolladores dar vida a propuestas que integran precisión y realismo, con respecto a la flexibilidad multiplataforma, Unreal Engine ha ampliado enormemente las posibilidades, alcanzando desde consolas y PC hasta dispositivos móviles y realidad virtual [19].

Este motor se conoce por tener una curva de aprendizaje más pronunciada en comparación con otros motores de juegos, principalmente debido a su amplia variedad de herramientas y su enfoque en gráficos avanzados. Utiliza principalmente C++ para su programación principal. Además, cuenta con un sistema visual de programación llamado Blueprints, que permite crear lógica de juego y prototipos sin necesidad de escribir código.

<sup>15</sup> <https://store.epicgames.com/es-ES/>

<sup>16</sup> Conocidos comúnmente como triple A, se denominan así porque los respaldan grandes compañías



Unreal Engine dispone de las siguientes ventanas principales que abarca al desarrollo de videojuegos:

- **Barras de pestaña:** Proporciona acceso a herramientas y comandos para trabajar en niveles superiores del editor.
- **Barra de herramientas:** Ofrece accesos rápidos a diversas herramientas y operaciones de uso común.
- **Actores:** Agrega modos de edición especializados al flujo de trabajo.
- **ViewPoints (Puntos de vista):** Muestra visualmente el mundo virtual en desarrollo.
- **Navegador de contenido:** Organiza y muestra todos los contenidos agregados o colocados en el proyecto.
- **Esquema mundial:** Presenta una estructura jerárquica de todos los objetos en la escena, permitiendo selección y modificación directa.
- **Detalles:** Permite ver y editar materiales utilizados en los objetos del usuario, además de facilitar la edición de los propios objetos.

A continuación, se exhibe en la Figura 9 la herramienta de Unreal con sus interfaces correspondientes.



Figura 9 Principales interfaces del motor gráfico de Unreal Engine

En resumen, los motores gráficos de Unity y Unreal han sido fundamentales en la industria del desarrollo de videojuegos. No solo son herramientas, sino que también han estimulado la creatividad y han difundido la creación de juegos, sentando las bases para una innovación constante en el entretenimiento. Estos motores siguen inspirando la imaginación y promoviendo la excelencia, preparando el terreno para futuras experiencias digitales cautivadoras y emocionantes.

## 2.2.7. Comparación de motores gráficos para desarrollo de videojuegos

En esta sección, se lleva a cabo una detallada comparación entre dos de los motores gráficos más relevantes en la industria del desarrollo de videojuegos: Unity y Unreal Engine, lo que permite explorar sus características distintivas, capacidades, flexibilidad, comunidad, etc. Tal como se aprecia en la Tabla 10, con el objetivo de elegir el motor más apropiado para el desarrollo del presente proyecto.

Características	Unity	Unreal Engine
Lenguaje de Programación	C-Sharp Script (C#)	C++ y Blueprints
Curva de Aprendizaje	Más rápida, más amigable para principiantes.	Más pronunciada, requiere más experiencia.
Precio	Gratuito para ciertos ingresos, planes de suscripción.	Gratuito para uso personal, regalías por ventas.
Ecosistema de la Comunidad	Amplia comunidad de usuarios y documentación.	Comunidad sólida con abundantes recursos.
Versatilidad de Plataformas	Soporte para múltiples plataformas, incluyendo móviles, consolas y PC.	Versatilidad para múltiples plataformas.
Gráficos y Renderizado	Gráficos de alta calidad, renderizado adaptable.	Gráficos avanzados, renderizado impresionante.
Herramientas Visuales	Funcionalidades visuales sólidas con el sistema de nodos.	Blueprints para la lógica visual, potentes herramientas editoriales.
Flexibilidad	Flexibilidad en términos de personalización y adaptación.	Mayor control y flexibilidad para proyectos complejos.
Optimización	Menos control en términos de optimización de rendimiento.	Más control sobre la optimización de rendimiento.
Documentación y Soporte	Amigable y accesible, abundante documentación y tutoriales.	Documentación extensa, soporte técnico robusto.
Tiempo de Desarrollo	Desarrollo más rápido, especialmente para prototipos.	Desarrollo más lento, pero con mayor control y rendimiento.
Uso en la Industria	Ampliamente utilizado en juegos móviles, indie y AR/VR.	Amplio uso en juegos AAA y proyectos de alto rendimiento.

Tabla 10 Comparación de motores gráficos para el desarrollo de videojuegos.

Tras analizar detalladamente la Tabla 10 y considerando las circunstancias actuales del proyecto junto con el tiempo estipulado, se ha tomado la decisión de optar por el motor gráfico de Unity. Esta elección se basa en varios factores como; su curva de aprendizaje, tiempo de desarrollo requerido para realizar el producto, además porque se alinea perfectamente con mi gusto por el lenguaje de programación.



## 3.Propuesta

El objetivo del presente trabajo es desarrollar un videojuego de rol RPG en tercera persona que se distinga por sus clases de personajes y habilidades únicas, utilizando el motor gráfico de Unity 3D, para la construcción del proyecto se plantea como punto clave la creación de un sistema de combate estratégico, mismo que se fundamenta en la integración de habilidades y estilos de combates únicos para cada clase de personaje, ya sea jugable o no jugable. El proceso a desarrollar debe ser capaz de ajustarse para la integración y creación de nuevos personajes y habilidades, por tal motivo se plantea utilizar código eficiente, es decir, emplear POO<sup>17</sup> para elaborar un videojuego escalable. De igual manera, se propone diseñar e integrar un sistema que permita jugador progresar por medio de niveles, los cuales se superan acumulando experiencia al derrotar enemigos.

Para abordar la integración de una amplia gama de habilidades, se propone utilizar Scriptable Objects<sup>18</sup>, mismos que actúan como contenedores de información detallada sobre cada habilidad, incluyendo el nombre, tipo y cualquier otro dato relevante para el correcto funcionamiento, permitiendo una fácil modificación y creación de nuevas habilidades durante el desarrollo del proyecto. Esto además proporciona una flexibilidad significativa en el diseño del sistema, debido a que se separa los datos de las habilidades de la lógica del código, evitando la modificación directa del mismo.

En conclusión, el videojuego a desarrollar adopta un estilo Stylized<sup>19</sup> inspirado en un mundo de fantasía, donde se busca representar el sistema de combate de una manera visualmente impactante.

### 3.1. Problemática

La dificultad principal a abordar reside en encontrar enfoques eficientes y precisos en el diseño y desarrollo de juegos RPG con sistemas de combate estratégico. Esto implica asegurar la estabilidad a largo plazo del proyecto, permitiendo la adición continua y efectiva de nuevas funcionalidades y contenido para enriquecer la experiencia del jugador.

### 3.2. Definición de objetivos/especificaciones del producto

En esta sección se desglosan aspectos cruciales del proyecto que conforman la propuesta central a desarrollar, que van desde el diseño de niveles, personajes, habilidades, enemigos, los requisitos funcionales y no funcionales, donde se detallan metas específicas que se buscan lograr a lo largo del proceso de creación. El producto final debe cumplir con estándares de funcionalidad, rendimiento y experiencia de usuario, lo que proporciona una estructura ordenada para el proceso de creación, garantizando que se cubran las expectativas y necesidades del proyecto.

---

<sup>17</sup> Programación Orientada a Objetos

<sup>18</sup> Clase que permite almacenar datos, independiente de las instancias de scripts

<sup>19</sup> Estilizado, es un diseño que se destaca y enfatiza por un estilo artístico

### 3.2.1. Plan de niveles

Se propone un nivel centrado en una pequeña aldea que se sitúa en medio de las montañas, el entorno presenta una combinación exuberante de vegetación, como árboles frondosos y coloridos pastizales, se disponen senderos que guíen al jugador a lo largo de la travesía, asimismo, se plantea incorporar grandes rocas que enriquecen el paisaje, y como se ha mencionado con anterioridad el paisaje presenta un estilo Stylized, Tal como se aprecia en la Figura 10.



Figura 10 Propuesta del diseño de nivel a integrar en el proyecto.

### 3.2.2. Plan de personajes jugables

La propuesta contempla la creación de una variedad de personajes humanoides organizados en distintas clases, lo que permite la presencia de una amplia gama de tipos, tales como:

- **Guerrero:** es un personaje distintivo que emplea armas de una o dos manos, que en combinación con una reserva de ira como recurso principal puede lograr desencadenar poderosas habilidades, su estilo de combate se inclina hacia la agresividad, especialmente cuando se enfrenta a enemigos en solitario. El guerrero es una clase que destaca por su capacidad de asestar golpes poderosos y su enfoque en el combate directo.
- **Mago:** es un personaje que emplea bastones como armas, canaliza poder a través del mana, que es su recurso principal para desatar habilidades de rango, este personaje se destaca por su dominio de las artes de la magia y su enfoque a distancia, es capaz de infligir daño a múltiples enemigos, pero su resistencia en el combate cuerpo a cuerpo es frágil, por lo que debe estar constantemente fuera del alcance del enemigo.
- **Cazador:** es un personaje conocido por su dominio en el arco, un arma de gran alcance que define su estilo de combate, su recurso principal es la estamina, que utiliza para desencadenar potentes proyectiles capaces de impactar a sus objetivos con efectos variados, se destaca por su versatilidad, lo que permite adaptarse a diversas situaciones, es decir, aprovecha la

naturaleza y su entorno como principal medio para emplear trampas o tácticas evasivas. Con respecto al combate directo, no es tan frágil y siempre es capaz evadir esta situación y poder estar al rango de su objetivo.

- **Curandero:** es un personaje especializado en el arte de la magia curativa y mística y se vale de bastones como su principal arma, utiliza energía espiritual como recurso principal para enfocar habilidades de curación. Su estilo de combate se centra en priorizar la supervivencia y el bienestar de sus aliados durante el combate, convirtiéndolo en un apoyo indispensable con la capacidad de sanar y proteger mientras ocasionalmente, inflige daño a los adversarios.

La concepción de estos variados personajes se fundamenta en enriquecer la experiencia de juego, ofreciendo a los jugadores una gama diversa de estilos y estrategias. Estos personajes están destinados a ofrecer una experiencia de juego dinámica y equilibrada, brindando opciones atractivas para distintos estilos de juego y facilitando una mayor inmersión en el mundo del juego, propiciando así una experiencia envolvente y memorable para los usuarios. A continuación, la Figura 11 muestra el estilo de personajes que se integran al proyecto.



Figura 11 Propuesta de personajes jugables a integrar en el proyecto

### 3.2.3. Plan de habilidades

Se plantea un sistema de habilidades único para cada clase de personaje, abarcando principalmente beneficios, daño y curación. Estas habilidades se clasifican en tres tipos principales: habilidades directas, de proyectil y AoE (área de efecto, ya sea en forma de cono o círculo). Cabe destacar que la naturaleza de estas habilidades puede ser canalizada o no, según su utilidad.

Las habilidades de tipo proyectil incluyen tanto disparos físicos como habilidades mágicas, mientras que las habilidades directas comprenden golpes o efectos de explosión activados de inmediato. Por otro lado, las habilidades AoE pueden ser tanto ancladas al personaje como aplicadas directamente al terreno. El objetivo es proporcionar una amplia variedad de mecánicas de combate durante los enfrentamientos con los enemigos, permitiendo a los jugadores explorar diferentes estrategias y

tácticas de juego dentro del contexto de sus respectivas clases. Seguidamente, la Figura 12 muestra el arte conceptual de las habilidades a integrar en el proyecto.



Figura 12 Propuesta de habilidades a integrar en el proyecto.

### 3.2.4. Plan para tipos de enemigos

En el juego se encuentran una variedad de enemigos que están dispersos en el entorno, desde criaturas más comunes hasta enemigos poderosos. Estos últimos requieren un enfoque estratégico específico para enfrentarlos con éxito, ya sea mediante la interacción con elementos del entorno, el uso de habilidades específicas o la implementación de tácticas especiales.

Finalmente, se presenta la existencia de un gran jefe, que es una criatura formidable que representa un gran desafío, lo que requiere un profundo conocimiento de su estilo de combate, la coordinación estratégica y la aplicación efectiva de habilidades para superar sus distintas fases o mecánicas únicas. Derrotar a estos jefes proporciona recompensas significativas y marca hitos importantes en la progresión del juego. La Figura 13 muestra los enemigos a integrar al proyecto.



Figura 13 Propuesta de enemigos a integrar en el proyecto.

La diversidad de enemigos es crucial para mantener la emoción y desafío en el juego, lo que se planifica la creación de enemigos variados, cada uno con comportamientos y habilidades únicas.

### 3.2.5. Requisitos funcionales

En función al problema y el análisis de los trabajos relacionados, se especifican los siguientes requerimientos funcionales, mismos que se plantean a continuación:

- **Sistema de combate:**
  - **RF-01:** Implementar un sistema de combate estratégico que permita la interacción del jugador con diferentes habilidades y movimientos.
  - **RF-02:** Desarrollar una variedad de habilidades únicas para cada clase de personaje ya sea jugable o no, lo que permite la diversificación de estrategias de combate.
  - **RF-03:** Permitir la combinación de habilidades con el propósito de generar combos y efectos especiales durante los enfrentamientos.
- **Progresión del Personaje:**
  - **RF4:** Crear un sistema de niveles y experiencia que permita al jugador avanzar y mejorar a medida que avanza en el juego.
  - **RF5:** Desarrollar un sistema que recompense al jugador cuando derrote enemigos.
- **Interacción del Jugador:**
  - **RF6:** Diseñar un entorno interactivo que permite al jugador explorar el mundo del juego y pueda participar en múltiples desafíos.
  - **RF7:** Facilitar la interacción del personaje con respecto a los NPCs.

### 3.2.6. Requisitos no funcionales

Se han identificado los siguientes requerimientos no funcionales, listados a continuación:

- **Rendimiento y Optimización**
  - **RNF1:** Garantizar un rendimiento fluido y estable del juego, evitando tiempos de carga prolongados y problemas de framerate<sup>20</sup>.
  - **RNF2:** Optimizar el uso de recursos para garantizar que el juego funcione de manera eficiente en dispositivos de PC.
- **Estilo Visual y Narrativa:**
  - **RNF3:** Desarrollar un estilo artístico y gráfico coherente que refleje la estética y la temática del juego.
  - **RNF4:** Crear una narrativa atractiva que sumerge al jugador en el mundo del juego y complementa la mecánica de combate.
- **Estabilidad:**
  - **RNF5:** Realizar pruebas exhaustivas para garantizar la estabilidad del juego, minimizando errores, cierres inesperados y fallos técnicos.

---

<sup>20</sup> Se expresa como FPS (cuadros por segundo).

## 4. Diseño

La sección de diseño es el centro creativo y conceptual del proyecto, es aquí donde se explican los elementos fundamentales que conforman el mundo del juego: sus personajes, mecánicas y entornos. Es el espacio donde las ideas iniciales cobran vida con detalle, describiendo el proceso de creación de una experiencia de juego cautivadora, además, se analizan las decisiones que dan vida al mundo del juego, combinando la imaginación con la funcionalidad para plasmar una aventura única e inolvidable.

Los puntos clave de esta sección incluyen: una descripción detallada de la herramienta utilizada y los recursos empleados, la arquitectura del proyecto y la funcionalidad del juego.

### 4.1. Herramientas y entorno de desarrollo

Esta sección se enfoca en presentar las plataformas y herramientas fundamentales que se emplean para desarrollar el proyecto, también se detalla el uso de herramientas esenciales como Unity, Visual Studio Code, GitLab y Sourcetree, destacando su papel integral en el proceso de creación y gestión.

- Unity (versión 2021.3.19f1):** Como se menciona en la sección 2.2.7 Comparación de motores gráficos para desarrollo de videojuegos, la elección de este motor responde a preferencias personales, especialmente en el ámbito de la programación. La versión seleccionada de esta herramienta se justifica en función al tiempo dedicado a realizar el presente proyecto, además porque es LTS<sup>21</sup>. Esta elección garantiza estabilidad y no requiere actualizaciones constantes con cada nueva versión. Unity ofrece numerosas ventajas en todas las etapas del desarrollo de un videojuego, desde la construcción de la escena hasta la integración del comportamiento del entorno, convirtiéndolo en el motor ideal, lo que otorga la capacidad de incorporar prácticamente cualquier idea que se haya plasmado dentro de la propuesta. Seguidamente la Tabla 11 describe los requisitos técnicos del entorno de desarrollo

Requisitos técnicos de la herramienta Unity		
	Mínimos	Requeridos
<b>Sistema operativo</b>	Windows 8 (64 bits)	Windows 8, 10, 11 (64 bits)
<b>Procesador</b>	Con soporte para instrucciones SSE2	Con soporte para instrucciones SSE2
<b>Memoria RAM</b>	4 GB de RAM	8 GB de RAM
<b>Espacio en disco</b>	10 GB de espacio en disco	20 GB de espacio en disco
<b>Tarjeta gráfica</b>	Compatible con DX10 (Shader Model 4.0)	Compatible con DX10 (Shader Model 4.0) o superior.

Tabla 11 Requisitos técnicos mínimos y recomendados para que funcione la herramienta Unity

<sup>21</sup> Long Team Support “Soporte a largo plazo”

- **Visual Studio Code:** Permite integrar diversas extensiones que simplifican la construcción del proyecto al trabajar con diferentes frameworks, el control de versiones a través de Git y una amplia selección de paquetes que mejoran la eficiencia del desarrollador. Esta herramienta se utiliza para desarrollar la lógica de programación del videojuego, el cual está construido principalmente con el lenguaje C#. Este lenguaje es fundamental para gestionar las acciones y operaciones dentro del entorno del videojuego en Unity.
- **GitLab:** Es una plataforma completa para la gestión de proyectos de desarrollo de software, lo que permite dar seguimiento de problemas, revisión de código, integración continua y despliegue. El uso principal de esta herramienta en el proyecto es mantener un control exhaustivo sobre el desarrollo en curso, lo que permite tener una visión clara del progreso del trabajo realizado en cada etapa del proyecto.
- **Sourcetree:** Es una aplicación visual y amigable que simplifica el manejo de repositorios Git y Mercurial, además, ofrece una interfaz intuitiva que facilita la visualización y gestión de ramas, confirmaciones (commits) y cambios en el código. SourceTree optimiza el flujo de trabajo y ayuda a mantener un control preciso sobre el código en desarrollo.

## 4.2. Inventario de recursos - Assets

La presente sección detalla los activos y recursos específicos empleados en el proyecto, los cuales han sido adquiridos desde la tienda de Unity y de Humble Bundle. Estos recursos, conocidos como "Assets", representan elementos fundamentales para el desarrollo del videojuego, abarcando desde gráficos y modelos en 3D hasta efectos visuales, scripts y otros componentes clave que enriquecen la construcción del videojuego. En seguida en la Tabla 12 se presentan los recursos que se emplean en el proyecto.

Tipo de recurso	Nombre del recurso	Descripción del recurso
<b>Personajes principales</b>	Modular Fantasy Stylized Human Female	Se utiliza un personaje humanoide femenino para representar las distintas clases disponibles para jugar en el RPG, integra sus propias animaciones.
	Modular Fantasy Stylized Human Male	Se utiliza un personaje humanoide masculino para representar las diferentes clases disponibles para jugar en el RPG, integra sus propias animaciones.
<b>Personajes enemigos</b>	Stylized Fantasy Monkey	Se utiliza un personaje humanoide con características de un mono para representar al jefe final, mismo que dispone de dificultad avanzada, integra sus propias animaciones.

	Stylized Fantasy Sasquatch	Se emplea un personaje humanoidee con características de un gorila como un enemigo de dificultad media, integra sus propias animaciones.
	Stylized Fantasy Mushroom Guardian	Se utiliza un personaje humanoidee con características de un hongo como un enemigo de dificultad baja, integra sus propias animaciones.
<b>Ambiente y escenario</b>	Idyllic Nature Environment	El recurso principal utilizado para recrear el entorno del pueblo con sus vastos pastizales.
<b>Armas y otros recursos 3D</b>	Stylized Newbie Weapons Pack	Conjunto de armas utilizado por los personajes jugables y no jugables.
<b>Efectos visuales</b>	RPG VFX Bundle	Un conjunto de efectos visuales que abarca desde proyectiles hasta efectos mágicos de habilidades.
	Character Auras	Un conjunto de efectos visuales destinado a ser utilizados como auras para los personajes.
	Character Auras 3	Un conjunto de efectos visuales destinado a ser utilizados como auras para los personajes.
	Top-Down Effects	Paquete de efectos visuales que se emplea para que los enemigos apliquen efectos de habilidades de área.
	Sword slashes PRO	Un paquete de efectos visuales diseñado específicamente para representar los impactos de golpes de espada.
	Magic Hits	Este paquete de efectos visuales se usa para asignar impactos o efectos de golpes.
<b>Sonidos y música</b>	Medieval Fantasy Game Sound Effects	Efecto de sonido ambiental, de habilidades y espadas.
<b>Frameworks</b>	RPG Cameras & Controllers	Este recurso es fundamental para desarrollar el sistema de control de personaje y cámara de los jugadores. Generalmente, se utiliza para manejar al personaje controlado por el jugador.
	Scripts internos de NHAvatar	En los recursos 'Modular Fantasy Stylized Human Female' y 'Modular Fantasy Stylized Human Male', se incluye un sistema de gestión de skins y equipamiento de armas que se utiliza para administrar estas características de los personajes.

Tabla 12 Inventario de Assets para la construcción del proyecto.



### 4.3. Enfoque de juego

En esta sección, se ofrece una descripción detallada del contenido del videojuego RPG en el proceso de creación, donde se abarcan aspectos claves que comprenden las mecánicas, dinámicas y la estética del juego, todo ello diseñado acorde la propuesta inicial, los recursos disponibles y el tiempo preestablecido para dar cumplimiento al proyecto. Es importante subrayar que la conceptualización se ha fundamentado en investigaciones previas, dado que los RPG comparten similitudes en la implementación de estos elementos característicos, véase en la sección 2.2.5 Comparación de trabajos relacionados.

#### 4.3.1. Mecánicas, Dinámicas y Estética

Las mecánicas integradas en el videojuego RPG abarcan los siguientes aspectos:

- **Sistema de habilidades:** Los jugadores utilizan habilidades y hechizos exclusivos de las distintas clases de personajes, las cuales están organizadas en un panel conocido como la "Barra de Acción". Esta barra administra las diez habilidades disponibles para cada personaje, mostrando además el tiempo de recarga mediante un sistema de cooldown. Es relevante destacar que estas habilidades comparten un tiempo global de reutilización, es decir, debe transcurrir al menos un segundo para usar otra habilidad. Este tiempo permite que el personaje complete la animación asociada a la habilidad en curso.
- **Sistema de combate:** Los jugadores pueden utilizar habilidades de daño o simplemente el ratón para señalar a los objetivos. Una vez seleccionado, el objetivo se muestra dentro de la "Barra de Estado del Objetivo" que se localiza junto a la "Barra de Estado del Jugador", revelando información como su nombre, salud y recursos. Además, proporciona un enfoque sobre el objetivo actual, el cual debe ser seleccionado previamente para marcarlo como objetivo de combate, en términos más simples, el jugador requiere fijar un objetivo para proceder a emplear mecánicas de combate, asimismo, el sistema de combate responde a como se acomode el personaje del jugador, es decir, no puede atacar ni emplear habilidades si su objetivo esta fuera del alcance o este se encuentra a espaldas del personaje.
- **Progresión de personajes:** El protagonista avanza de nivel y adquiere nuevas habilidades a medida que acumula experiencia, misma que se obtiene al momento de derrotar enemigos, esta información se evidencia desde la "Barra de experiencia".
- **Sistema de recompensas:** El jugador es recompensado al derrotar enemigos, obteniendo ítems de armas progresivamente más poderosas a medida que avanza en el juego. Estos recursos se almacenan en el inventario para luego ser integrados al "Gestor de equipamiento".
- **Sistema de clases:** Una variedad de clases con habilidades distintivas está disponible en el juego, como se ha mencionado anteriormente. Estas clases están predefinidas en el menú de selección, permitiendo ofrecer al jugador una clase ya establecida para que se sumerja directamente en la experiencia de juego.

- **Exploración y misiones:** Se asignan una misión con la finalidad de cumplir los objetivos del juego, esto hace que el usuario explore y se familiarice con el entorno.

La dinámica integrada en el proyecto implica:

- **Jugador contra entorno:** El juego presenta encuentros con enemigos controlados por inteligencia artificial, con una variedad de dificultades, para este caso, se incluyen enemigos humanoides diseñados para desafiar al jugador, fomentando la exploración y el uso estratégico de habilidades, es importante destacar que los enemigos también poseen habilidades propias, lo que aumenta la complejidad y el desafío durante los combates, estas habilidades pueden ser anticipadas por el jugador, permitiéndole esquivarlas y minimizar el daño recibido durante el enfrentamiento, dicho de otro modo, las habilidades de los enemigos, se aprecian con anterioridad sobre el terreno, cabe destacar que los NPCs cuentan con un sistema de reseteo; si el objetivo se aleja demasiado de su punto inicial, se restablece, volviéndose inmune al daño durante esta fase mientras regresa a su posición origen, también estos personajes buscan objetivos a rango, pudiendo atacar a otros NPCs que no concuerden con sus características o simplemente deambular según patrones preestablecidos.

La estética se define por:

- **Ambiente de fantasía:** Un pintoresco pueblo rodeado de montañas, envuelto en una exuberante vegetación y vastos pastizales se lo consigue al emplear recursos artísticos del estilo Stylized, lo que implica definir las dimensiones de la escena con precisión, requiriendo una cuidadosa planificación y un detallado relieve en el terreno, además, se integran recursos 3D fantásticos, como piedras de rituales, frondosos árboles y efectos visuales de ambiente.
- **Diseño de personajes:** Los personajes humanoides representan fielmente la estética y cultura humanas, manteniendo la coherencia con el estilo artístico del entorno, estos avatares están diseñados para realizar acciones que imitan de forma natural las acciones humanas. Son esenciales para brindar una experiencia de juego inmersiva, ya que refuerzan el enfoque del videojuego en el jugador y su interacción con el mundo virtual.

El diseño del personaje está meticulosamente estructurado para portar armas en sus manos, agregando un nivel de realismo y coherencia al juego.

- **Efectos visuales y sonoros:** Los efectos visuales y sonoros presentes en los hechizos, los combates y los entornos proporcionan una experiencia envolvente y coherente al juego. Estos elementos ofrecen una retroalimentación inmediata al jugador, permitiéndole reaccionar y sumergirse completamente en las situaciones del juego. Además, los efectos de sonido tienen un papel crucial, contribuyendo significativamente al desarrollo y la inmersión del juego.

Un aspecto esencial que merece especial atención y detalle son las animaciones de los personajes. Estas deben alinearse con el contexto del juego, ya que las animaciones de los estilos de combate (con y sin armas, como espadas, hachas, bastones o arcos) son vitales para ofrecer una experiencia realista y coherente al usuario. La calidad y cohesión de estas animaciones son cruciales, ya que usar animaciones simples o que no se correspondan con las acciones realizadas pueden disminuir el interés

del jugador en el juego. Por tanto, se plantea la inclusión de animaciones que reflejen fielmente las acciones ejecutadas por el jugador o por el personaje en su entorno, buscando así aumentar la inmersión y la satisfacción del usuario en la experiencia de juego.

Por esta razón, los recursos antes mencionados. Personajes principales y enemigos incluyen animaciones específicas adaptadas a los personajes del proyecto. Esta estrategia elimina la necesidad de recurrir a otros recursos externos, asegurando que las animaciones se ajusten perfectamente a las características y acciones de los personajes, evitando cualquier discrepancia entre ambos elementos.

#### 4.4. Arquitectura general de la aplicación

Esta sección aborda la arquitectura general de la aplicación, donde se detalla el esquema que define las diversas escenas claves en el desarrollo del RPG. Estas escenas constituyen los pilares fundamentales en la experiencia del juego, incluyendo la Escena de Selección de Personaje, que permite al jugador elegir su avatar y definir su participación en la aventura; la Escena de Interfaz de Usuario, donde se gestionan elementos visuales y funcionales, como las barras de acción, inventario y barra de estado del personaje; y la Escena de Juego que es el epicentro de la acción donde el jugador se sumerge y disfruta del RPG, interactuando con personajes no jugables, enemigos y explorando un mundo lleno de desafíos. Tal como se aprecia en la Figura 14.

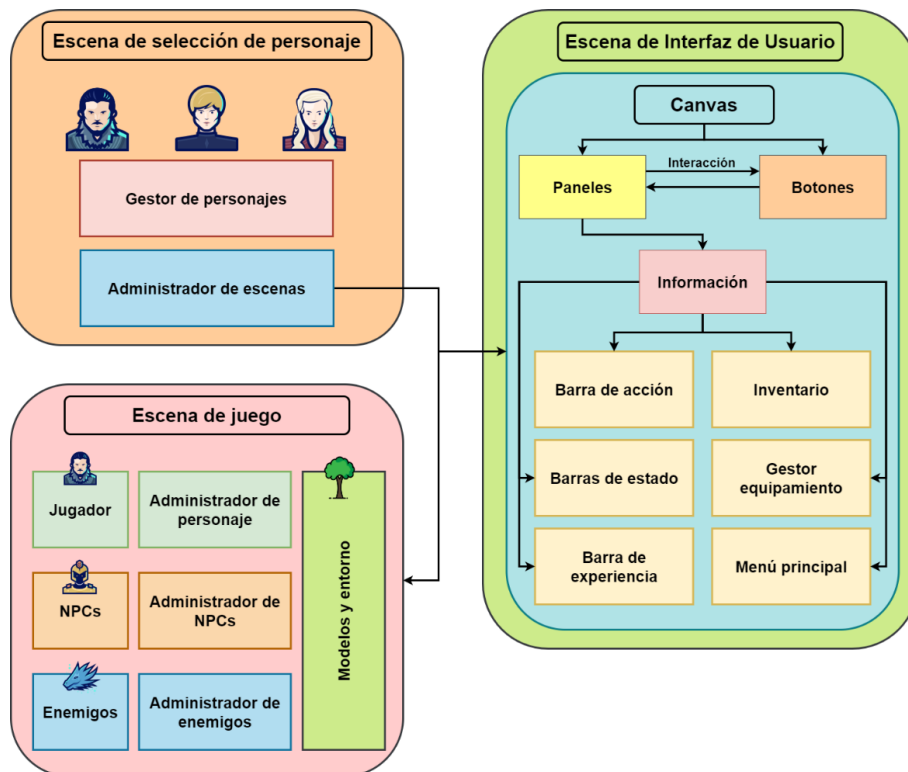


Figura 14 Arquitectura general del videojuego RPG

## 4.5. Arquitectura de la información y diagramas de navegación

En la presente sección, se exploran aspectos claves de la arquitectura de la información y los diagramas de navegación del proyecto. Aquí se analizan tres elementos fundamentales para la estructura del juego, mismas que a continuación se describen a detalle para un mejor entendimiento.

### 4.5.1. Entidad relación sistema de habilidades mediante Scriptable

#### Object

En esta sección, se explora el desarrollo del sistema de habilidades utilizando Scriptable Objects. Aquí se detalla cómo se crean estas habilidades mediante una jerarquía de herencia, además, se profundiza en el diseño y la estructura de estas habilidades mostrando cómo se organiza y crea una amplia gama de poderes, todo esto se lo realiza con el objetivo de comprender mejor los siguientes diagramas que se describen más adelante, por lo que tienden a mencionar esta parte de habilidades mediante datos almacenados en Scriptable Objects. A continuación, se muestra en la Figura 15 el diagrama de entidad-relación del sistema de habilidades, el cual permite crear habilidades de daño y curación que se clasifican en: directo, proyectil, desplazamiento, AoE en área o cono y adicionalmente efectos temporales de beneficio (buff), daño y curación en el tiempo.

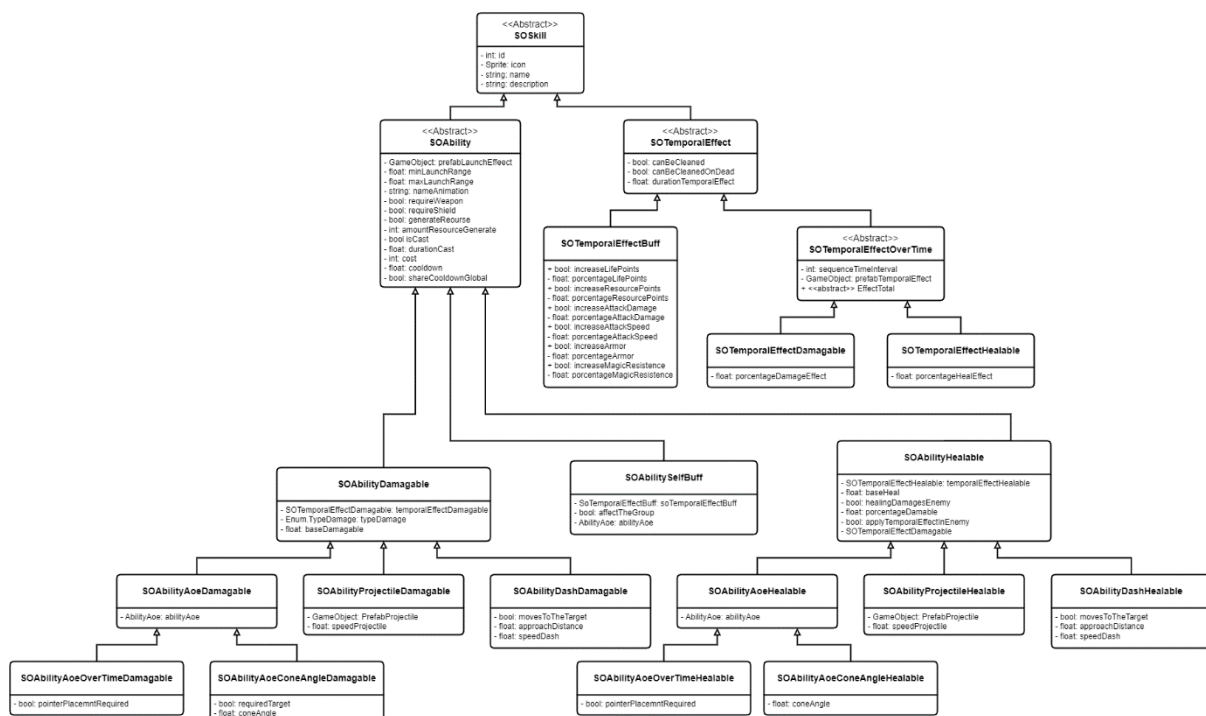


Figura 15 Entidad relación del sistema de creación de habilidades y manejo de datos de los mismos.

La creación de este sistema tiene como objetivo separar la lógica de programación de los datos de habilidades, permitiendo la creación ilimitada de poderes para cada clase existente en el RPG, lo que asegura al mismo tiempo la estabilidad del proyecto al evitar conflictos en el manejo de la información.

### 4.5.2. Entidad relación sistema de combate

La Figura 16 muestra la arquitectura que representa el sistema de combate de los personajes a través del diagrama de clases entidad-relación, bajo el paradigma de Programación Orientada a Objetos. Este diagrama expone la relación y estructura entre las entidades principales del juego a través de elementos fundamentales como herencia, clases abstractas e interfaces. Seguidamente se muestra el diagrama de entidad-relación del funcionamiento del sistema de combate de los personajes.

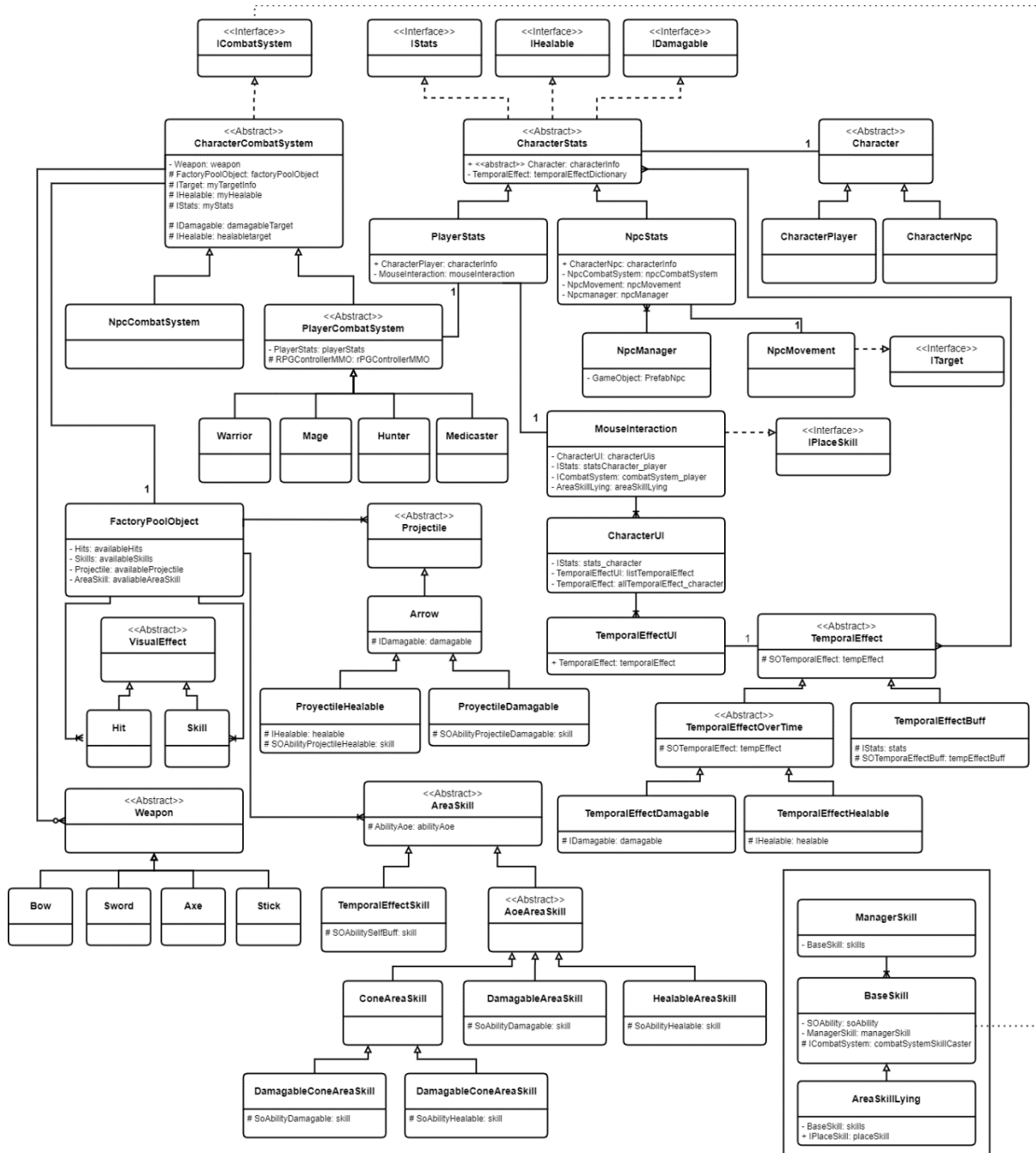


Figura 16 Entidad relación del sistema de combate de personajes jugables y no jugables

La implementación de herencia que refleja la jerarquía y relación entre las distintas clases de personajes, enemigos y elementos del entorno del juego, asimismo, las clases abstractas son empleadas para establecer comportamientos y propiedades comunes, permitiendo una estructura más sólida y coherente entre los elementos del juego. El uso de interfaces facilita la implementación de múltiples comportamientos y funcionalidades, brindando una mayor flexibilidad en la programación y permitiendo que objetos de diferentes clases puedan compartir funcionalidades específicas. El polimorfismo, otra característica clave de la POO, se emplea para dar una misma funcionalidad a diferentes clases, lo que permite tratar objetos de clases distintas de manera uniforme, otorgando versatilidad y eficiencia al sistema. En conjunto, esta arquitectura orientada a objetos busca optimizar la organización, la reutilización de código y la coherencia en el diseño del videojuego, permitiendo una representación eficaz de la relación y comportamiento de sus entidades principales.

### 4.5.3. Árboles de decisiones y máquinas de estado de la IA

Este apartado se centra en el comportamiento de la IA de los personajes no jugables (NPCs) y enemigos, lo cual es fundamental para la jugabilidad y la inmersión en el entorno del RPG. Los NPCs y enemigos se rigen por una arquitectura que define sus decisiones y acciones autónomas en el juego. Cada personaje no jugable tiene asignado un compartimento que se identifica mediante un código único denominado FactionID (Identificador de facción). Este identificador determina si el personaje debe atacar a otros de su misma especie. Además, la existencia de facciones y estados para personajes neutrales es esencial, ya que permite interacciones realistas entre personajes aliados y aquellos que se encuentran en combate a su alrededor. Esto da lugar a la creación de manadas de enemigos, aliados y personajes neutrales, mejorando la experiencia de juego al permitir interacciones específicas con distintos tipos de entidades.

Se debe integrar una IA robusta y realista, capaz de adaptarse al entorno y a otros personajes para evitar inconsistencias, lo cual es crucial ante situaciones como, por ejemplo; cuando el personaje principal muera en combate, en lugar de reiniciar el juego en tales situaciones, la IA ajustará su estado para reflejar la situación, lo que garantiza un comportamiento más coherente y realista. Otro aspecto clave es la interacción entre las diferentes IA, es decir, si una IA elimina a otra, deben poder interactuar entre sí para mejorar la experiencia del usuario.

Aparte de la capacidad que tiene la IA de ejercer un reinicio, esta misma debe tener la libertad de moverse por el entorno, por ello se ha diseñado tres estilos de navegación: 'Estado de Guardia', donde el NPC vigila una zona sin moverse, útil para supervisar áreas clave; 'Estado de Patrulla', donde el NPC monitorea puntos de control de manera aleatoria o secuencial para cubrir un área específica; y 'Estado Aleatorio', que permite al NPC moverse dentro de un área establecida utilizando puntos de control que el mismo define bajo sus propias reglas, asegurando que sean transitable para un comportamiento más dinámico.

Seguidamente en la Figura 17, se describe el árbol de decisiones para la IA de los personajes no jugables dentro del RGP, cabe indicar que se opta por este estilo, debido a que es más fácil de integrar.

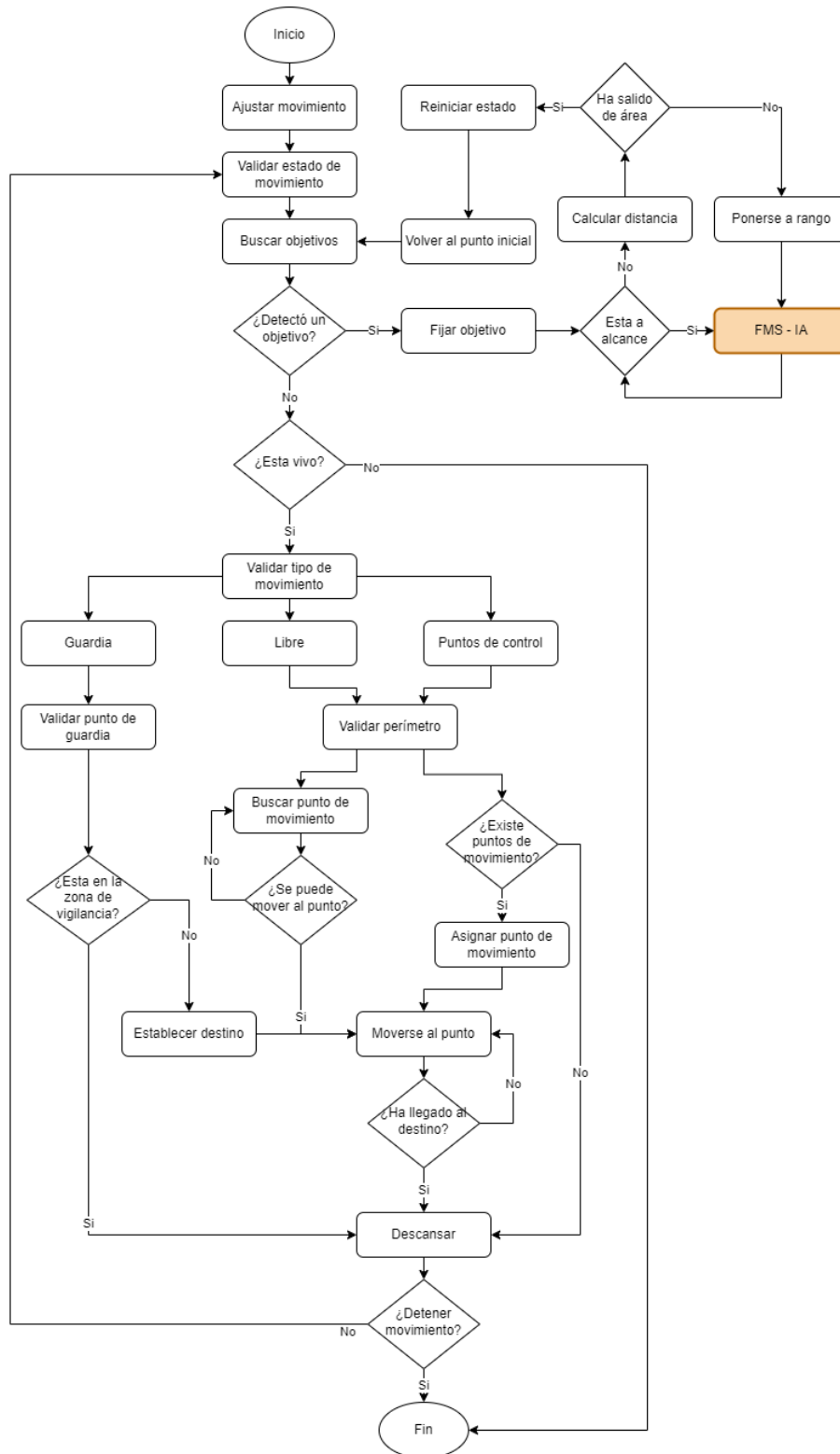


Figura 17 Árbol de decisión de la IA para gestión de movimiento y objetivo

Como se aprecia en la Figura 17 la conducta de la IA está estrechamente ligado al entorno el objetivo y su estado de vida, enseguida se describe la Máquina de Estados Finita (FMS) del sistema del combate.

A lo largo de este trabajo, se ha subrayado la importancia de que los personajes no jugables posean una IA capaz de manejar ataques y habilidades mediante un sistema de estados finitos. Este sistema de combate debe persistir hasta que ocurra una de tres situaciones: la eliminación del objetivo, la pérdida del objetivo o la eliminación de la propia IA. La Figura 18 muestra el Diagrama de Estados Finitos (FMS) del gestor de combate de la IA, el cual prioriza habilidades según la fase y luego procede con habilidades de ataque, considerando la presencia o ausencia de un arma.

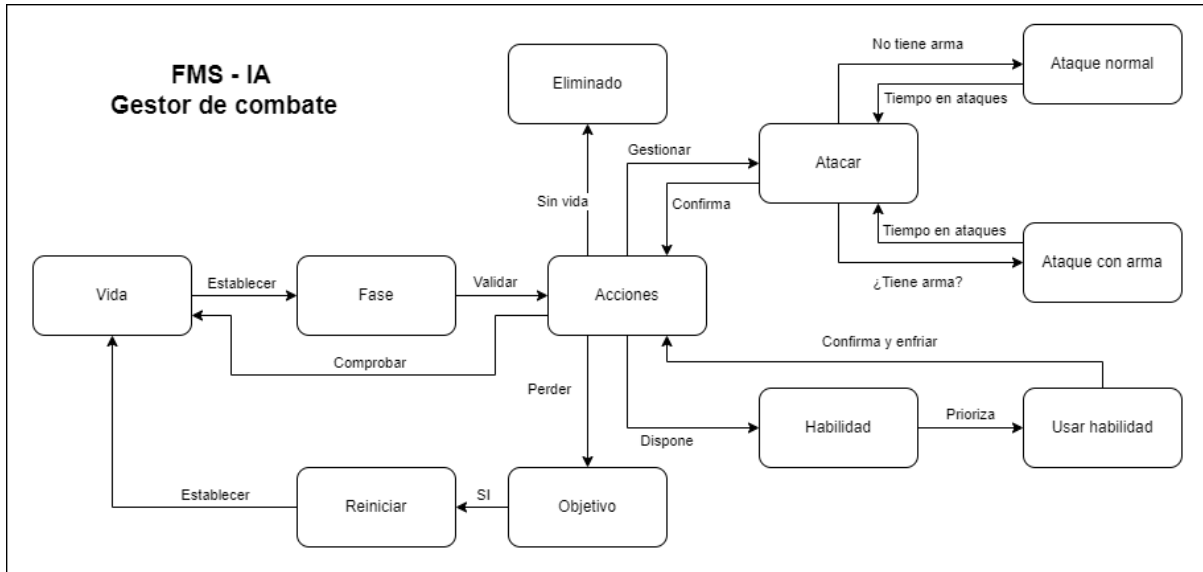


Figura 18 FMS – IA Gestor de combate para personajes no jugables que poseen o no habilidades

Estos análisis y diagramas representan los cimientos estructurales del juego, proporcionando una visión detallada y exhaustiva de la compleja interacción entre el sistema de habilidades mediante Scriptable Objects, el sistema de combate y las intrincadas máquinas de estado y árboles de decisión de la IA. Estas secciones resumen la esencia de la jugabilidad, la profundidad estratégica y la dinámica desafiante que define la experiencia dentro del RPG, sentando las bases técnicas y conceptuales sobre las cuales se construye videojuego.

#### 4.6. Diseño gráfico e interfaces

Este apartado se enfoca en la conceptualización del diseño de la interfaz de usuario asociada al juego de rol RPG en desarrollo, con el propósito de comprender en detalle cómo dicha interfaz influye en la experiencia de navegación durante el juego.

El objetivo del diseño, tal como se evidencia en la Figura 19, es establecer una interfaz de usuario que sea simple y fácil de comprender. Esta estrategia busca primordialmente lograr que el usuario se familiarice de manera inmediata con las funciones del juego, permitiéndole emplear cada elemento de la interfaz de manera efectiva, se busca la implementación de un diseño que sea intuitivo y amigable



para el usuario, enfocándose en priorizar la funcionalidad y la facilidad de uso por encima de la apariencia estética.

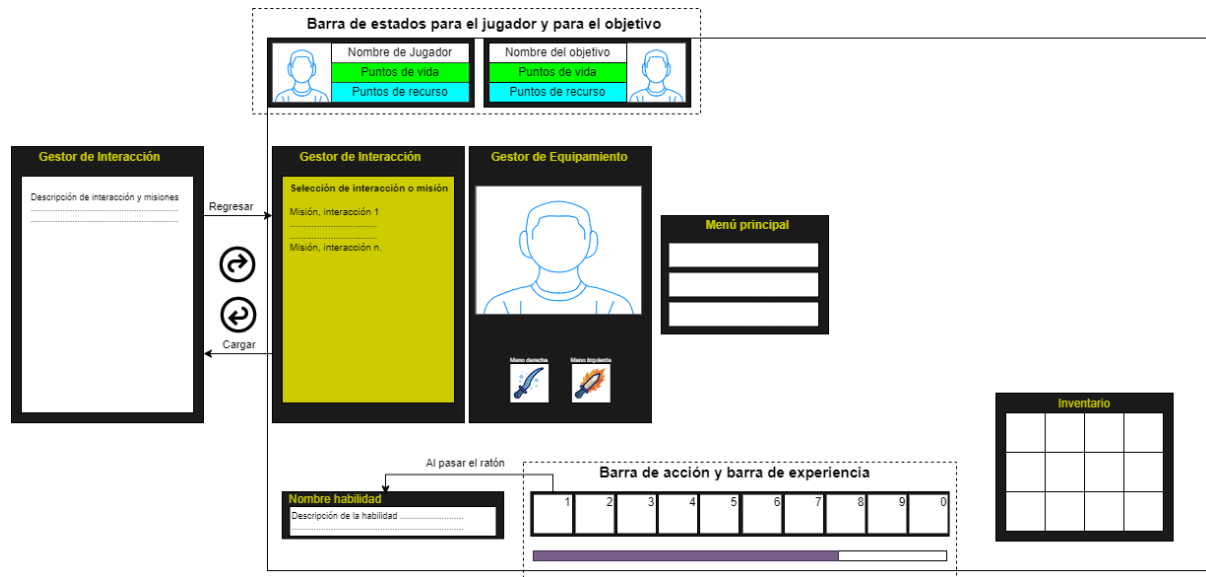


Figura 19 Diseño de la UI que se integra al desarrollo del RPG

Como se aprecia en la Figura 19, la UI diseñada, dispone de los siguientes componentes:

- **Gestor de interacción:** es una interfaz que facilita la comunicación directa entre el jugador y el entorno del juego. A través de esta ventana, el usuario puede interactuar con los NPCs, lo que posibilita iniciar misiones y acceder a otros recursos informativos dentro del juego.
- **Gestor de equipamiento:** está diseñada para brindar a los jugadores la capacidad de cambiar y administrar las armas de sus personajes de manera eficiente dentro del juego. Esta interfaz proporciona una vista clara y accesible del arma actual que dispone el personaje.
- **Barras de estado del jugador y el objetivo:** Esta interfaz ofrece información detallada sobre el estado actual del personaje y el objetivo a través de elementos visuales como el nombre, barras de salud y recurso. La barra de salud refleja el nivel de vitalidad del personaje, mientras que la barra de recurso indica su capacidad para habilidades específicas, como ira, energía o mana, representada por un cambio de color acorde al tipo de recurso utilizado.

Esta disposición visual proporciona una vista clara y rápida del estado actual del personaje y del objetivo, permitiendo a los jugadores entender de manera inmediata la situación. Esta información es crucial para tomar decisiones estratégicas durante el juego, ya que ofrece una comprensión inmediata de los estados actuales, facilitando la toma de acciones acertadas en el momento preciso.

- **Inventario:** permite almacenar recursos, centrándose específicamente en la capacidad de almacenar armas, ya que son los únicos recursos despojables en el juego. A través del inventario, el jugador puede gestionar la colección de armas y equiparlas en el gestor de equipamiento, ofreciendo la ventaja de conservar armas al realizar un cambio en el equipo.
- **Barra de acción:** este espacio alberga las habilidades del personaje limitado a un máximo de 10 espacios disponibles, ofreciendo una representación visual, clara y accesible de las habilidades únicas del personaje. Al posicionar el ratón sobre cada habilidad se despliega una ventana emergente que muestra información detallada correspondiente a la misma, esta característica permite al usuario obtener información adicional sobre cada habilidad de manera rápida y sencilla, facilitando la comprensión de su funcionamiento y uso estratégico durante el juego.

Además de mostrar las habilidades, proporciona una funcionalidad interactiva: al hacer clic en ellas, el usuario puede activar la habilidad correspondiente. Asimismo, visualmente presenta el tiempo de reutilización de cada habilidad, indicando cuándo estará disponible para ser utilizada nuevamente.

- **Barra de experiencia:** esta función refleja la experiencia acumulada durante el juego, con un límite máximo establecido en el nivel 5. La experiencia obtenida se utiliza como moneda de progreso para adquirir nuevas habilidades a lo largo del juego.
- **Menú principal:** sirve como plataforma para diversas acciones, como salir del juego, ajustar los efectos de sonido y cerrar el nivel actual para regresar al menú principal de selección de personaje. Ofrece una serie de funcionalidades que permiten al usuario gestionar la experiencia de juego, desde ajustes técnicos hasta la navegación entre diferentes secciones del juego.

## 5. Implementación

Esta sección, no solo detalla los pasos concretos y los hitos alcanzados en la materialización de este proyecto, sino que también destaca la simbiosis entre la metodología ágil Scrum y la potencia creativa de Unity, donde se describen los desafíos enfrentados, las soluciones encontradas y las lecciones aprendidas a lo largo de esta experiencia que comprende la creación de un videojuego RPG que trascienda las fronteras entre lo imaginario y lo tangible. Seguidamente, a lo largo de este capítulo se explica el desarrollo de cada sprint descrito en la sección 1.4.3 Plan de lanzamiento, con la finalidad de seguir el progreso detallado y la evolución paso a paso del proyecto. Esto permite comprender en profundidad cómo cada sprint contribuye al logro de los objetivos establecidos, así como los ajustes realizados para adaptarse a las necesidades emergentes durante el proceso de desarrollo.

## 5.1. Sprint 1

Para la ejecución del primer sprint, es imperativo realizar una revisión exhaustiva de las tareas asociadas a cada historia de usuario que compone dicho sprint. A continuación, la Tabla 13 exhibe la planificación detallada correspondiente a esta fase inicial del desarrollo del proyecto.

<b>US-01: Programar el movimiento de los personajes jugables</b>	<ul style="list-style-type: none"> <li>• Implementar un recurso (asset) que posibilite el desplazamiento del personaje jugable, al mismo tiempo que gestione la cámara y las distintas colisiones dentro del entorno del juego.</li> </ul>
<b>US-02: Programar el movimiento de los personajes no jugables</b>	<ul style="list-style-type: none"> <li>• Programar el desplazamiento o movimiento de los NPCs mediante la implementación de Nav Mesh Agent.<sup>22</sup></li> <li>• Administrar distintas modalidades de movimientos, como la patrulla, la posición de guardia y movimientos aleatorios. Además, permitir la capacidad de seguir a otros personajes.</li> </ul>
<b>US-03: Animación de personajes jugables y no jugables</b>	<ul style="list-style-type: none"> <li>• Elaborar y ajustar el controlador de animaciones para cada uno de los personajes.</li> <li>• Gestionar de manera eficiente la transición entre animaciones mediante programación.</li> </ul>

Tabla 13 Planificación de tareas del sprint 1

- **US-01: Programar el movimiento de los personajes jugables**

En el marco de la tarea asignada para el primer sprint, se ha decidido explorar una solución que facilite la manipulación y dirección de la cámara principal del jugador, con el propósito de brindar una navegación dinámica y otorgar fluidez en el desarrollo del proyecto en términos de tiempo. En este sentido, el proyecto incorpora el recurso denominado (RPG Cameras & Controllers)<sup>23</sup>, como se ha mencionado dentro de la sección 4.2 Inventario de recursos – Assets, donde se agrega al código base un segmento adicional que habilita el empleo de habilidades de desplazamiento, preservando así la integridad del código original y la responsabilidad asociada a la gestión del movimiento del personaje. Es relevante señalar que este recurso opera a través del nuevo sistema de entrada (new Input system)<sup>24</sup>, y la configuración pertinente ha sido establecida por el autor del recurso. De igual manera, el movimiento de la cámara ha sido desarrollado para permitir la movilidad del personaje, garantizando la coexistencia de ambos aspectos: el desplazamiento del personaje y el movimiento de la cámara, con el objetivo de proporcionar una perspectiva mejorada del entorno mediante una cámara orbitaria. A continuación, la Figura 20 presenta la configuración del movimiento del personaje jugable, detallando sus respectivos controles.

<sup>22</sup> <https://docs.unity3d.com/es/530/Manual/class-NavMeshAgent.html>

<sup>23</sup> <https://assetstore.unity.com/packages/tools/camera/rpg-cameras-controllers-145324>

<sup>24</sup> <https://docs.unity3d.com/Packages/com.unity.inputsystem@1.7/manual/index.html>



Figura 20 Asignación de teclado para el control de movimiento y cámara del personaje jugable

Para asegurar un correcto funcionamiento del control de movimiento del personaje jugable, es esencial que el modelo 3D del personaje tenga adjuntos los siguientes componentes de script, como se ilustra en la Figura 21. Estos componentes son críticos para garantizar la integridad y eficiencia de la funcionalidad de movimiento del personaje, siendo parte integral de la arquitectura que respalda la interactividad del jugador con el entorno del juego. Además, cabe destacar que el diseño y la implementación de estos scripts se han desarrollado específicamente para integrarse sin inconvenientes en el entorno de desarrollo Unity.

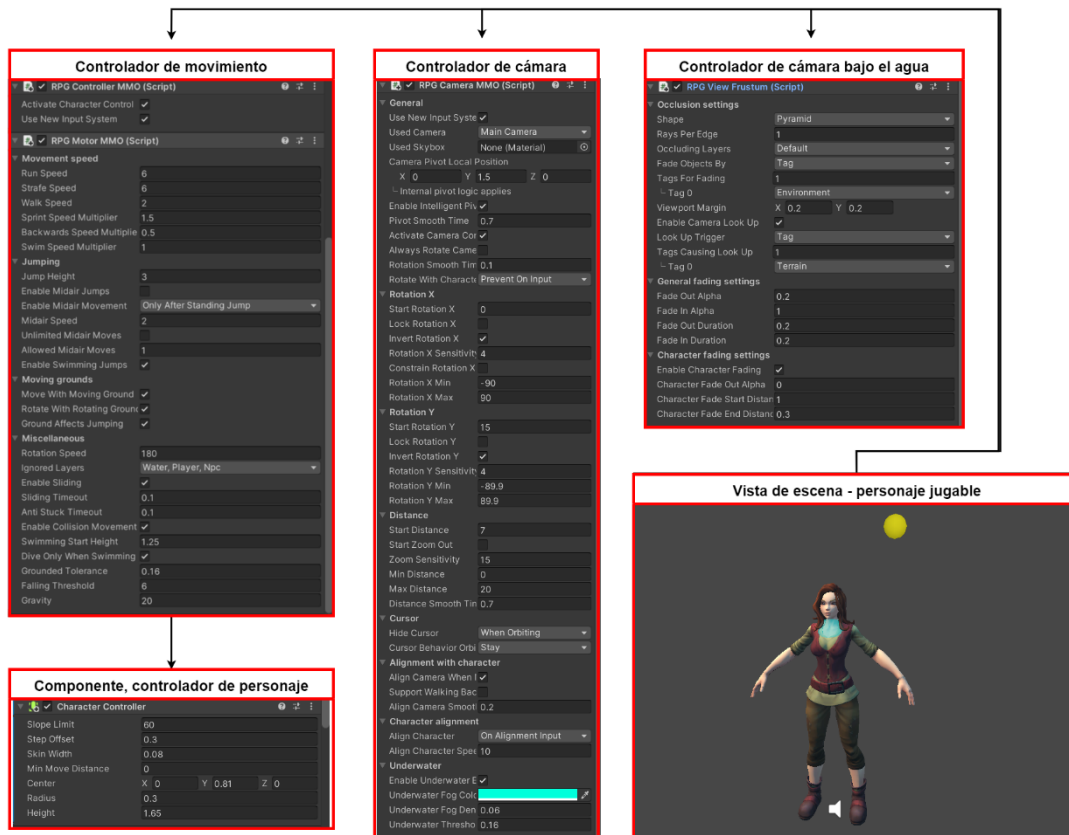


Figura 21 Funcionalidad para el movimiento del personaje jugable

El personaje jugable está equipado con un sofisticado conjunto de controladores que optimizan su experiencia en el entorno del juego, mismos que incluyen:

- **Controlador de Movimiento:** El controlador de movimiento desempeña un papel fundamental al gestionar las acciones y el movimiento fluido del personaje en relación con el terreno y otros recursos 3D, como colisiones. Este componente garantiza una interacción coherente y eficiente del personaje con el entorno, permitiendo una experiencia de juego inmersiva y sin contratiempos.
- **Controlador de Cámara:** El controlador de cámara opera como un elemento clave para ofrecer una perspectiva dinámica y controlada del entorno, donde la cámara se encuentra diseñada para que sea de tipo orbitaria, este controlador permite al jugador mover la cámara alrededor del personaje, proporcionando una visión completa del entorno. Además, facilita el control de la rotación del personaje cuando se mantiene presionado el clic derecho del ratón, ofreciendo flexibilidad y precisión en la exploración del mundo del juego.
- **Controlador de Cámara Bajo el Agua:** Este controlador especializado se encarga de gestionar la cámara cuando el personaje se sumerge en el agua, su principal función es poder crear la sensación inmersiva de estar debajo del agua al controlar la perspectiva y los efectos visuales. Proporciona una experiencia visual única y realista al adaptar dinámicamente la cámara a las condiciones subacuáticas, contribuyendo así a la coherencia y autenticidad del entorno acuático en el juego.

Estos componentes de scripts se encuentran diseñados de tal manera que colaboran armoniosamente para brindar una experiencia de juego integral y cautivadora, asegurando que el jugador pueda explorar y disfrutar del mundo virtual con fluidez y realismo.

- **US-02: Programar el movimiento de los personajes no jugables**

Para gestionar de manera efectiva el movimiento de los NPCs y otras funcionalidades que se integran al entorno del juego, es esencial comenzar con el diseño de un nivel básico. Este enfoque, conocido como programación en caja blanca, permite una optimización eficiente de los recursos al abordar directamente todas las necesidades del desarrollo.

La creación de un nivel básico sirve como punto de partida para la programación, facilitando la identificación y corrección eficiente de posibles errores o efectos no deseados (glitches) que puedan surgir durante el desarrollo. La Figura 22 presenta la escena base diseñada con esta finalidad, marcando el inicio del proceso de revisión y dando paso al desarrollo continuo del proyecto. Este enfoque proactivo garantiza la solidez y calidad del código, contribuyendo a la eficiencia y estabilidad del proyecto en su conjunto, es importante resaltar que el escenario dispone de obstáculos y pendientes con el propósito de integrar las funcionalidades a este tipo situaciones, de esta manera se asegura que toda la lógica a integrar cumpla con las necesidades requeridas dentro de cada sprint.

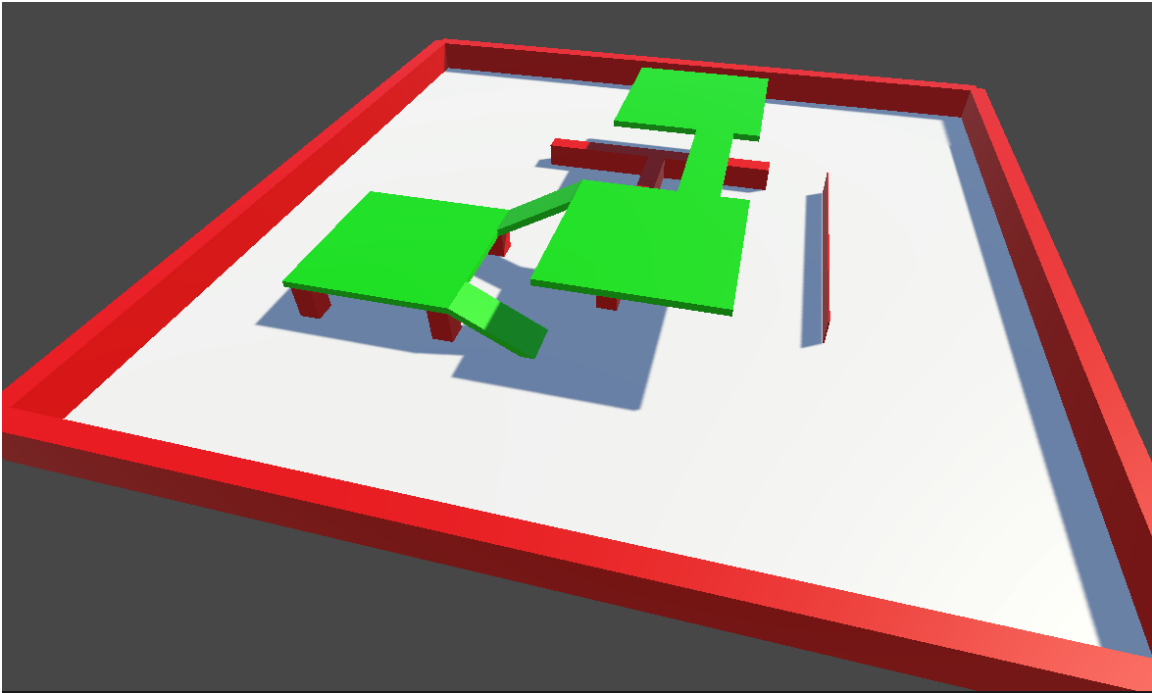


Figura 22 Creación y diseño del nivel base para desarrollar las actividades del juego RPG

En relación con el diseño y programación de los NPCs, se ha adoptado el enfoque delineado en el diagrama de flujo presentado en la Figura 17. Este diagrama conceptualiza los tres estados de movimiento fundamentales que definen el comportamiento del NPC:

- **Guardia:** En este estado el NPC permanece estático, dedicado a la vigilancia de una zona específica en una dirección predeterminada, esto permite mantener una postura defensiva, ideal para la protección de áreas críticas dentro del entorno del juego.
- **Patrulla:** El estado de patrulla implica que el NPC se desplace entre puntos de control o waypoints predeterminados dentro de la escena. Este movimiento estructurado proporciona la capacidad de explorar y vigilar diferentes áreas, añadiendo dinamismo a su comportamiento y enriqueciendo la interacción con el entorno.
- **Aleatorio:** Cuando el NPC entra en el estado aleatorio, experimenta un movimiento libre dentro de un perímetro específico, permitiendo una variación más orgánica en el comportamiento del NPC, simulando acciones menos predecibles y otorgándole una mayor adaptabilidad a escenarios cambiantes.

El diseño de estos estados de movimiento se ha concebido con el objetivo de proporcionar al NPC una gama diversa de comportamientos que enriquezcan su participación en el juego, donde cada estado ha sido cuidadosamente programado para contribuir a la inteligencia artificial del NPC, aportando complejidad y realismo a su actuación en el entorno virtual, lo que proporciona una experiencia de juego más inmersiva y desafiante para el jugador.

Además de los estados de movimiento previamente mencionados, el NPC cuenta con un algoritmo especializado que le permite seguir a otros personajes ya sean jugables o no, dentro de un área

permitida. Este algoritmo de seguimiento ha sido diseñado para ofrecer una experiencia dinámica y reactiva, brindando la capacidad de ajustar su posición con respecto a otros personajes en tiempo real. El algoritmo de seguimiento permite configurar la distancia a la que el NPC seguirá al personaje objetivo, lo cual es totalmente ajustable según las necesidades del diseño del juego, ofreciendo flexibilidad al momento de personalizar diversos tipos de personajes, adaptándola a la narrativa y mecánicas específicas del juego.

Es importante destacar que el algoritmo utiliza las capacidades de navegación integradas en Unity, aprovechando el sistema de Navigation<sup>25</sup> para calcular rutas óptimas y asegurar que el NPC siga a los personajes dentro de las áreas definidas. Esta integración con las herramientas de navegación de Unity garantiza un seguimiento preciso y fluido, evitando obstáculos y mejorando la coherencia del comportamiento del NPC en el entorno del juego.

Cuando el NPC se desplaza fuera de los límites predefinidos o se aleja del área designada, el script activa el mecanismo de reinicio, lo que permite o implica que el NPC regrese automáticamente a su punto de origen, restableciendo su posición inicial, esta característica no solo contribuye a mantener la coherencia espacial en el entorno del juego, sino que también evita posibles situaciones anómalas que podrían surgir debido a la exploración descontrolada del NPC.

La capacidad de reinicio se ha implementado cuidadosamente para asegurar que el mismo pueda adaptarse a distintos escenarios de juego y mantener una actuación predecible y alineada con las expectativas del diseño. En última instancia, esta funcionalidad refuerza la robustez y la consistencia del comportamiento del NPC, elevando la calidad general de la interacción con el jugador. Seguidamente se muestra en la Figura 23 la funcionalidad del movimiento del NPC.

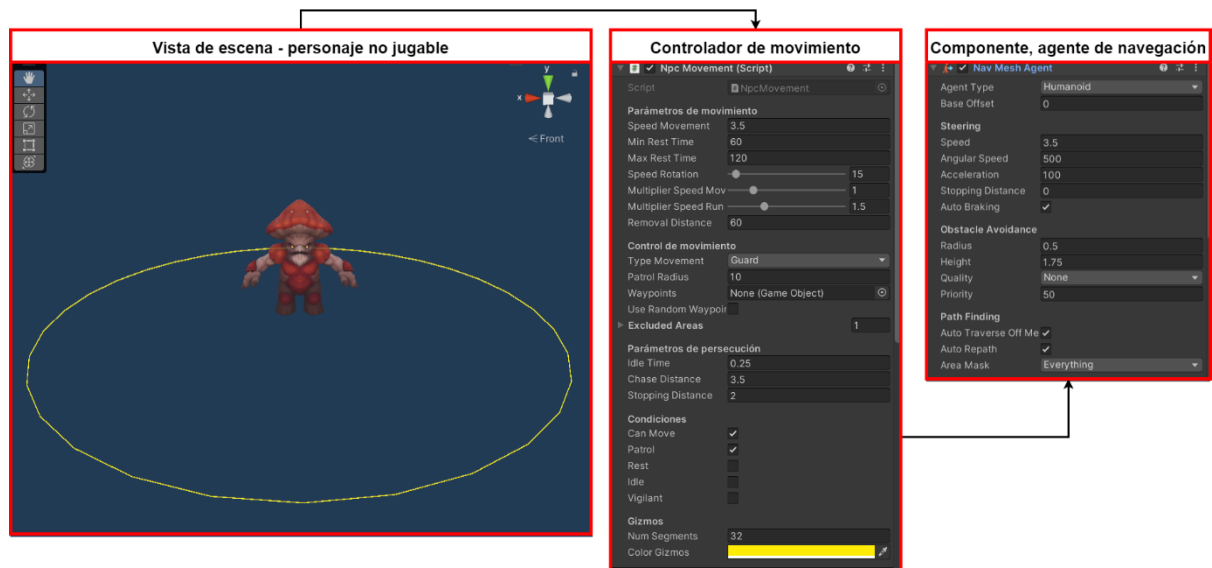


Figura 23 Funcionamiento para el movimiento del personaje no jugable

<sup>25</sup> <https://learn.unity.com/tutorial/navigation-basics>

Como se puede apreciar en la Figura 23, el NPC incluye un círculo de color amarillo ubicado en su punto central, generado mediante un script que utiliza el método `OnDrawGizmos`<sup>26</sup> de Unity. La razón detrás de este círculo es permitir una visualización clara del origen del NPC y facilitar la verificación de su área de patrulla aleatoria, este aspecto es particularmente relevante cuando se elige la opción "TypeMovement" con el movimiento de tipo "Random" y se ajusta el radio de patrulla a un valor mayor que cero. La incorporación de este círculo amarillo sirve como un recurso visual estratégico, brindando una representación gráfica precisa del área de patrulla del NPC en casos específicos, como el movimiento aleatorio. Esta implementación no solo mejora la configuración eficiente de parámetros, sino que también contribuye a una comprensión intuitiva y rápida del comportamiento del NPC durante el desarrollo del juego.

- **US-03: Animación de personajes jugables y no jugables**

La animación de personajes tanto jugables como no jugables, es un componente vital para infundir vida y realismo en el entorno del juego, es por ello que para realizar de manera efectiva este proceso se realiza mediante el uso de un controlador de animación que brinda Unity, el cual se combina con una amplia variedad de animaciones para lograr transiciones suaves y expresivas. A continuación, se detalla los componentes que se han integrado y combinado para crear un controlador de animaciones que defina con precisión cada una de las acciones de los personajes, mismos que son:

- **Animator Controller**<sup>27</sup>: Actúa como el director de la orquesta para las animaciones en el juego, lo que permite definir estados, transiciones y condiciones que determinan cómo y cuándo se reproducen las animaciones. Al integrar este controlador, se logra una gestión coherente y dinámica de las acciones de los personajes lo que contribuye significativamente a la inmersión del jugador.
- **Blend Tree**<sup>28</sup>: Este componente es especialmente útil cuando se tienen múltiples animaciones relacionadas con una acción específica y se necesita suavizar las transiciones entre ellas. Su uso es común en situaciones donde las acciones del personaje no son discretas, sino que existen variaciones continuas.
- **StateMachineBehaviour**<sup>29</sup>: Para un control más preciso y adaptativo de las animaciones, se incorpora código en el Animator Controller mediante el uso de StateMachineBehaviour. Este componente de Unity permite ejecutar scripts específicos en respuesta a eventos o condiciones dentro del controlador de animación, ayudando a ajustar el comportamiento de las animaciones de manera dinámica, respondiendo a cambios en el juego o en la interacción del jugador. Por ejemplo, dentro del juego cuando el personaje resulta aturdido por una habilidad, este no puede atacar ni caminar, esto se gestiona mediante el uso de este componente.

<sup>26</sup> <https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnDrawGizmos.html>

<sup>27</sup> <https://docs.unity3d.com/Manual/class-AnimatorController.html>

<sup>28</sup> <https://docs.unity3d.com/Manual/class-BlendTree.html>

<sup>29</sup> <https://docs.unity3d.com/ScriptReference/StateMachineBehaviour.html>



- **Avatar Mask**<sup>30</sup>: Es un componente que permite controlar qué partes específicas del avatar de un personaje afectarán o serán afectadas por las animaciones, lo que proporciona una forma precisa de definir áreas anatómicas específicas para la aplicación selectiva de movimientos o cambios de forma durante las animaciones.

Al utilizar el Avatar Mask, se pueden establecer restricciones o enmascarar ciertas partes del cuerpo del personaje, lo que resulta útil para optimizar la interacción de animaciones complejas, por ejemplo, dentro del juego el personaje tiene la capacidad de realizar ataques en movimiento. En esta dinámica, la animación de la parte inferior del cuerpo representa el caminar, mientras que la parte superior ejecuta la animación correspondiente al ataque, lo que permite que el personaje ataque de manera fluida mientras se desplaza, logrando una representación coherente y realista en la interacción del jugador con el personaje en el juego.

La combinación de estos componentes ofrece una flexibilidad excepcional en la implementación de animaciones lo que permite no solo gestionar las transiciones entre animaciones de manera fluida, sino también sincronizarlas con eventos del juego.

**Personajes jugables** La Figura 24 exhibe el controlador creado para las animaciones del personaje jugable, integrando estados como reposo, caminar, girar, combate, uso de habilidades, entre otros.

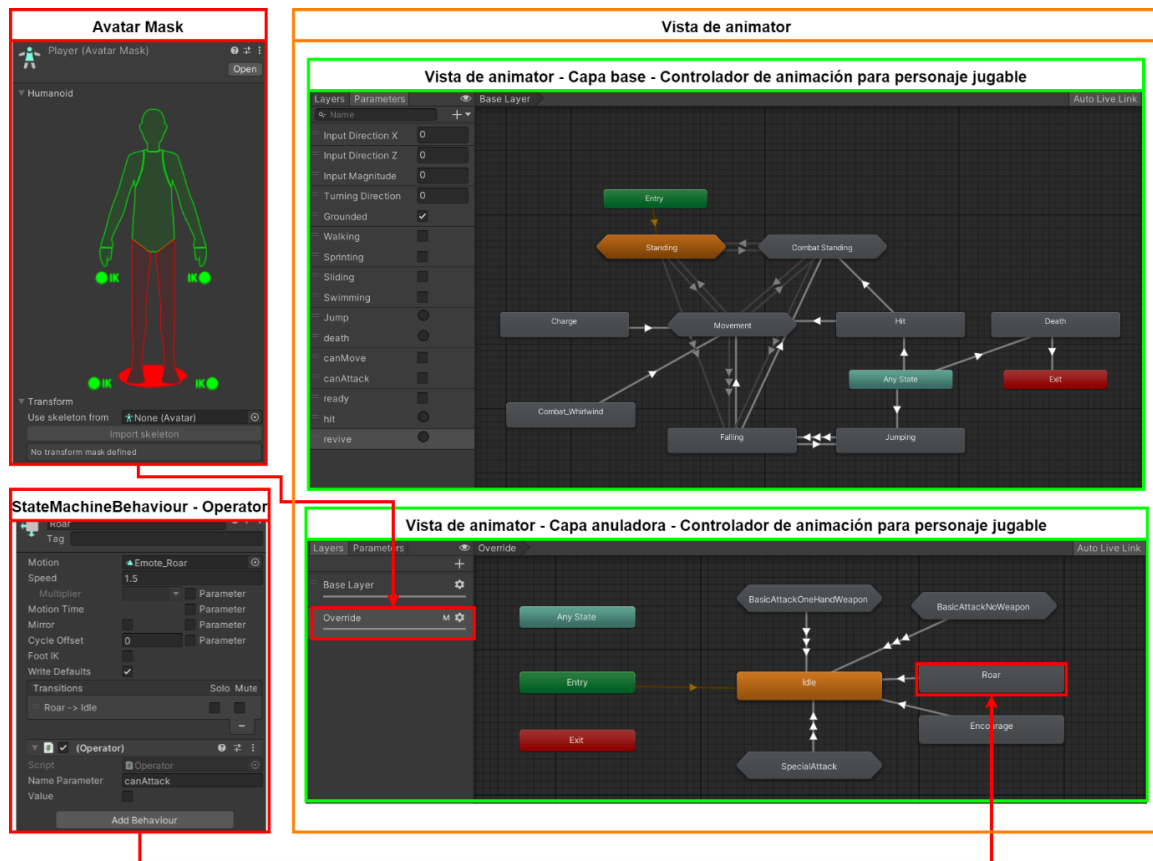


Figura 24 Controlador de animaciones para personajes jugables – clase guerrero

<sup>30</sup> <https://docs.unity3d.com/es/530/Manual/class-AvatarMask.html>

Como se evidencia en la Figura 24 el controlador de animaciones para el personaje de la clase Guerrero se compone de dos capas fundamentales, la primera es la capa base, que permite ejecutar una gran variedad de animaciones de movimiento, cubriendo acciones tanto terrestres, acuáticas y aéreas, como saltos y caídas, además incluye estados cruciales como muerte, recibir golpes y habilidades que requieren ejecutarse con el cuerpo completo del personaje.

Por otro lado, la capa anuladora se encarga de ejecutar animaciones específicas del torso hacia arriba, como ataques y el uso de habilidades como lo es la habilidad de rugir. Esta separación de capas facilita una gestión eficiente y modular de las animaciones, garantizando una representación coherente y detallada de las acciones del personaje.

Ciertas animaciones están vinculadas al script de Operator, el cual establece medidas clave para evitar que el personaje realice acciones de manera incoherente, dicho de otra manera, bloquea parámetros dentro del controlador cuando se ejecuta una animación prioritaria, esto con el fin de que las animaciones sean quienes decidas que debe reproducirse y más no lo script asociado al controlador de movimiento del personaje, Este script es crucial para mantener la integridad de las animaciones, asegurando su ejecución precisa y en consonancia con la lógica del juego.

**Personajes no jugables:** En la Figura 25, se muestra el controlador de animaciones del personaje no jugable, específicamente del Monarca Fungico. Este controlador capacita al NPC poder llevar a cabo ataques, lanzar habilidades y presentar la animación de muerte y recibir golpes.

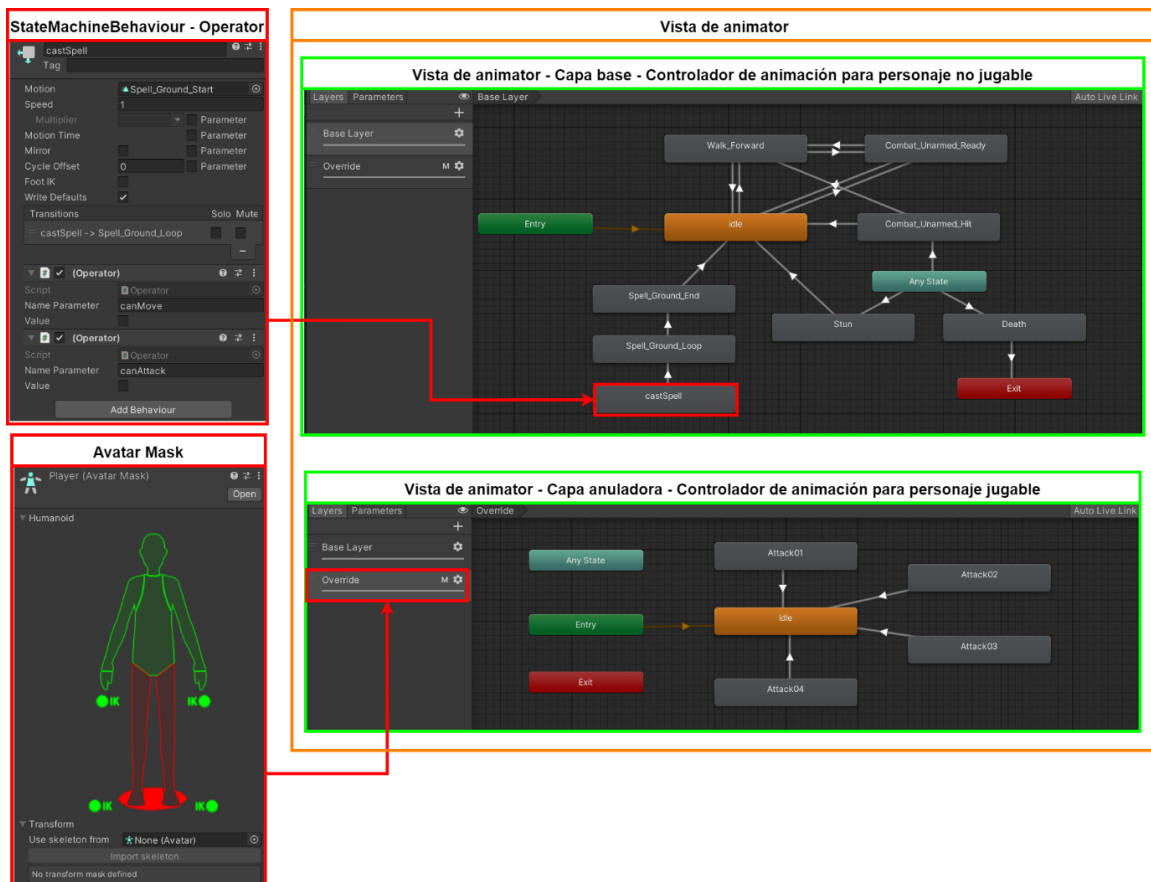


Figura 25 Controlador de animaciones para personajes no jugables – Monarca Fungico

Al igual que el controlador del personaje jugable, los personajes no jugables tienen la capacidad de llevar a cabo diversas animaciones, destacando especialmente la capacidad de lanzar habilidades. Esta habilidad de "casteo" permite al personaje no jugable presentar visualmente una animación antes de realizar el evento asociado. Es importante señalar que, durante la ejecución de esta animación, el personaje no jugable se encuentra temporalmente inhabilitado para realizar ataques o moverse, brindando la oportunidad de anticipar acciones de manera efectiva.

## 5.2. Sprint 2

La programación del segundo sprint está determinada por dos historias de usuario. A continuación, se detallan en la Tabla 14 las tareas correspondientes a cada historia.

<p><b>US-04: Programar sistema de habilidades</b></p>	<ul style="list-style-type: none"> <li>• Desarrollar un sistema de habilidades que permita la creación y configuración de una amplia variedad, que abarque desde habilidades de aumento (buffs) hasta habilidades de curación y daño, las cuales se dividen en categorías como habilidades directas, desplazamiento, proyectiles y de área (AoE) con patrón en forma de cono.</li> <li>• En el sistema de habilidades, incorporar la capacidad de aplicar efectos temporales, tanto de daño como de curación a lo largo del tiempo, junto con buffs. Estos efectos pueden ser configurados y creados directamente desde el editor de Unity, ofreciendo una flexibilidad adicional en el diseño y ajuste de las habilidades.</li> </ul>
<p><b>US-05: Programar IA para controlar el sistema de habilidades de los personajes no jugables</b></p>	<ul style="list-style-type: none"> <li>• Desarrollar un sistema que permita a los personajes no jugables utilizar habilidades, organizadas por fases específicas.</li> <li>• Configurar y gestionar la inteligencia artificial (IA) de los NPCs desde el editor de Unity, facilitando su ajuste y personalización.</li> </ul>

Tabla 14 Planificación de tareas del sprint 2

- **US-04: Programar sistema de habilidades**

El sistema de habilidades, se encuentra desarrollado mediante el uso de Scriptable Objects e implementación de herencia y POO, lo que permite una estructura modular y escalable para las habilidades. Notablemente, estas habilidades pueden ser generadas directamente desde el editor de Unity, proporcionando una interfaz amigable para su creación, y brindando la flexibilidad de configuración manual para adaptarse a las necesidades específicas del juego.

Las habilidades se encuentran clasificadas por tipo y categoría, de esta manera es mucho más dinámico poder crear nuevas habilidades. Tal como se aprecia en la figura 26.

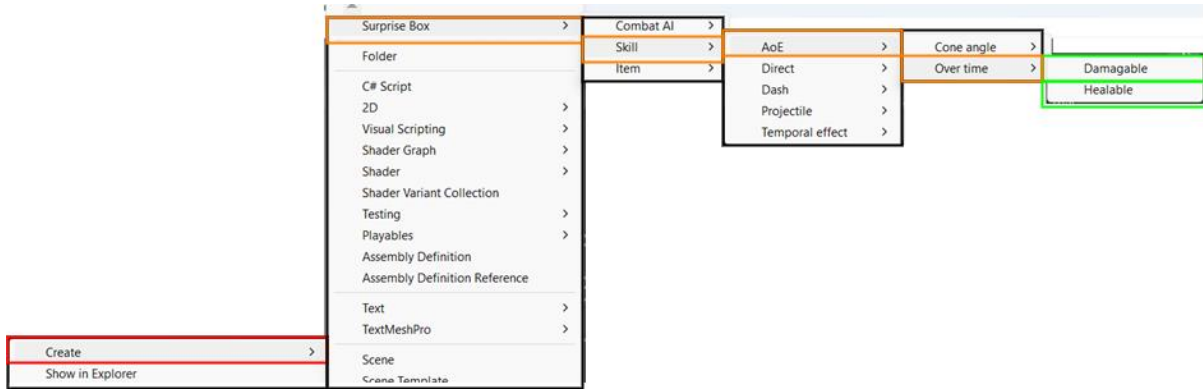


Figura 26 Ruta del sistema de habilidades para la creación de una nueva habilidad.

Como se evidencia en la Figura 26, en la ruta "Create/Surprise Box/Skill" se presenta una amplia variedad de habilidades, organizadas según el tipo de habilidad que se desea crear, permitiendo a través de este enfoque no solo ofrecer una excelente organización, sino que también proporciona una guía clara sobre el tipo de habilidad que se pretende desarrollar. Al seguir este proceso, se genera un nuevo Scriptable Object encargado de almacenar información detallada sobre la habilidad, dicho objeto incluye datos como el nombre, efecto de partículas, sonido, animación asociada a la habilidad, daño o curación según el tipo, y otras características relevantes. Este enfoque es de suma importancia, ya que separa la lógica de datos de la programación de uso de habilidades, facilitando un desarrollo más claro y mantenible.

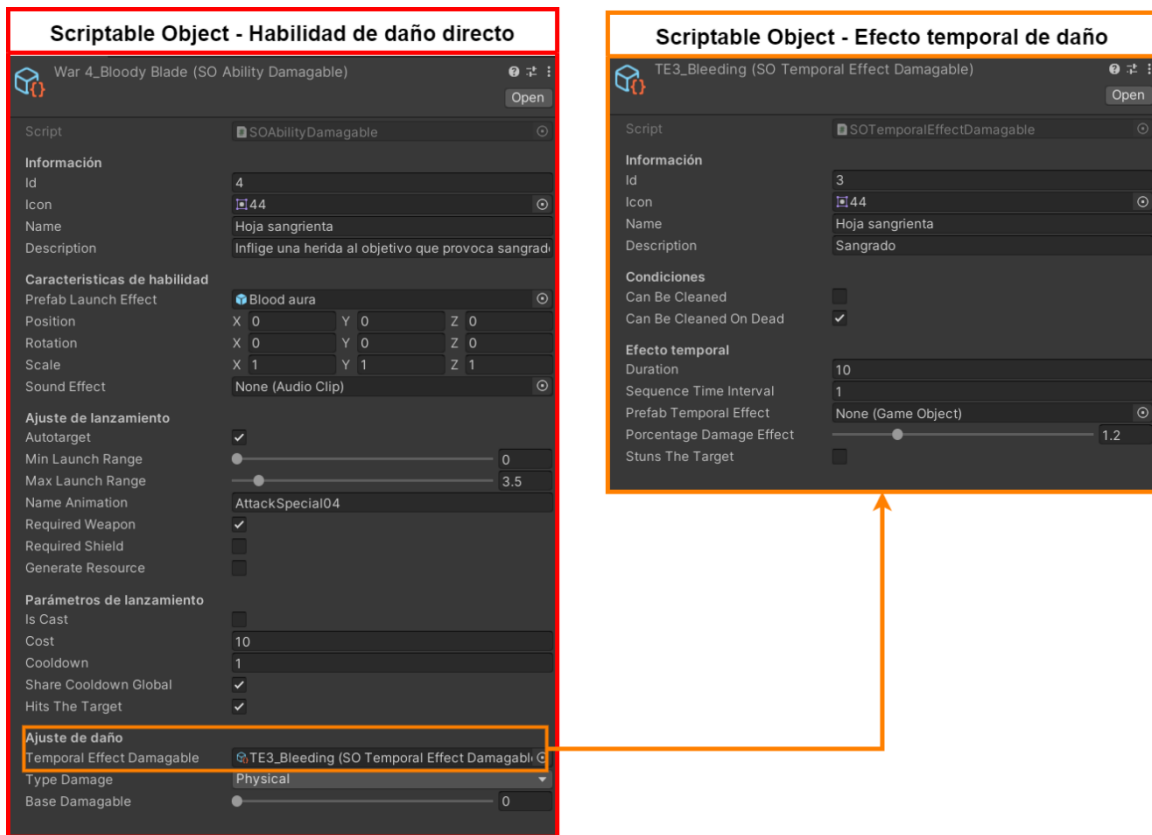


Figura 27 Scriptable Object de habilidad de daño directo que integra efecto temporal

Como se muestra en la Figura 27, el formato del Scriptable Object corresponde a una habilidad de daño directo. Por tanto, es crucial rellenar los campos adecuadamente para asegurar el correcto funcionamiento de la habilidad. En caso de no contar con referencias, la habilidad no podrá ejecutarse correctamente debido a la falta de información necesaria. Del mismo modo, la creación de efectos temporales puede realizarse utilizando la misma ruta mencionada al inicio de este sprint.

- US-05: Programar IA para controlar el sistema de habilidades de los personajes no jugables.**

Para gestionar eficientemente la IA de habilidades, mismas que pueden ser integradas o utilizadas por los NPCs, ya sea de manera obligatoria o no, se puede crear Scriptable Objects que almacena las habilidades de los NPC. Este objeto tiene la capacidad de almacenar tanto las fases, como las habilidades que puede emplear en cada una de ellas, es por ello que se toma en cuenta que las fases nos NPC deben gestionarse acorde al porcentaje de vida, asegurando así una adaptación dinámica de la IA en respuesta a la situación de combate. Previamente la Figura 28 muestra la ruta para poder crear la máquina de estados que almacena cada una de las fases del personaje no jugable con sus respectivas habilidades que puede usar.

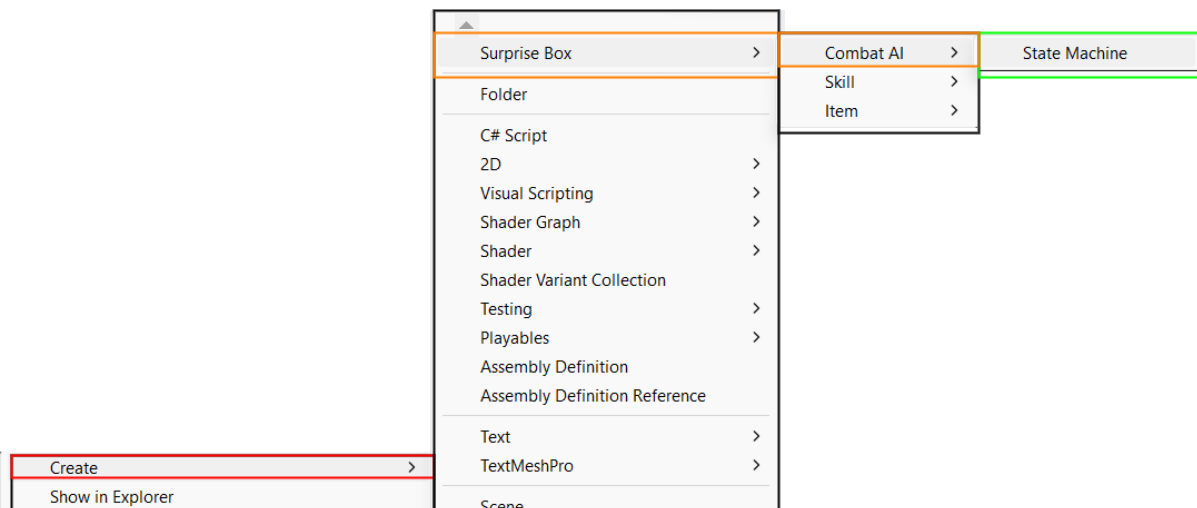


Figura 28 Ruta del sistema de máquina de estados que controla la IA de habilidades del personaje no jugable.

Como se observa en la Figura 28, la ruta para crear el controlador que gestiona la inteligencia artificial (IA) de los NPC presenta una única alternativa. De esta manera, se pueden crear diversas máquinas de estados para distintos tipos de personajes dentro del RPG. Es crucial tener en cuenta que los estados a desarrollar cuentan con un listado de fases, las cuales deben ajustarse según el porcentaje de vida deseado para concluir cada fase.

En la Figura 29, se presenta el Scriptable Object del controlador de habilidades del NPC, específicamente para el personaje NPC Monarca Fúngico, que posee dos fases. La primera fase, llamada "Llamado de la naturaleza", finaliza cuando el personaje pierde el 60% de su vida (ver variable

"MaxLifePorcentage (0.6)". Durante esta fase, el personaje puede lanzar la habilidad "Brotos de naturaleza". Una vez que la vida baja por debajo del 60%, se pasa a la segunda fase, "Llamado del caos", donde el personaje utiliza exclusivamente la habilidad "Terremoto". Es importante destacar que se pueden añadir tantas habilidades y fases como sea necesario en el gestor. Esto proporciona una amplia variedad de fases y opciones de juego en el RPG. Además, al separar los datos de la lógica, el programa se vuelve más escalable, permitiendo la integración de una diversidad de estados.

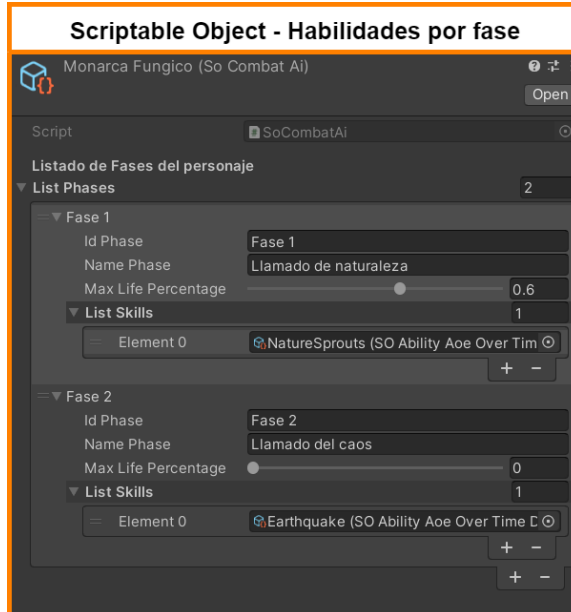


Figura 29 Scriptable Object de gestor de AI de habilidades para los personajes no jugables

### 5.3. Sprint 3

La planificación del tercer sprint está determinada por dos historias de usuario. A continuación, se detallan en la Tabla 15 las tareas correspondientes a cada historia.

<p><b>US-06: Programar sistema de estadísticas de los personajes</b></p>	<ul style="list-style-type: none"> <li>• Desarrollar un sistema de estadísticas para todos los tipos de personajes dentro del RPG, con el objetivo de asignar estadísticas principales tanto ofensivas como defensivas. Estas estadísticas incluyen información del personaje, nombre, vida, recurso (nada, mana, ira, energía), armadura, resistencia mágica, poder de ataque, poder de habilidad, velocidad de ataque, entre otros.</li> <li>• El sistema de estadísticas, al gestionar recursos tanto ofensivos como defensivos, debe ser capaz de manejar las acciones que le suceden al personaje, como perder o recuperar vida y recursos, entre otras. Además, es responsable de gestionar las habilidades que afectan al personaje, tales como efectos de daño, curación, efectos temporales y buffs.</li> </ul>
--	--

	<ul style="list-style-type: none"> <li>• El sistema de estadísticas debe contar con la capacidad de reiniciar, con el propósito de evitar que, en situaciones donde un NPC derrote al jugador, la vida del NPC quede a medio camino. De igual manera, al ocurrir la muerte del personaje jugador y resurgir, su vida deberá recuperarse por completo.</li> </ul>
<p><b>US-07: Programar sistema de combate de los personajes</b></p>	<ul style="list-style-type: none"> <li>• Programar el sistema de combate de los personajes para que puedan ejecutar acciones como ataques básicos y utilizar los Scriptable Objects de las habilidades descritas en el sprint 2.</li> <li>• Habilitar al sistema de combate para que los personajes puedan equiparse con una amplia variedad de armas y responder de manera efectiva a cada una de ellas.</li> <li>• Implementar un sistema de combate para los personajes jugables que les permita realizar ataques o habilidades siempre y cuando el objetivo se encuentre frente a ellos y a una cierta distancia preestablecida.</li> <li>• Para los personajes no jugables, asegurar que puedan usar habilidades o ataques siempre y cuando no exista un obstáculo entre el NPC y el objetivo, si esto sucede, proceder a que el NPC se acerque lo más posible al objetivo, para evitar obstáculos.</li> </ul>

Tabla 15 Planificación de tareas del sprint 3

- **US-06: Programar sistema de estadísticas de los personajes**

Es esencial priorizar la creación inicial del sistema de estadísticas de los personajes, ya que esta característica capacita al personaje el poder adaptarse a diversas situaciones, siendo el combate con otros personajes una de las más destacadas en un RPG. Para llevar a cabo enfrentamientos efectivos, es crucial conocer y gestionar los puntos de vida, así como comprender la cantidad de daño o resistencia que posee cada personaje.

En el diseño, se emplea la herencia y clases abstractas junto con el uso de interfaces para establecer una comunicación efectiva con el sistema de estadísticas, lo que permite que tanto personajes jugables y no jugables adopten esta funcionalidad. Al adoptar este enfoque, se consigue una organización eficiente y coherente, lo que posibilita que tanto los personajes jugables como los no jugables hereden y utilicen de manera efectiva las funcionalidades del sistema de estadísticas. Además, tienen la flexibilidad de integrar funcionalidades únicas si es necesario, evitando redundancias en el código base a largo plazo. Este enfoque no solo optimiza la funcionalidad del juego, sino también las llamadas y el rendimiento del sistema en general.

A continuación, en la Figura 30 se exhibe el Diagrama Entidad-Relación del sistema de estadísticas. Este diagrama utiliza interfaces, siendo crucial para la comunicación de otros subsistemas que se incorporan a medida que se desarrolla el RPG.

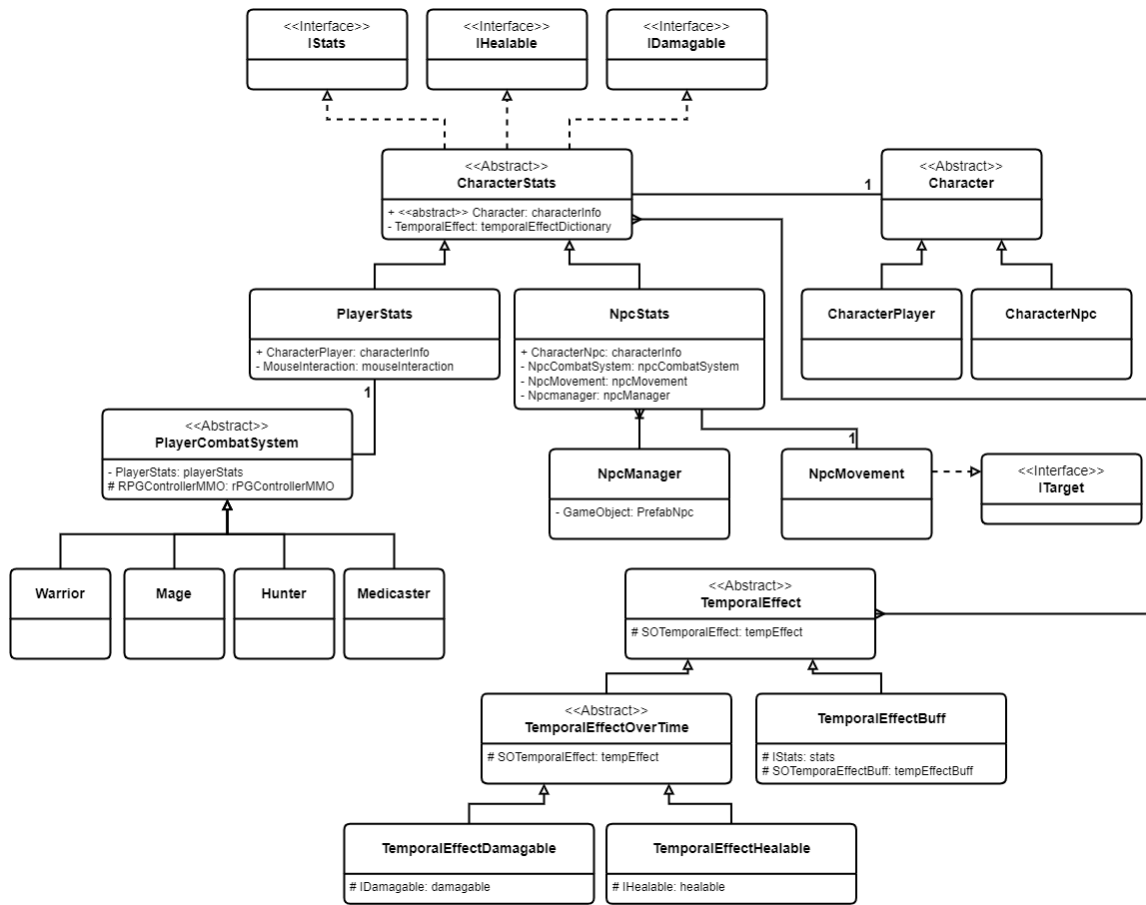


Figura 30 Diagrama Entidad-Relación del sistema de estadísticas de los personajes

El sistema de estadísticas está equipado con un diccionario que tiene la capacidad de almacenar todos los efectos que un personaje recibe durante el juego. Este diccionario actúa como un registro dinámico que registra y gestiona todos los cambios en las estadísticas del personaje causados por efectos temporales, buffs o debuffs.

El sistema de estadísticas incorpora funcionalidades clave de reinicio en diversas circunstancias. En primer lugar, si un NPC logra eliminar a su objetivo, el sistema se reinicia para prepararse para posibles interacciones futuras. Además, si el gestor de movimiento detecta que un NPC ha salido de su perímetro de patrulla designado, el sistema de estadísticas se reinicia para restablecer adecuadamente el estado del personaje no jugable. En el caso del personaje jugable, si este experimenta la derrota y muere durante el juego, el sistema de estadísticas se reinicia para permitir que el jugador vuelva a participar en la acción. Este enfoque de reinicio en situaciones específicas asegura una dinámica de juego continua y coherente, manteniendo la integridad de las estadísticas y facilitando una experiencia de juego fluida y equilibrada.

El script, visualizado desde el editor de Unity, sigue el formato que se muestra en la Figura 31. Este script debe ser adjuntado como un componente dentro del personaje correspondiente, siendo PlayerStats para el personaje jugable y NpcStats para el personaje no jugable en este caso.



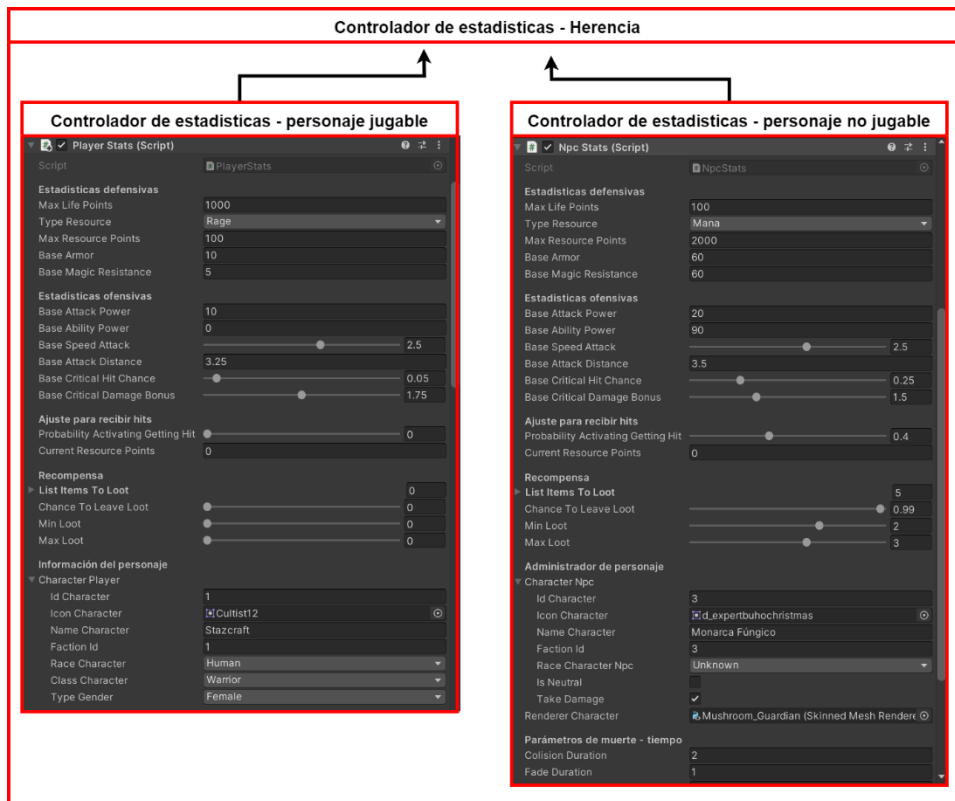


Figura 31 Script del controlador de estadísticas para cada tipo de personaje

Como se evidencia en la Figura 31, las estadísticas pueden ser modificadas desde el editor de Unity, lo que garantiza que se puede crear una gran variedad de personajes. Este nivel de flexibilidad no solo permite ajustar las estadísticas de los personajes no jugables, como se ilustra en el caso del NPC Monarca Fúngico, sino que también facilita la personalización directa del personaje jugable. De esta manera, los desarrolladores tienen la capacidad de adaptar dinámicamente las características y habilidades del personaje principal en función de las necesidades específicas del juego.

El hecho de poder modificar directamente las estadísticas del personaje jugable desde el editor facilita la iteración y la toma de decisiones rápidas en el diseño del juego. Esta versatilidad contribuye a la eficiencia del proceso de desarrollo, permitiendo a los creadores ajustar y afinar las características del personaje de manera ágil y efectiva para lograr una experiencia de juego envolvente y bien equilibrada.

- **US-07: Programar sistema de combate de los personajes.**

El sistema de combate constituye un componente de gran complejidad en el contexto de este trabajo, ya que su función principal abarca tanto el manejo de habilidades como la utilización de armas. Su objetivo fundamental es proporcionar una experiencia de combate extremadamente dinámica, que es la esencia y propósito central de este Trabajo de Fin de Máster (TFM).

Este sistema de combate actúa como el núcleo que conecta con otro sistema esencial, encargado de implementar el patrón de diseño fábrica y pool Object Véase Figura 32. Esta conexión estratégica tiene como finalidad gestionar de manera efectiva tanto las armas como las habilidades. La implementación de estos patrones específicos permite evitar la creación y destrucción constante de instancias de

habilidades, favoreciendo en cambio la reutilización eficiente de estas instancias. Esto no solo mejora el rendimiento del juego, sino que también contribuye significativamente a la optimización de recursos y a una experiencia de juego más fluida y eficaz.

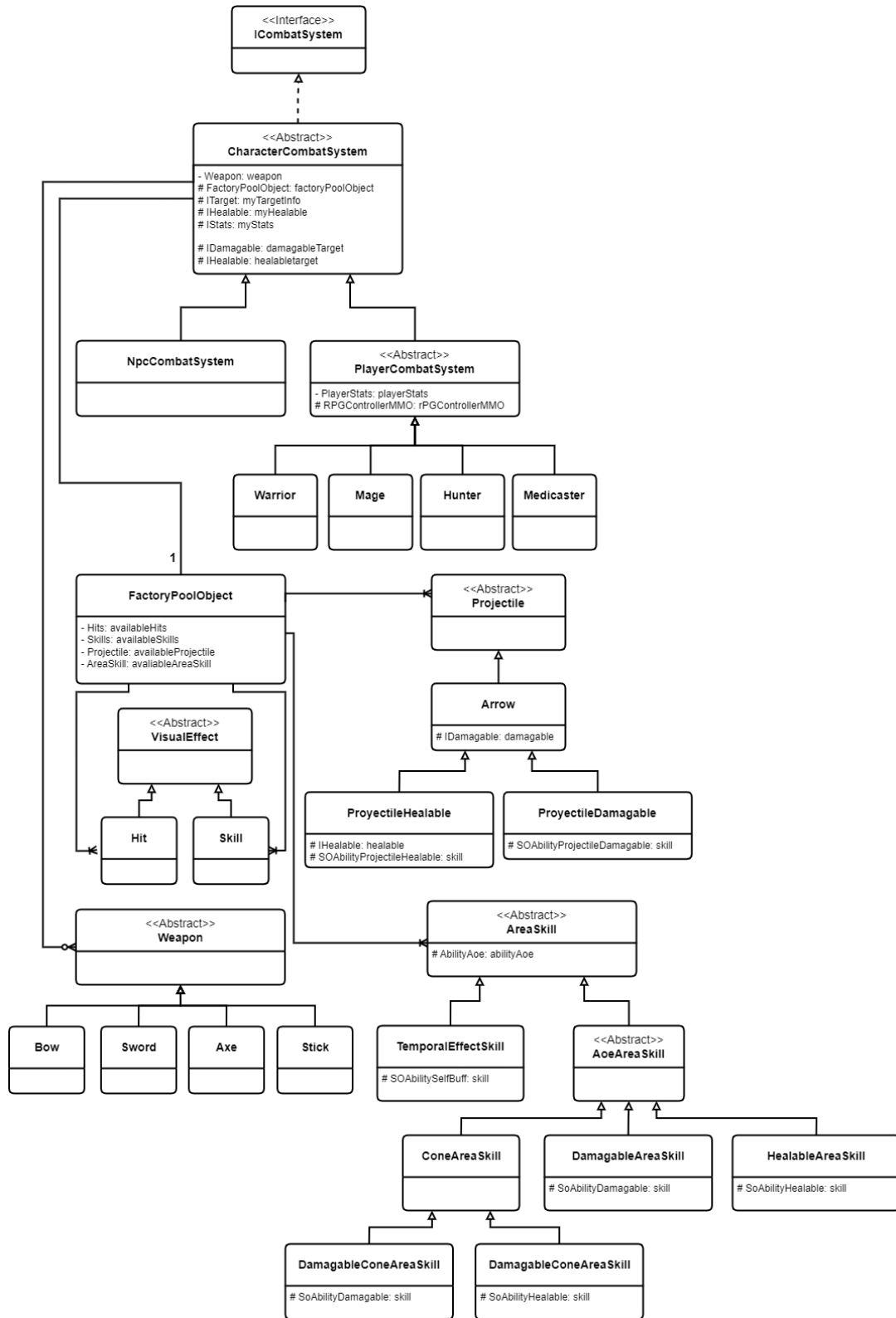


Figura 32 Diagrama Entidad-Relación del sistema de combate de los personajes

Desde una perspectiva más dinámica, la Figura 33 ilustra la interacción del diagrama previamente visto. Esto se realiza con el objetivo de proporcionar un mejor entendimiento del sistema de combate.

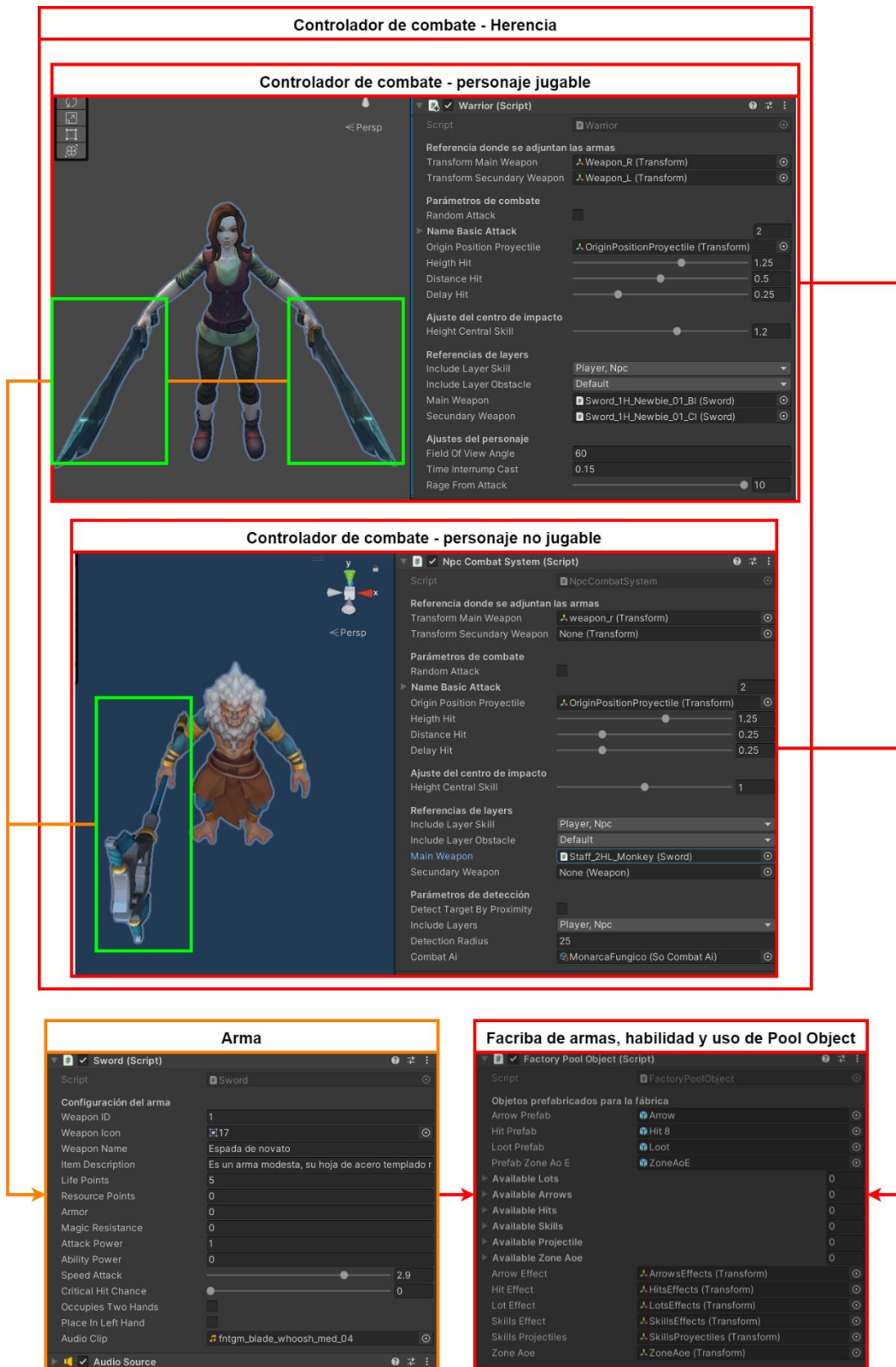


Figura 33 Sistema de control de combate de los personajes desde la perspectiva del editor de Unity

En esta sección del sprint, se detallará el sistema de combate diseñado para cada tipo de personaje. Ambos, al utilizar la herencia, comparten las mismas variables, métodos y otros componentes heredados de una clase superior. No obstante, cada clase, ya sea NpcCombatSystem o PlayerCombatSystem (Warrior), incorpora lógicas únicas propias de su categoría. Este enfoque garantiza que el personaje jugable pueda utilizar habilidades de manera oportuna según las decisiones del jugador, mientras que el personaje no jugable emplea habilidades de acuerdo con el estado programado, como se especifica en la Historia de Usuario 5 del sprint 2: Programar IA para controlar el sistema de habilidades de los personajes no jugables.

**Sistema de combate para personajes jugables:** El sistema de combate diseñado para este tipo de personaje se caracteriza por requerir un componente esencial para lograr un uso eficiente de cada habilidad perteneciente a la clase. Tal como se aprecia en la Figura 34.

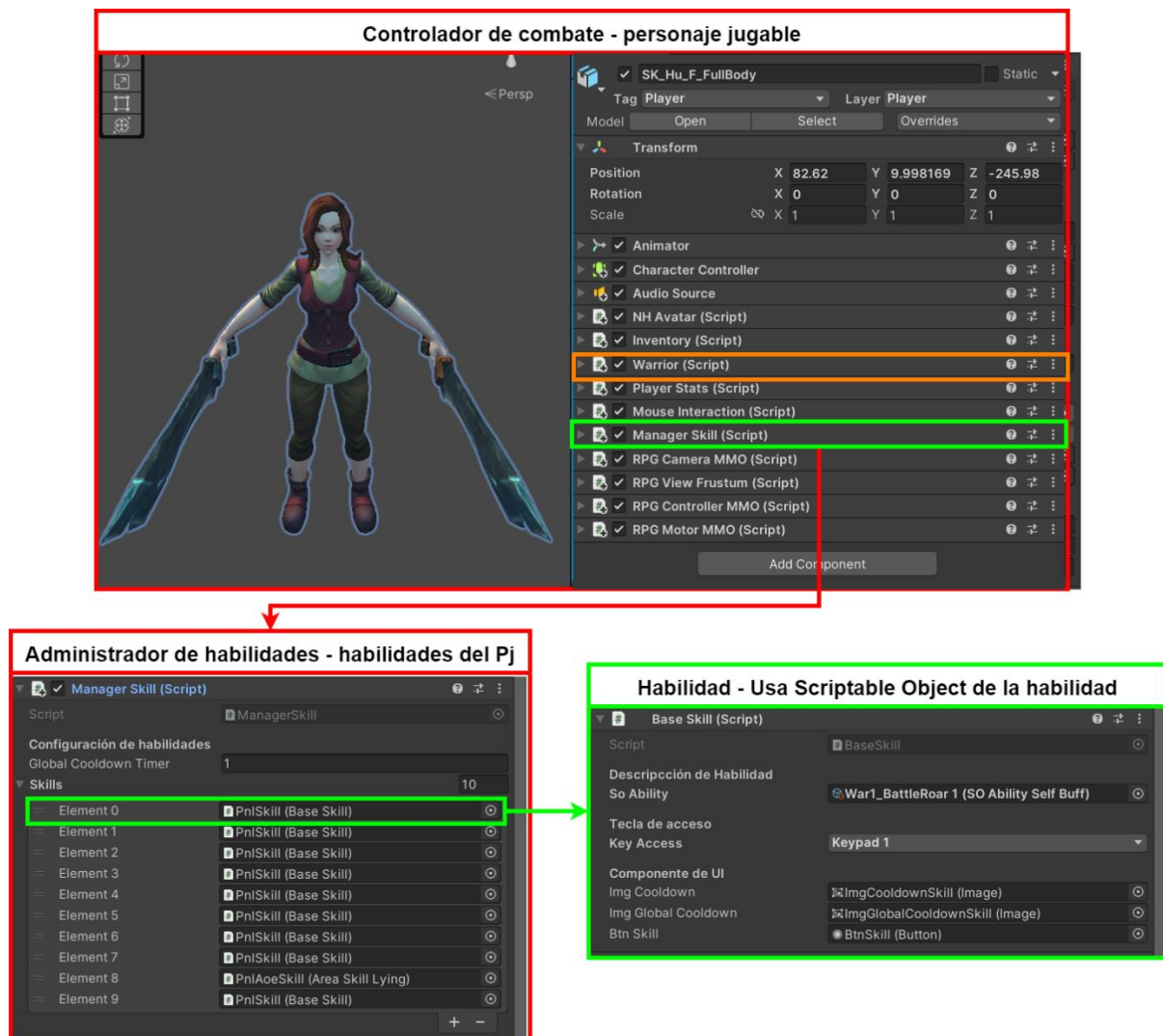


Figura 34 Sistema de combate para el personaje jugable

Este componente es el Administrador de Habilidades (ManagerSkill), el cual almacena todas las habilidades disponibles, donde cada habilidad está asociada a un Scriptable Object que contiene toda la información relevante. Cuando el jugador presiona la tecla asignada a una habilidad, esta se activa de inmediato, desencadenando una serie de eventos en segundo plano, este proceso incluye, en primer lugar, la validación de si el personaje puede usar la habilidad, seguida de la verificación de si dispone de suficientes recursos, si el objetivo está vivo y si se encuentra dentro del alcance, entre otras condiciones. Estas validaciones aseguran un lanzamiento exitoso de la habilidad, véase Figura 35.

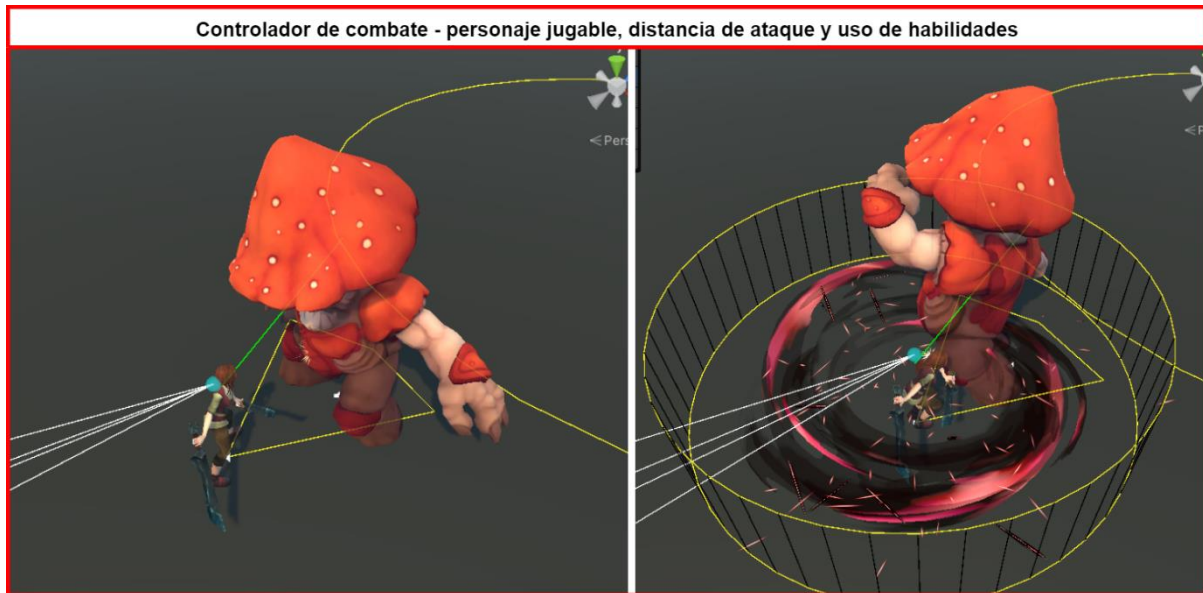


Figura 35 Sistema de combate mediante ataques y uso de habilidades del personaje jugable

**Sistema de combate para personajes no jugables:** Con respecto al personaje no jugable, este dispone dentro su script una variable que permite integrar el Scriptable Object con la información de las habilidades que puede llegar a implementar durante un encuentro, Véase la Figura 36.

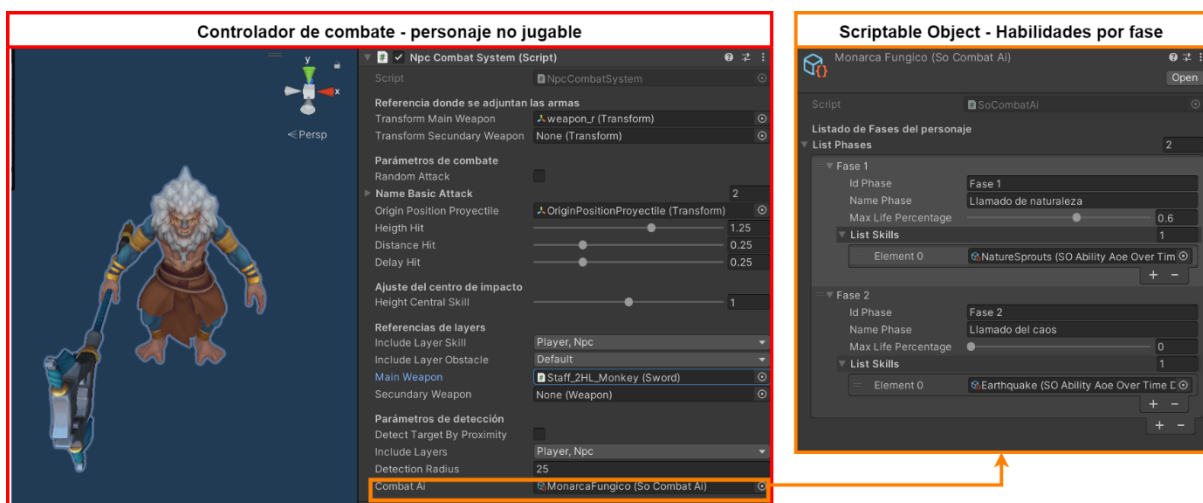


Figura 36 Sistema de combate para el personaje no jugable

Como se mencionó anteriormente, las habilidades utilizadas por el NPC deben ser dinámicas y estar sincronizadas con las animaciones además el mismo no está en la obligación de integrar este componente de habilidades, debido a que si no lo integra solo tendrá la opción de realizar ataques básicos.

Una necesidad para este tipo de circunstancias es de que se requiere un componente clave para que el jugador pueda anticipar la ubicación de dichas habilidades. Es por ello que la fábrica encargada de gestionar la creación de instancias y otros recursos también posibilita la creación de una zona que informa al jugador sobre el área donde caerá la habilidad. El objetivo es permitir al jugador anticiparse a estas situaciones, brindándole retroalimentación visual. De esta manera, se evita la incertidumbre acerca de la ubicación o el momento en que se lanzará la habilidad. Como se puede observar en la Figura 37, se presenta la zona donde se aplicará la habilidad del NPC, después de un cierto tiempo, la habilidad se activa y se muestra, es crucial que la configuración del área de impacto en la zona coincida con la configuración establecida en el Scriptable Object correspondiente a esa habilidad.



Figura 37 Sistema de combate y habilidades del personaje no jugable

La integración de la gestión de habilidades mediante la creación de zonas de impacto brinda al jugador una experiencia más inmersiva y estratégica. Con estos avances, se sientan las bases para futuros desarrollos que enriquecerán aún más la jugabilidad del RPG.

#### 5.4. Sprint 4

Para la ejecución de cuarto sprint, es imperativo realizar una revisión exhaustiva de las tareas asociadas a cada historia de usuario que compone dicho sprint. A continuación, la Tabla 16 exhibe la planificación detallada correspondiente a esta fase inicial del desarrollo del proyecto.



<p><b>US-08: Diseñar y programar una UI para visualizar las barras de salud de los personajes</b></p>	<ul style="list-style-type: none"> <li>• Diseñar, desarrollar y programar la lógica que permita al usuario visualizar el estado de combate y otros componentes esenciales de la interfaz de usuario (UI), como la barra de acciones, las barras de estado, el inventario y el gestor de equipamiento.</li> <li>• Es crucial representar aliados, enemigos y criaturas neutrales dentro de la UI para que el jugador pueda identificar el tipo de enemigo al que se enfrenta y visualizar la vida y los recursos del objetivo.</li> <li>• La UI también debe ser un medio para que el jugador pueda ver las recompensas obtenidas al derrotar monstruos.</li> <li>• Otro aspecto importante es mostrar a través de la UI características clave de todo RPG, como las habilidades con su respectiva información y las armas con sus detalles correspondientes. Esto tiene como objetivo familiarizar al jugador con los conceptos del juego.</li> </ul>
<p><b>US-09: Programar sistema de Respawn de NPC</b></p>	<ul style="list-style-type: none"> <li>• Programar un sistema que permita a los NPC reaparecer después de ser eliminados, con un tiempo de reparación predefinido.</li> <li>• Los NPC deben generarse de manera aleatoria dentro de un perímetro determinado.</li> </ul>

Tabla 16 Planificación de tareas del sprint 4

- **US-08: Diseñar y programar una UI para visualizar las barras de salud de los personajes**

En esta sección, se aborda sobre el diseño y la programación de una interfaz de usuario (UI) destinada a visualizar las barras de salud de los personajes dentro del proyecto. Para esta tarea, se aprovecha el sistema de Canvas de Unity, permitiendo la creación y disposición eficiente de elementos gráficos en la pantalla, donde cada componente de la UI, como: la barra de acción, las barras de estado, el inventario y el gestor de equipamiento, es creado y organizado como contenedores separados. Es importante destacar que, en este contexto, la UI se mantiene sencilla en cuanto a diseño, con una paleta de colores básica y el uso de sprites para elementos gráficos, buscando una apariencia limpia y funcional sin excesos visuales.

Seguidamente, se presenta en la Figura 38 el diseño de la interfaz de usuario (UI) que incorpora todas las necesidades e información requerida para brindar una experiencia de juego agradable. Es importante destacar que el diseño se ha elaborado con la finalidad de que sea intuitivo y evite generar confusiones. Cada ventana de la UI cuenta con un texto de encabezado que permite al usuario identificar rápidamente el tipo de información que presenta.

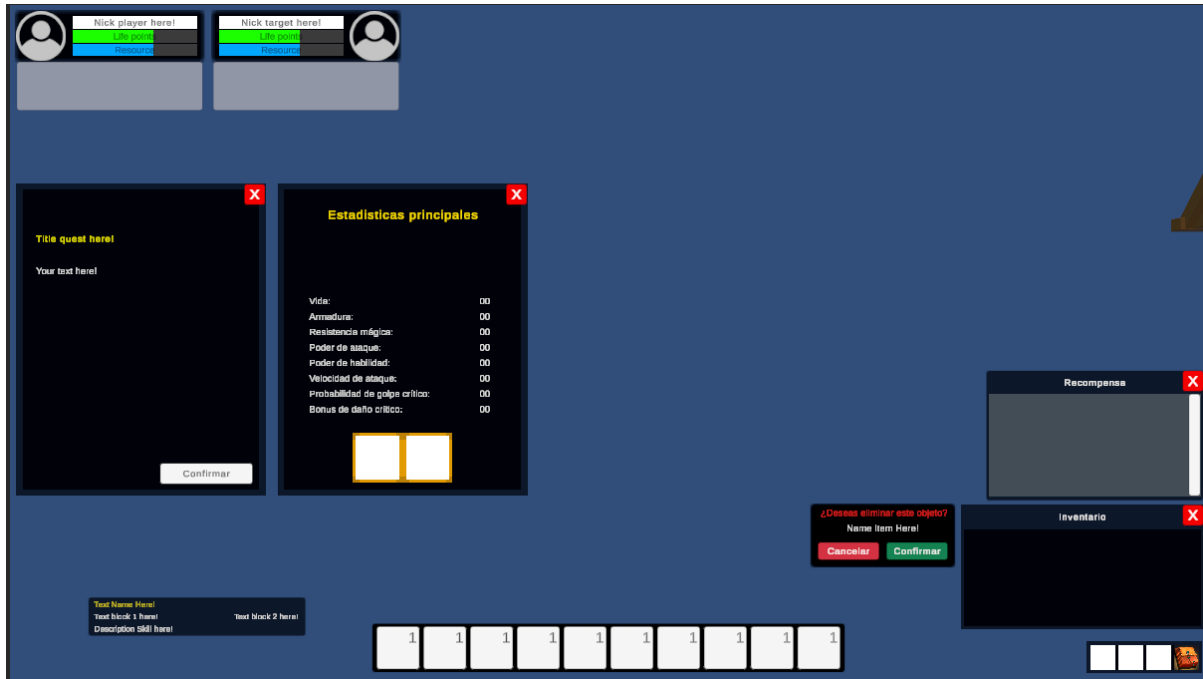


Figura 38 Diseño final de UI que integra el juego RPG, basada en el diseño propuesto

Como se aprecia en la Figura 38, la interfaz de usuario (UI) presenta secciones o ventanas que no ocupan el espacio central de la pantalla, con el objetivo de proporcionar una experiencia de juego sin obstruir el punto central de los acontecimientos en el videojuego. Es importante destacar que esta interfaz, al ser sencilla, no genera inconvenientes para que el usuario comprenda la función de cada sección, lo cual facilita y brinda intuitividad al jugar por primera vez.

- **US-09: Programar sistema de Respawn de NPC**

En la implementación del sistema de Respawn de NPC, se ha desarrollado un script que se encarga de gestionar la generación de NPC de acuerdo con un prefab establecido. Este script establece límites para la cantidad de NPC que pueden generarse y especifica dentro de qué perímetro deben aparecer. Es importante destacar que el gestor de Respawn de NPC también se encarga de generar un nuevo NPC cada vez que uno de ellos muere, lo que permite una experiencia de juego RPG en la que no hay límites para el reaparecimiento de NPCs, permitiendo a los jugadores explorar y enfrentarse a enemigos de manera continua.

Para asegurar la efectividad del script encargado de gestionar la aparición de los NPC, se ha implementado el uso de OnDrawGizmos, una herramienta esencial en el desarrollo de este videojuego. Además, se han utilizado disparadores del NavMesh mediante código para verificar un área específica. Este enfoque garantiza que los NPC se generen a la altura del terreno, evitando que aparezcan sobre superficies inexistentes o queden atrapados en áreas inaccesibles. Como se ilustra en la Figura 39, el gestor de spawn de NPCs cumple con la función esencial de generar todos los personajes no jugables en el juego, este enfoque ha sido diseñado para optimizar el rendimiento del juego durante la carga, evitando así una sobrecarga de modelos 3D en el entorno.



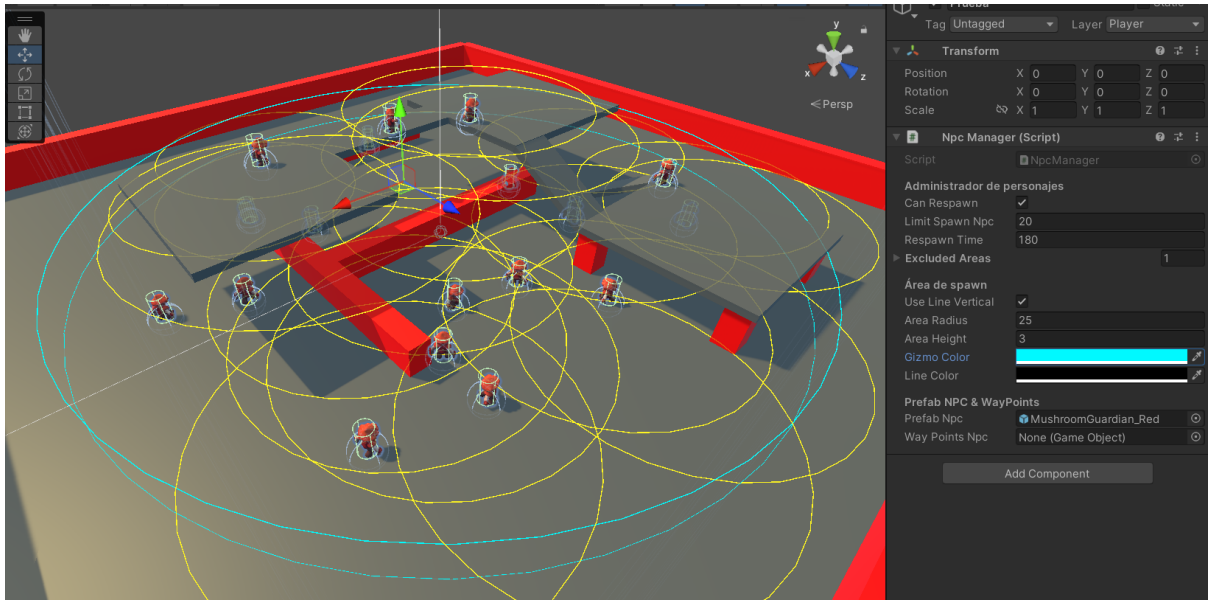


Figura 39 Funcionalidad del gestor de spawn de NPCs

Como se aprecia en la Figura 39, el círculo de color celeste representa el perímetro dentro del cual los NPC pueden generarse, estos personajes tienen la capacidad de generarse en todas las superficies previamente reconocidas por el sistema de navegación.

## 5.5. Sprint 5

La planificación del sprint cinco se enfoca en tres historias de usuario cruciales. La primera historia se centra en implementar un sistema de recompensas en el videojuego RPG permitiendo que el jugador obtenga recompensas despojables al eliminar NPCs. La segunda historia aborda la creación de un escenario final donde se produce toda la trama y la tercera historia y final es poder integrar sonidos que vayan en armonía con el videojuego. La tabla 17 detalla las tareas específicas asignadas a cada una de estas historias de usuario.

<p><b>US-10: Programar sistema de recompensas</b></p>	<ul style="list-style-type: none"> <li>• Programar un sistema de recompensas que brinde al jugador la oportunidad de obtener botín al derrotar criaturas.</li> <li>• El sistema de despojo o botín se generará en función de la probabilidad de despojo del ítem. Así, los ítems más valiosos tendrán una frecuencia de aparición menor, mientras que los ítems comunes serán más recurrentes.</li> <li>• Las recompensas se presentarán de inmediato al derrotar al enemigo, mostrando de manera visual una sección de botín que caerá junto al personaje vencido. Este enfoque proporciona una experiencia visual gratificante justo después de la victoria sobre el enemigo.</li> </ul>
---	--

<p><b>US-11: Creación del escenario</b></p>	<ul style="list-style-type: none"> <li>• Generar un entorno de fantasía con un estilo artístico Stylized, que combine una ambientación clara y detallada.</li> <li>• Integrar una pequeña villa como la principal zona donde se desarrolla la historia</li> <li>• Incorporar paisajes extensos y decoraciones que proporcionen un ambiente característico de un juego de rol.</li> </ul>
<p><b>US-12: Integración de sonidos</b></p>	<ul style="list-style-type: none"> <li>• Integrar efectos de sonido para cada una de las acciones dentro del juego, que incluyan sonidos ambientales, de armas, habilidades y enemigos.</li> </ul>

Tabla 17 Planificación de tareas del sprint 5

• **US-10: Programar sistema de recompensas**

El sistema de botín se integra al script encargado de gestionar las estadísticas de los personajes, es debido a que este sistema controla la vida y detecta cuándo un personaje muere, haciendo que se encargue de generar el botín tras este evento.

Generalmente los NPC tienen una variada lista de ítems que pueden soltar y es por ello que se debe configurar la cantidad mínima y máxima de ítems que pueden caer como parte del botín, es sumamente importante mencionar que los valores mínimo y máximo son solo referencias, ya que el código genera un número aleatorio entre estos límites para determinar la cantidad exacta de botín. Los ítems o tesoros son objetos de Scriptable Objects, es esencial utilizar este marco de trabajo con dicha estructura de datos, debido a que se asegura que los ítems (como armas) que el personaje jugable se puede equipar pasen por la fábrica y el pool Object, evitando así crear instancias innecesarias y garantizando una gestión eficiente de memoria y recursos, lo que contribuye a un aumento en los FPS del juego.

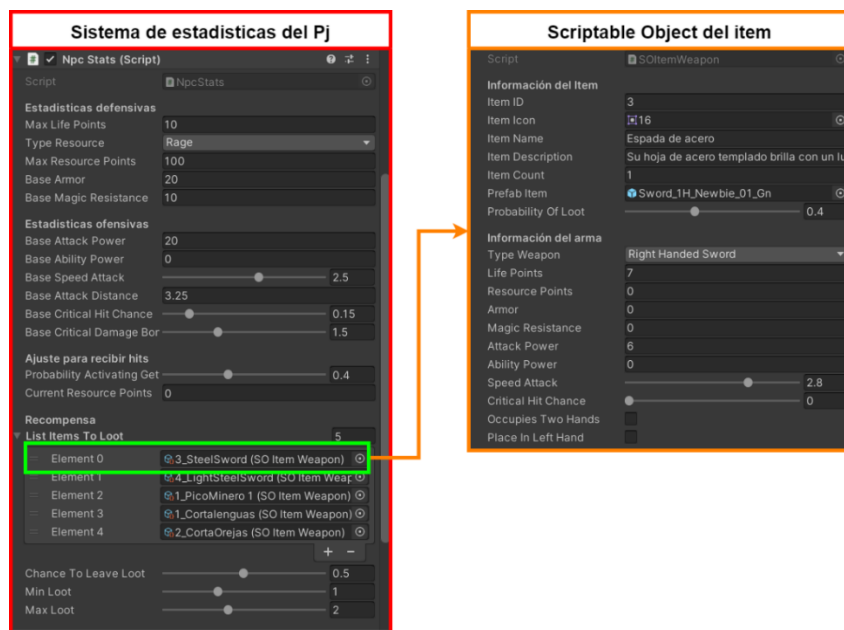


Figura 40 Sistema de gestión de recompensas por parte de los personajes

Una vez que el NPC es derrotado, deja caer un tesoro que resplandece cerca de él, proporcionando una perspectiva visual llamativa para indicar que algo importante ha ocurrido con ese personaje. Para explorar el contenido del botín, el jugador puede hacer clic derecho sobre el NPC caído. Una vez que se ha recogido todo el botín, el efecto visual de la recompensa desaparece, dando la sensación de que el personaje ya no tiene nada relevante que ofrecer. La Figura 41 presenta de manera detallada el proceso descrito anteriormente.



Figura 41 Sistema de despojo de recompensa de los NPC

Cabe mencionar que las recompensas, al ser recogidas, se integran automáticamente en el inventario del jugador, permitiéndole acumular una amplia variedad de equipamiento para diversas situaciones. Este sistema de combate basado en el botín enriquece la experiencia del jugador, ya que le brinda la posibilidad de mejorar y personalizar su personaje con armas, objetos y otros elementos adquiridos durante sus enfrentamientos con los NPC. La gestión eficiente del inventario se vuelve esencial para adaptarse estratégicamente a diferentes desafíos y maximizar el rendimiento en el juego.

- **US-12: Creación del escenario**

En la creación del escenario, se han empleado recursos previamente mencionados en la sección 4.2, utilizando assets del inventario de recurso, donde el terreno se ha configurado con dimensiones de 600x600 unidades, aprovechando las funcionalidades del sistema de relieves para generar paisajes detallados con variaciones en la topografía, asimismo se utiliza el sistema de generación de pastizales y árboles para dotar al entorno de elementos naturales.

Este escenario busca fusionar elementos de fantasía, siguiendo un estilo artístico Stylized para lograr una ambientación clara y detallista, es por ello que el escenario dispone de cuatro áreas principales:

(A) La villa, (B) el bosque encantado, (C) las ruinas (D) la zona bajo la montaña. Cada una de estas áreas contribuye a la diversidad del entorno, proporcionando distintos contextos y desafíos para la experiencia de juego. Véase la Figura 42 para una mejor representación de cada zona.

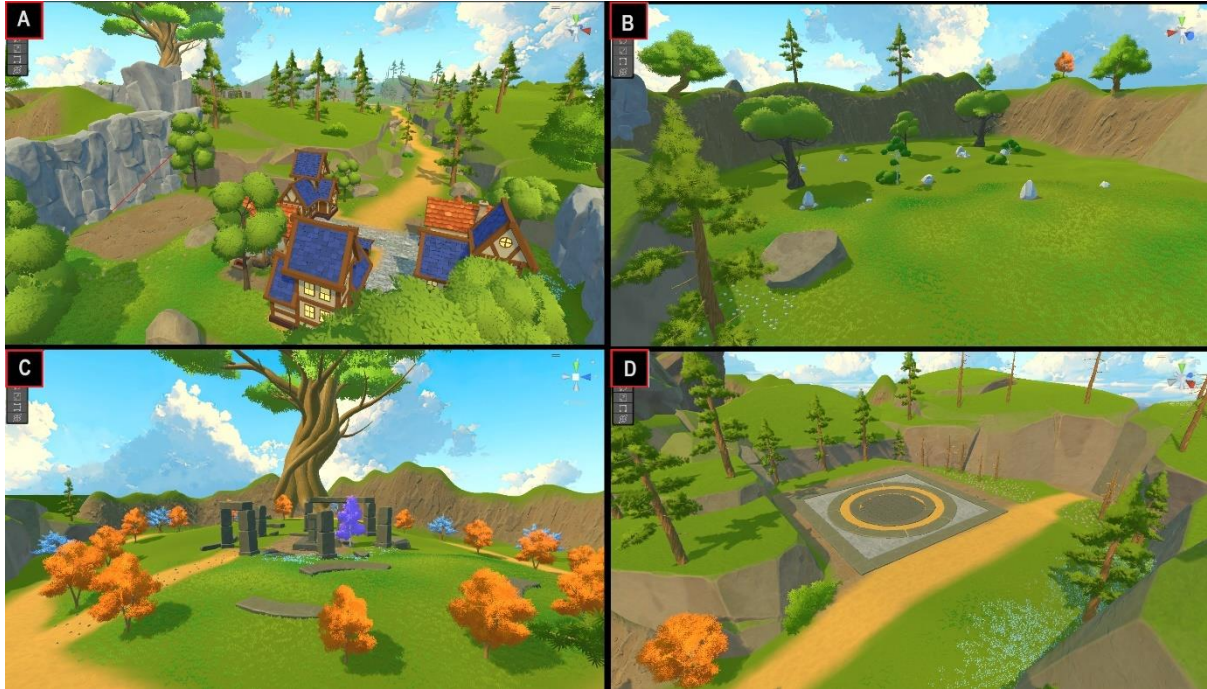


Figura 42 Ambiente creador para el videojuego RPG estilo Stylized

El diseño del escenario ha sido cuidadosamente planificado para proporcionar al jugador una experiencia dinámica y no lineal, donde cada zona del escenario presenta niveles de dificultad variados, permitiendo que el jugador se aventure y explore de manera más libre y flexible, en contraste con la estructura lineal comúnmente encontrada en otros juegos.

Esta decisión busca fomentar la exploración y adaptar el desafío a la progresión del jugador, ofreciendo una experiencia más inmersiva y emocionante.

- **US-13: Integración de sonidos**

El último paso es integrar sonidos al juego RPG, con el objetivo de proporcionar una experiencia auditiva inmersiva y enriquecedora, es por ello que se incorporan distintos tipos de sonidos para diversas acciones y eventos dentro del juego, como efectos ambientales que refuercen la atmósfera de fantasía, sonidos de armas que añadan realismo a los combates, efectos de habilidades para resaltar la ejecución de poderosas acciones y sonidos específicos para la interacción con enemigos y otros elementos del entorno.

El objetivo es que el jugador se sumerja no solo visualmente, sino también auditivamente en el mundo del juego, creando una experiencia envolvente y emocionante.

## 6. Demostración

### 6.1. Guía de usuario

Esta sección está diseñada para proporcionar una comprensión detallada del funcionamiento del videojuego RPG. A pesar de que la interfaz de usuario y otros contenidos están diseñados de manera intuitiva para facilitar la experiencia del usuario, la inclusión de esta sección es necesaria para abordar posibles dificultades que puedan surgir, especialmente para aquellos que se están iniciando en el mundo de la exploración de juegos RPG y pueden enfrentar desafíos en la comprensión de la funcionalidad del juego.

- **Como jugar con teclado y ratón:** los atajos se encuentran dispuestos de la siguiente manera, tal como se aprecia en la Figura 43.



Figura 43 Teclas de acceso para jugar

- **Movimiento principal:** El jugador puede mover el personaje con las siguientes teclas:
  - **W, A, S, D:** Es el movimiento principal para poder desplazar hacia todas las direcciones posibles
  - **Q, E:** es para brindar un movimiento lateral hacia al frente
- **Movimiento secundario:** El jugador puede mover el personaje con las siguientes teclas secundarias:
  - **[↑], [↓], [←] y [→]:** Es el movimiento secundario al igual que el inicial funcionan de la misma manera.
- **Realizar salto:** la tecla espacio se emplea para realizar saltos.

- **Habilidades:** El personaje dispone de diez habilidades, mismas que se encuentran ordenadas en función al teclado número superior.
- **Rotar cámara:** Se necesita mantener presionado para realizar este tipo de acciones, donde la cámara órbita alrededor del personaje.
- **Rotar cámara y Pj:** se necesita mantener presionado el clic esto posibilita además de rotar la cámara, hacer que el personaje pueda dar seguimiento a la rotación.



# Conclusiones y líneas de futuro

## 6.2. Conclusiones

Se logró implementar con éxito un sistema de combate que se inspira en elementos tácticos de héroes y jefes de mazmorras de World of Warcraft. La toma de decisiones tácticas durante los enfrentamientos con personajes no jugables ha sido un logro destacado.

La elección de la metodología ágil Scrum demostró ser efectiva para la gestión y planificación del proyecto. Facilitó un seguimiento continuo y una organización eficiente del tiempo, permitiendo obtener un producto final de alta calidad.

La creación de un ambiente de fantasía, con un estilo artístico Stylized, contribuyó a una ambientación clara y detallista. El escenario, dividido en diversas zonas, ha permitido una progresión dinámica para los jugadores.

No fue posible integrar las diversas clases propuestas en el TFM debido a que, aunque se cuenta con la lógica necesaria, el desarrollo y ajuste de habilidades, movimientos y otros elementos propios de cada clase demanda tiempo. Es por ello que el producto no pudo cumplir con este aspecto, al igual que la integración de ciertas características presentes en varios juegos RPG, como un sistema de progresión de personaje.

Al implementar el sistema de recompensas que proporciona botín al eliminar criaturas, añadiendo un componente gratificante al juego. La gestión eficiente de los ítems y su integración en el inventario fue un aspecto clave. Debido a que esto es parte esencial para un juego de estrategia.

## 6.3. Líneas de futuro

Para futuras iteraciones del juego, se plantea una expansión significativa que llevará la experiencia de juego a un nuevo nivel.

Las líneas futuras incluirán la implementación de un modo multijugador, permitiendo a los jugadores enfrentarse entre sí o colaborar en equipos para enfrentar desafíos más complejos, esto busca ampliar el universo del juego mediante la inclusión de un mayor número de NPC, cada uno con sus propias historias y desafíos únicos.

Asimismo, se pretende generar nuevos jefes (bosses) que presenten desafíos aún más emocionantes y estratégicos, exigiendo a los jugadores desarrollar tácticas específicas para superarlos. La

integración de las clases propuestas inicialmente se llevará a cabo, brindando a los jugadores una gama más amplia de opciones y estrategias para abordar el juego.

Se puede implementar un sistema de cadenas de misiones para proporcionar a los jugadores una narrativa más profunda y oportunidades adicionales de exploración. Estas cadenas de misiones ofrecerán historias ramificadas y desafíos variados, permitiendo a los jugadores sumergirse aún más en el mundo del juego y tomar decisiones que afecten directamente el desarrollo de sus personajes.

Estas líneas futuras tienen como objetivo enriquecer la experiencia del jugador, ofreciendo nuevas posibilidades de exploración, combate y colaboración. Se buscará mantener un equilibrio entre desafíos atractivos y la diversidad de opciones, proporcionando así una experiencia de juego continuamente emocionante y envolvente.



## Bibliografía

- [1] Mota D. Experiencia de usuarios en los videojuegos RPG (Role Playing Game) - Paradigma 2023. <https://www.paradigmadigital.com/dev/experiencia-usuario-videojuegos-rpg/> (accessed November 20, 2023).
- [2] Navarro Cadavid A, Fernández Martínez JD, Morales Vélez J. Revisión de metodologías ágiles para el desarrollo de software. Prospect ISSN-e 2216-1368, ISSN 1692-8261, Vol 11, N° 2 (Julio-Diciembre), 2013, Págs 30-39 2013;11:30–9.
- [3] Trigás Gallego M. Metodología Scrum 2012.
- [4] Atlassian. Qué es Trello: descubre sus funciones, usos y todo lo que ofrece | Trello n.d. <https://trello.com/es/tour> (accessed November 21, 2023).
- [5] Compton C. What is an RPG? 2018. <https://www.gamedeveloper.com/design/what-is-an-rpg> (accessed November 28, 2023).
- [6] Vogel J. The Glorious, Profitable, Inescapable Art of Addiction 2019. <https://www.gamedeveloper.com/business/the-glorious-profitable-inescapable-art-of-addiction> (accessed November 28, 2023).
- [7] Koster R. Ten Years of World of Warcraft 2014. <https://www.gamedeveloper.com/design/ten-years-of-world-of-warcraft> (accessed November 28, 2023).
- [8] Antognoli D. Transformation: Avatars, Role-playing, and Narrative Perspective in Games 2023. <https://www.gamedeveloper.com/blogs/transformation-avatars-role-playing-and-narrative-perspective-in-games> (accessed November 28, 2023).
- [9] Wei H, Bizzocchi J, Calvert T. Time and space in digital game storytelling. Int J Comput Games Technol 2010. <https://doi.org/10.1155/2010/897217>.
- [10] Sim B. 10 Interesting Game Mechanics Every Game Developer Should Know 2017. <https://www.gamedeveloper.com/production/10-interesting-game-mechanics-every-game-developer-should-know> (accessed November 28, 2023).
- [11] Kinch J. Game Studies: Physical Elements of Play 2021. <https://www.gamedeveloper.com/design/game-studies-physical-elements-of-play> (accessed November 28, 2023).
- [12] Buttfeld-Addison M. Unity Game Engine: A Comprehensive Guide 2023. <https://www.gamedeveloper.com/production/a-comprehensive-guide-to-the-unity-2d-3d-game-engine> (accessed November 28, 2023).
- [13] Liu Y. The Emotional Experience of Interaction with NPCs in RPG Games 2022. <https://www.gamedeveloper.com/blogs/the-emotional-experience-of-interaction-with-npcs-in-rpg-games> (accessed November 28, 2023).
- [14] Marrone M. Game Dev Digest Issue #81 - 2D, Animation, and more 2021. <https://www.gamedeveloper.com/audio/game-dev-digest-issue-81---2d-animation-and-more> (accessed November 29, 2023).

- [15] Hill C. Composing Music for an Indie Video Game 2020.  
<https://www.gamedeveloper.com/audio/composing-music-for-an-indie-video-game> (accessed November 29, 2023).
- [16] Plut C, Pasquier P. A framework for generative and adaptive game music 2022.  
<https://www.gamedeveloper.com/audio/deep-dive-generative-music-in-video-games> (accessed November 29, 2023).
- [17] Unity. Plataforma de desarrollo en tiempo real de Unity | Motor de 3D, 2D, VR y AR 2023.  
<https://unity.com/es> (accessed November 29, 2023).
- [18] Dickson PE, Block JE, Echevarria GN, Keenan KC. An experience-based comparison of unity and unreal for a stand-alone 3D game development course. Annu Conf Innov Technol Comput Sci Educ ITiCSE 2017;Part F128680:70–5. <https://doi.org/10.1145/3059009.3059013>.
- [19] Unreal Engine. La herramienta de creación 3D en tiempo real más potente - Unreal Engine 2023. <https://www.unrealengine.com/es-ES> (accessed December 1, 2023).