



Instituto Politécnico Nacional  
Escuela Superior de Cómputo



Ingeniería en Inteligencia Artificial, Análisis y diseño de sistemas  
Sem: 2024-2, 4BM2, Reporte técnico, Fecha: 27-06-24

## ECOS DEL ALMA: PROTOTIPO DE JUEGO ROGUE LITE CON INTERACCIÓN JUGADOR-NPC DINÁMICA

*Presentan*

Angeles López Erick Jesse<sup>1</sup>  
Corona Anchondo José Antonio<sup>2</sup>  
Espinosa Martínez José Carlos<sup>3</sup>  
Esquivel García Thania Paola<sup>4</sup>

**Resumen:** Ecos del alma es el prototipo de un videojuego 2D pixel art estilo “Roguelite” con generación procedural y metapersistencia de progreso. Este proyecto implementa tecnología de procesamiento de lenguaje natural (NLP) para maximizar el nivel conversacional entre el jugador y los personajes no jugables (NPCs). El objetivo es generar y extraer conversaciones que puedan servir como recurso para el análisis terapéutico. A través de esta innovación, se busca ofrecer una herramienta valiosa tanto para la mejora de la experiencia del jugador como para el apoyo a profesionales en el campo de la salud mental.

**Palabras Clave:** Conversación, inteligencia artificial, personaje no jugable, procesamiento del lenguaje natural, procedural, salud mental, videojuego.

---

<sup>1</sup>eangeles11700@alumno.ipn.mx

<sup>2</sup>jcoronaa1900@alumno.ipn.mx

<sup>3</sup>jespinosam1900@alumno.ipn.mx

<sup>4</sup>tesquivelg2000@alumno.ipn.mx

# Índice

<b>1. Introducción</b>	<b>4</b>
<b>2. Estado del arte</b>	<b>5</b>
<b>3. Justificación</b>	<b>6</b>
<b>4. Objetivo general</b>	<b>7</b>
<b>5. Objetivos particulares</b>	<b>7</b>
<b>6. Marco teórico</b>	<b>8</b>
6.1. Videojuego . . . . .	8
6.2. Procesamiento del lenguaje natural . . . . .	8
6.2.1. Detección de emociones . . . . .	9
6.2.2. Análisis léxico . . . . .	9
6.2.3. Aprendizaje automático . . . . .	9
6.3. Salud mental . . . . .	9
6.3.1. Emociones . . . . .	10
6.4. Tratamiento terapéutico psicológico . . . . .	11
<b>7. Análisis</b>	<b>12</b>
7.1. Metodología . . . . .	12
7.2. Tecnología . . . . .	12
7.2.1. Motor de desarrollo Unity . . . . .	12
7.2.2. Lenguaje de programación C# . . . . .	13
7.2.3. Lenguaje de programación Prolog . . . . .	13
7.2.4. MongoDB . . . . .	13
7.2.5. Star UML . . . . .	14
7.2.6. Pixilart . . . . .	14
7.3. Requerimientos . . . . .	15
7.3.1. Requerimientos funcionales . . . . .	15
7.3.2. Requerimientos no funcionales . . . . .	16
7.3.3. Requerimientos mínimos de sistema . . . . .	18
<b>8. Diseño</b>	<b>19</b>
8.1. Diagrama de contexto . . . . .	19
8.2. Diagrama de flujo de datos . . . . .	19
8.3. Diagrama de casos de uso . . . . .	20
8.3.1. Especificaciones de casos de uso del jugador(personaje) . . . . .	21
8.4. Diagrama de clases . . . . .	31

<i>Formulación del Proyecto</i>	3
<b>9. Prototipo</b>	<b>32</b>
<b>10.Conclusiones</b>	<b>33</b>
<b>11.Referencias Bibliográficas</b>	<b>34</b>

## 1. Introducción

Los videojuegos han trascendido su papel en las últimas décadas no solo en el campo del entretenimiento, sino también en áreas como la educación, la formación profesional y la salud mental. Las recientes tecnologías de inteligencia artificial han dado cabida a nuevas experiencias inmersivas y significativas para los jugadores. Sin embargo, muchas veces los niveles de comunicación están limitados a flujos conversacionales predefinidos y no procedurales, lo que representa un área de oportunidad para implementar procesamiento del lenguaje natural (NLP) y mejorar la comunicación con los sistemas inteligentes.

La posibilidad de generar material para un especialista que busca conocer la interacción de un jugador y cómo este se comunica con las personas se incrementa con sistemas inteligentes y mayor libertad de comunicación. Aunque existe un vacío en la investigación de videojuegos terapéuticos, este proyecto no pretende realizar diagnósticos ni prediagnósticos. En cambio, busca desarrollar una experiencia inmersiva y personalizada para el jugador basada en sus interacciones con los personajes no jugables (NPCs) dentro del juego.

Este enfoque no solo mejorará la experiencia del jugador, sino que también enriquecerá el repertorio de recursos disponibles para los especialistas en psicología clínica, abriendo nuevas posibilidades para el análisis y apoyo a la salud mental de los jugadores.

A través de este proyecto, se espera demostrar cómo la tecnología de videojuegos y la inteligencia artificial pueden converger para ofrecer herramientas innovadoras y efectivas en el campo de la salud mental, proporcionando nuevos recursos que pueden ser utilizados por terapeutas y otros profesionales de la salud para mejorar la calidad de vida de los pacientes.

Mediante experiencias entretenidas y únicas, se busca cautivar a los jugadores, ofreciéndoles no solo un entretenimiento de calidad, sino también un recurso potencialmente valioso para el ámbito terapéutico.

## 2. Estado del arte

Algunos juegos y aplicaciones similares son:

1. Videojuego *Hades* desarrollado por Supergiant Games [1].
2. Videojuego *Celeste* desarrollado por Maddy Makes Games [2]
3. Videojuego *Hollow Knight* desarrollado por Team Cherry [3]
4. Videojuego *Coffee Talk* desarrollado por Toge Productions [4]
5. Aplicación *character.ai* desarrollada por Character Technologies, Inc. [5]

Producto	Características
Hades	Juego <i>Roguelite</i> con niveles generados proceduralmente conectando diferentes habitaciones, caracterizado por la acción rápida y la narrativa profunda.
Celeste	Juego de plataformas con jugabilidad invariable, conversación ocasional con NPCs y mecánicas de movimiento avanzadas, centrado en la superación personal.
Hollow knight	Juego <i>metroidvania</i> con poca interacción conversacional, mecánicas de movimiento y combate avanzadas y un mundo expansivo que fomenta la exploración.
Coffee talk	Juego de género novela visual y aventura conversacional, donde las respuestas (limitadas por el juego) definen las relaciones con los NPCs.
character.ai	Aplicación online para diseñar o conversar con bots inteligentes y personificados únicamente mediante líneas de texto, utilizada para crear experiencias interactivas y personalizadas.

Cuadro 1: Resumen de juegos y aplicaciones similares

### 3. Justificación

La Organización Mundial de la Salud (OMS) y la Organización Panamericana de la Salud (OPS) señalan que uno de cada siete menores de edad enfrenta alteraciones cognitivas, incluyendo trastornos como el TDAH, la ansiedad y la depresión. En el año 2021, el Instituto Nacional de Estadística y Geografía (INEGI) reportó más de ocho mil suicidios en menores de edad en México, con un alarmante 6.9 % de adolescentes que experimentaron pensamientos suicidas y un 6 % que intentó quitarse la vida [6].

La naturaleza de muchos videojuegos requiere la intervención de sistemas inteligentes capaces de desafiar y confundir al jugador. Estos sistemas, como los NPCs, utilizan algoritmos de inteligencia artificial para cumplir su rol dentro del juego, a menudo limitados por flujos de comportamiento preestablecidos durante el diseño [7].

La implementación de videojuegos terapéuticos sigue siendo un campo de exploración amplio y prometedor. Estos juegos tienen el potencial de motivar a los pacientes, reducir el abandono del tratamiento y facilitar la práctica controlada. Sin embargo, también plantean riesgos como la adicción y la influencia en las expectativas de los usuarios [8].

El desarrollo de un videojuego que permita una comunicación más dinámica con los NPCs mediante el uso de procesamiento de lenguaje natural (NLP) es crucial. Este enfoque no solo mejorará la experiencia del jugador, sino que también proporcionará datos valiosos para especialistas terapéuticos al analizar la forma de comunicación del jugador de manera más auténtica y sin limitaciones predefinidas.

No se busca crear un juego terapéutico ni un diagnosa virtual, sino explorar un nuevo enfoque en la implementación de algoritmos de NLP dentro de los videojuegos. Esto permitirá desarrollar herramientas que enriquezcan el repertorio de recursos disponibles para especialistas, abriendo nuevas posibilidades en el campo de la psicología clínica.

## 4. Objetivo general

Desarrollar el prototipo de un videojuego *rogue lite* con generación de niveles procedurales e interacción jugador-NPC desde el teclado mediante análisis de texto, simulación de emociones y toma de decisiones.

## 5. Objetivos particulares

- Desarrollar un generador de niveles procedurales con habitaciones prediseñadas estilo *dungeon*.
- Diseñar, para los NPCs, factores de personalidad y flujos conversacionales dinámicos.
- Desarrollar un modelo de procesamiento de lenguaje natural para analizar respuestas del jugador y generar respuestas de los NPCs.
- Recopilar conversaciones y respuestas entre el jugador y los NPCy que sirvan como material de apoyo para especialistas.

## 6. Marco teórico

### 6.1. Videojuego

Keith Burgun define a un videojuego como un sistema de reglas donde los participantes compiten haciendo decisiones ambiciosas, además de que los contrincantes no necesariamente deben ser humanos [9].

Los géneros de videojuegos son una clasificación no oficial que puede agrupar diferentes juegos con base en atributos como la jugabilidad, mecánicas, perspectiva e historia.

Para este proyecto se busca desarrollar un videojuego estilo mazmorras, conocido como “roguelite”, un subgénero de los juegos de rol y una derivación de los “roguelikes”. Este se caracteriza por:

- **Generación Procedural:** Los niveles de las mazmorras se generan automáticamente, ofreciendo experiencias únicas en cada partida.
- **Muerte Permanente:** Cuando el jugador muere dentro de la mazmorra, pierde todo su progreso en esa partida específica.
- **Dificultad:** Cada derrota se considera una experiencia de aprendizaje.

La diferencia fundamental entre un “roguelike” y un “roguelite” radica en la metapersistencia del progreso en el juego. En el primero, el jugador siempre comienza con estadísticas iniciales y la dificultad de la mazmorra no aumenta con cada partida; la experiencia es la única ventaja. En cambio, el “roguelite” permite cambios estadísticos que pueden modificar la dificultad o permitir la progresión en la historia. Aunque se pierde al morir, no todo el progreso se borra, lo que fomenta un sentido de avance y aprendizaje continuo a través de múltiples sesiones de juego [10].

### 6.2. Procesamiento del lenguaje natural

El procesamiento del lenguaje natural (PLN) es una rama de la inteligencia artificial que incluye técnicas, algoritmos y tecnologías para estructurar y dar significado a los textos mediante el aprendizaje automático [11].

El PLN se aplica en diversas áreas:

- **Reconocimiento de voz:** Identifica aspectos del habla como velocidad, entonación, énfasis, acentos, gramática, etc.
- **Desambiguación:** Establece el significado de palabras polisémicas<sup>5</sup>.
- **Etiquetado:** Asigna etiquetas gramaticales a palabras o fragmentos de texto según su uso (palabras homónimas<sup>6</sup>).

---

<sup>5</sup>Palabras con más de un significado dependiendo del contexto [12]

<sup>6</sup>Palabras que, con respecto a otra, tienen el mismo nombre [13]



- **Resolución de correferencias:** Identifica si dos o más palabras se refieren a la misma entidad.
- **Análisis de sentimientos:** Extrae cualidades subjetivas como actitudes, emociones, sarcasmo, confusión, sospecha, etc.
- **Generación de lenguaje natural:** Transforma información estructurada en lenguaje comprensible para humanos [14].

### 6.2.1. Detección de emociones

El análisis de sentimientos utiliza diversas técnicas, incluyendo el análisis léxico y el aprendizaje automático, para evaluar la satisfacción del cliente, analizar opiniones públicas, predecir comportamientos del consumidor, mejorar la atención al cliente, analizar redes sociales, chatbots y traducción automática basada en el contenido y la forma de los mensajes [15].

### 6.2.2. Análisis léxico

Se utiliza un diccionario de palabras con atributos de sentimientos para asignar una puntuación a cada palabra según su carga emocional. Estas puntuaciones se suman para determinar el sentimiento general del texto [15].

Como se observa en la figura 1, el analizador tiene múltiples componentes que se encargan de:

- Convertir el código fuente en ‘tokens’<sup>7</sup>.
- Agrupar mediante una estructura de datos tipo árbol los ‘tokens’ para analizar la semántica.
- Gestiona los errores y optimiza el código generado para obtener el código objetivo.

### 6.2.3. Aprendizaje automático

El aprendizaje automático (Machine Learning, ML) se centra en el desarrollo de sistemas que mejoran su rendimiento en función de los datos consumidos [17]. Mediante modelos de ML como clasificadores y redes neuronales, entrenados con datos etiquetados, se pueden reconocer patrones y asociarlos con fines emocionales en el contexto del PLN [15].

## 6.3. Salud mental

La salud mental es un estado de bienestar que permite a las personas enfrentar el estrés, desarrollar habilidades, aprender y trabajar de manera adecuada.

---

<sup>7</sup>Agrupación de características constituida por símbolos que forman sentencias del lenguaje como palabras reservadas, identificadores, símbolos, etc.

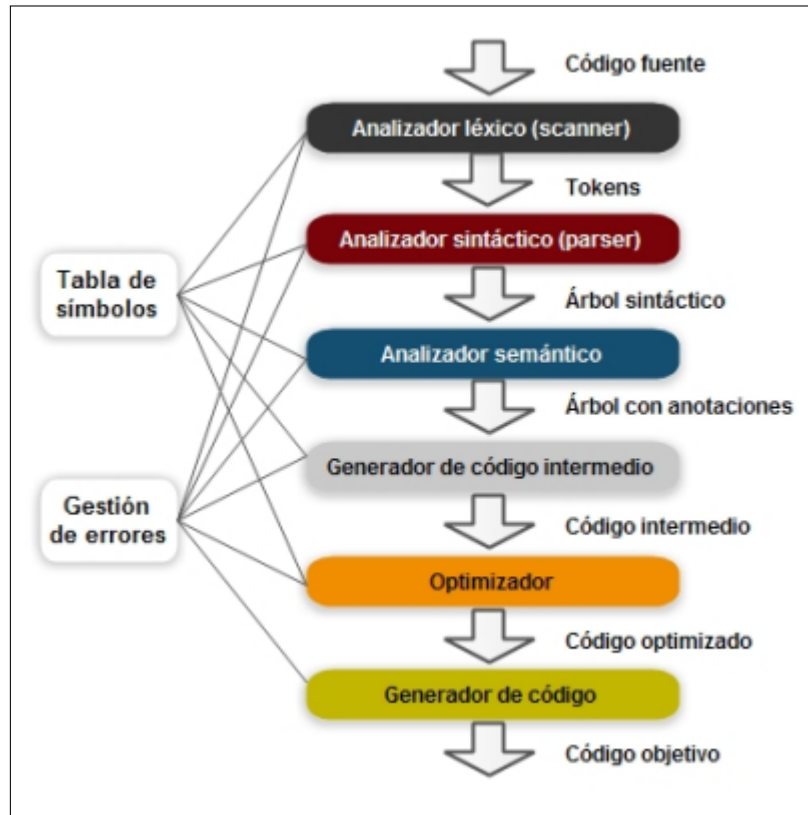


Figura 1: Proceso del analizador léxico [16].

Está determinada por factores psicológicos, biológicos individuales, circunstancias sociales, económicas, políticas y ambientales. La resiliencia se ve fortalecida por interacciones sociales positivas, educación, trabajo decente, seguridad, entre otros [18].

### 6.3.1. Emociones

Las emociones son la manera natural en que los seres humanos reaccionan ante las situaciones de su entorno. Estas se presentan en diferentes intensidades a lo largo del día [19].

Las emociones pueden afectar la salud mental de formas positivas y negativas. Para tener buena salud mental es necesario equilibrar y controlar nuestras emociones. El análisis de emociones puede ser mediante las acciones o su conversación, un experto en el área puede determinar como ciertas influencias ajenas e internas alteran las emociones y por ende, la salud mental [20].

## 6.4. Tratamiento terapéutico psicológico

La terapia es una rama de la medicina diseñada para tratar enfermedades, mejorar la salud de los pacientes, aliviar síntomas, mejorar funciones física o psicológicas y, en caso de ser posible, erradicar la afección. La terapia puede dividirse principalmente en dos grandes grupos: farmacológico y no farmacológico. El éxito o el fracaso de cualquier terapia se ve beneficiado por un análisis preciso, seguimiento de tratamiento y participación activa del paciente [21].

la psicología es una ciencia que estudia los procesos mentales cognitivos, afectivos y conductuales. Busca explicar el comportamiento humano para intentar predecir acciones futuras [22].

La psicoterapia es una subrama de la psicología y la terapia enfocada al tratamiento de afecciones mentales. Su objetivo es, mediante la conversación con un psicólogo o psiquiatra, aprender sobre problemas específicos, pensamientos, emociones y comportamientos que alteren el estado de ánimo de un individuo. Se busca que el paciente sea pueda llevar plenamente el control de su vida y que aprenda a resolver situaciones adversas.

La psicoterapia puede tratar problemas mentales como:

- Trastornos de ansiedad
- Trastornos de estados de ánimo
- Adicciones
- Trastornos de alimentación
- Trastornos de personalidad
- Esquizofrenia

Aunque puede ser beneficiosa con enfermedades diagnosticadas, resulta útil con ‘conflictos’ cotidianos, tales como:

- Resolver conflictos
- Aliviar ansiedad o estrés
- Enfrentar cambios importantes
- Control de emociones
- Recuperación de abuso o traumas [23]

Con este proyecto se busca que, las conversaciones generadas entre el NPC y el flujo conversacional permita generar material de apoyo sobre el jugador en situaciones específicas. Este material no busca detectar enfermedades mentales ni fungir como psicólogo, busca generar un recurso que pueda ser útil para un especialista.

## 7. Análisis

### 7.1. Metodología

La metodología para el proyecto es el modelo de entrega por etapas. Este modelo nos permite planificar y diseñar componentes de forma individual siempre orientados a los mismos objetivos.

El videojuego puede tener productos funcionales en las primeras etapas de desarrollo. Es por esto por lo que el modelo de entrega por etapas maximiza la productividad, permitiendo desarrollar elementos adicionales en cada una de sus etapas.

Estas implementaciones pueden ser desde niveles adicionales, mecánicas nuevas, correcciones y expansiones del mismo proyecto, permitiendo diversidad y adaptabilidad a pequeños cambios.

### 7.2. Tecnología

#### 7.2.1. Motor de desarrollo Unity

Unity es un motor y una plataforma de desarrollo de videojuegos para diseñar contenido audiovisual en 2D y 3D. Permite a programadores, artistas, arquitectos, diseñadores, cineastas, etc. diseñar, construir y desplegar sus propios proyectos en diferentes dispositivos [24].

Se opta por Unity sobre otras herramientas populares en el mercado como lo son Godot, Unreal Engine y Game Maker por las siguientes razones:

- Unity es el motor estandarizado para el desarrollo de proyectos nuevos, con una amplia comunidad y numerosos recursos disponibles.
- El juego requiere de *sprites* estilo *pixel art* que pueden ser segmentados y animados fácilmente mediante el IDE de Unity.
- El juego no requiere mucho procesamiento en los gráficos, por lo que Unity resulta una opción más cómoda y eficiente frente a Unreal Engine, que está más orientado a gráficos de alta calidad y realismo.
- Unity ofrece una excelente documentación y soporte técnico, lo que facilita la resolución de problemas y la implementación de características avanzadas.
- El editor visual de Unity es intuitivo y facilita la creación de niveles y la implementación de mecánicas de juego complejas sin requerir conocimientos avanzados en programación.
- Unity proporciona un excelente sistema de física y colisiones que puede ser aprovechado para crear interacciones detalladas y precisas en el juego.

### 7.2.2. Lenguaje de programación C#

Lenguaje de programación utilizado comúnmente en Unity para el desarrollo de videojuegos. Es un lenguaje orientado a objetos con un sistema de gestión automática de memoria [25].

### 7.2.3. Lenguaje de programación Prolog

Lenguaje de programación basado en declaración de hechos, establecimiento de reglas e inferencias. Permite generar una base de conocimientos relacionando diferentes hechos mediante secuencias lógicas enlazadas [26].

Su uso sera como una librería a la cual podremos acceder desde C# para generar la base de conocimientos y la generación de inferencias.

### 7.2.4. MongoDB

MongoDB es un sistema gestor de base de datos NoSQL orientado a documentos con estructuras BSON y esquematización dinámica.

- Incluye medidas de seguridad robustas como:
  - **Autenticación:** Registro mediante credenciales de Atlas, GitHub o Google, lo que garantiza un acceso controlado y seguro a la base de datos.
  - **Autorización:** Creación y asignación de roles y permisos para controlar el acceso y las operaciones que cada usuario puede realizar.
  - **Auditoría:** Monitoreo detallado de acciones en el entorno de MongoDB, permitiendo rastrear cambios y accesos para garantizar la seguridad y el cumplimiento de normas.
  - **Encriptación:** Protección de datos tanto en reposo como en tránsito, asegurando que la información sensible esté siempre protegida contra accesos no autorizados.
- Ofrece tiempos de consulta reducidos en comparación con sistemas relacionales, lo que mejora significativamente el rendimiento en aplicaciones con grandes volúmenes de datos.
- Escalabilidad horizontal y vertical, permitiendo ajustar la capacidad de la base de datos según las necesidades del proyecto. Esto es especialmente útil para aplicaciones que requieren manejar grandes cantidades de datos y usuarios.
- Herramientas de transcripción de modelos SQL a NoSQL, facilitando la migración y la integración de bases de datos existentes.
- Conectividad con diferentes herramientas y sistemas de gestión, proporcionando flexibilidad y facilitando la integración con otros componentes del ecosistema tecnológico del proyecto.

- Integración con frameworks de desarrollo populares, lo que facilita su uso en diversas aplicaciones y entornos de desarrollo [27].

#### 7.2.5. Star UML

Herramienta de modelado UML desarrollado por MKLab. Es una de las herramientas UML más populares en el mundo. Se ha descargado más de 4 millones de personas y se utiliza en más de 150 países.

Es compatible con los principales lenguajes de programación como Java, C# y C ++. Puede generar códigos fuente de sus modelos o construir un modelo a partir de código fuente mediante ingeniería inversa [28].

#### 7.2.6. Pixilart

Aplicación web y red social que permite diseñar, modificar, publicar y compartir imagenes estilo *pixel art* de forma gratuita [29].

Esta plataforma nos permite crear grupos privados donde se puedan compartir y editar diferentes diseños. Cuenta con un amplio abanico de herramientas para colorear y dibujar en cuadrícula, permitiendo formatos de alta calidad.

## 7.3. Requerimientos

### 7.3.1. Requerimientos funcionales

#### Respecto a la jugabilidad

- RF1 Al iniciar, el juego muestra el menú principal con opciones para cargar partida, borrar partida, salir y acceder a la configuración.
- RF2 El jugador controla las acciones y las interacciones del personaje jugable.
- RF3 Al inicio, el juego sitúa al jugador y a un número exacto de NPCs en el *respawn*.
- RF4 El jugador comienza un intento de progresión.
- RF5 Cuando el jugador inicia el intento o termina un nivel previo, el juego genera de forma procedural el siguiente nivel, compuesto por un número finito de habitaciones con obstáculos.
- RF6 Cuando el jugador termina el último nivel, regresa al *respawn*.
- RF7 Al finalizar cada nivel, el juego presenta una habitación con un NPC.
- RF8 Cuando el personaje agota sus vidas, reaparece con el último NPC con el que interactuó y entra en modo “after” (conservando los mismos controles), y sus estadísticas se determinan con base en el flujo de conversación.
- RF9 Si el personaje encuentra su lugar de muerte en modo “after”, regresa al modo normal.
- RF10 Si el personaje muere en modo “after”, vuelve al *respawn* con el último NPC con el que interactuó ahí, y sus estadísticas se ajustan de acuerdo con el flujo de conversación.

#### Respecto a la configuración

- RF11 El jugador puede configurar manualmente todos los controles o elegir la configuración predeterminada.
- RF12 Se permite al jugador controlar el volumen general del juego, los efectos y la música.
- RF13 Al terminar la interacción con un personaje, el juego guarda los datos de la partida.
- RF14 Al volver al menú principal, el jugador guarda los datos de la partida.

#### Respecto a los controles

- RF15 **Movimiento lateral:** El personaje puede desplazarse a lo largo del plano 2D.
- RF16 **Salto:** El personaje efectúa un salto vertical independiente del movimiento lateral, limitado por colisiones.
- RF17 **Dash:** El personaje realiza un impulso rápido en la dirección elegida por el jugador, que tiene un período de enfriamiento antes de poder reutilizarse.

#### **Respecto a la interacción con el entorno**

- RF18 **Conversación:** El personaje puede iniciar conversaciones con NPCs.
- RF19 **Finalizar conversación:** El jugador puede cerrar la ventana de diálogo con el NPC.
- RF20 **Cambio de escenario:** El personaje puede interactuar con puertas (o agujeros) para entrar, salir o cambiar de escenario.
- RF21 **Sistema de vidas:** El personaje pierde salud al colisionar con obstáculos y la recupera al completar habitaciones.

#### **Respecto a la interacción con el sistema inteligente**

- RF22 Cuando el jugador interactúa con un NPC, este responde según el flujo de conversación actualizado.
- RF23 El jugador introduce su respuesta en la conversación mediante el teclado.
- RF24 Con la respuesta del jugador, el juego procesa el mensaje, genera una respuesta y actualiza el flujo de conversación.
- RF25 El jugador tiene un número máximo de respuestas por encuentro con cada NPC.

#### **7.3.2. Requerimientos no funcionales**

- RNF1 **Seguridad de Acceso:** Solo los jugadores registrados podrán acceder al juego, asegurando una experiencia segura y personalizada.
- RNF2 **Personalización Basada en el Perfil del Jugador:** El contenido y las opciones disponibles en el juego variarán según el progreso y las acciones del jugador.
- RNF3 **Conexión a Internet para Funcionalidades del sistema inteligente:** Se requerirá conexión a internet para acceder al servidor que contiene el sistema inteligente, permitiendo la interacción dinámica con NPC. El juego no incluirá características multiplayer online.



- RNF4 **Mensajes de Error Claros:** Los mensajes de error serán informativos, ayudando a los desarrolladores a entender y resolver problemas rápidamente.
- RNF5 **Diseño Adaptable a Diferentes Resoluciones y Plataformas de Escritorio:** El juego, diseñado exclusivamente para escritorio (Windows y macOS), adaptará su interfaz y gráficos a diferentes resoluciones de pantalla.
- RNF6 **Usabilidad:** El juego será intuitivo y fácil de navegar, ofreciendo una experiencia agradable para un amplio rango de jugadores.
- RNF7 **Terminología Amigable:** Se utilizará un lenguaje claro y accesible, evitando jerga complicada para facilitar la comprensión del juego.
- RNF8 **Tiempo de Generación de Contenido Dinámico:** Contenidos como diálogos o eventos se generarán en menos de 10 segundos, manteniendo el flujo del juego activo.
- RNF9 **Generación Simultánea de Contenidos:** El juego será capaz de generar múltiples elementos de juego de forma simultánea sin degradar el rendimiento.
- RNF10 **Integridad del Contenido Generado:** Se asegurará que el contenido generado, como diálogos o eventos, sea relevante y libre de errores significativos.
- RNF11 **Feedback Visual Durante Cargas:** Se proporcionarán indicadores de progreso durante las cargas para mantener informados a los jugadores.
- RNF12 **Accesibilidad y Configuración de Teclado:** El juego permitirá a los usuarios configurar las teclas de acceso rápido y ajustar la sensibilidad del teclado.
- RNF13 **Latencia Baja en la Entrada del Teclado:** Se garantizará una latencia mínima en la respuesta a las entradas del teclado.
- RNF14 **Soporte para Atajos de Teclado:** El juego incluirá atajos de teclado intuitivos para acciones comunes.
- RNF15 **Estilo Visual Amigable y Pixel Art:** El diseño del juego, basado en pixel art, promoverá un estilo visual amigable y atractivo, adecuado para una amplia audiencia.
- RNF16 **Identificación Clara de Objetos y Elementos:** Tanto en el apartado visual como sonoro, los objetos y elementos importantes del juego serán fácilmente identificables, mejorando la experiencia de juego.

RNF17 **Juiciness en la Jugabilidad:** El juego incorpora elementos de "juiciness", ofreciendo retroalimentación inmediata y satisfactoria a las acciones del jugador a través de efectos visuales, sonoros y mecánicos que hacen que la experiencia de juego sea más dinámica y gratificante.

### 7.3.3. Requerimientos mínimos de sistema

Windows	
<b>Sistema operativo</b>	Windows 7 o mas reciente.
<b>Procesador</b>	Intel Core i3 M380.
<b>Memoria</b>	2GM de RAM.
<b>Gráficos</b>	Intel HD 4000.
<b>DirectX</b>	Versión 10.
<b>Almacenamiento</b>	1.5 GB de espacio disponible.

Cuadro 2: Requerimientos mínimos de sistema para Windows.

Macos	
<b>Sistema operativo</b>	Lion 10.7.5, 32/64 bits o más reciente.
<b>Procesador</b>	Intel Core i3 M380.
<b>Memoria</b>	2GM de RAM.
<b>Gráficos</b>	OpenGL 3.0 o superior.
<b>Almacenamiento</b>	1.5 GB de espacio disponible.

Cuadro 3: Requerimientos mínimos de sistema para Macos.

# 8. Diseño

## 8.1. Diagrama de contexto

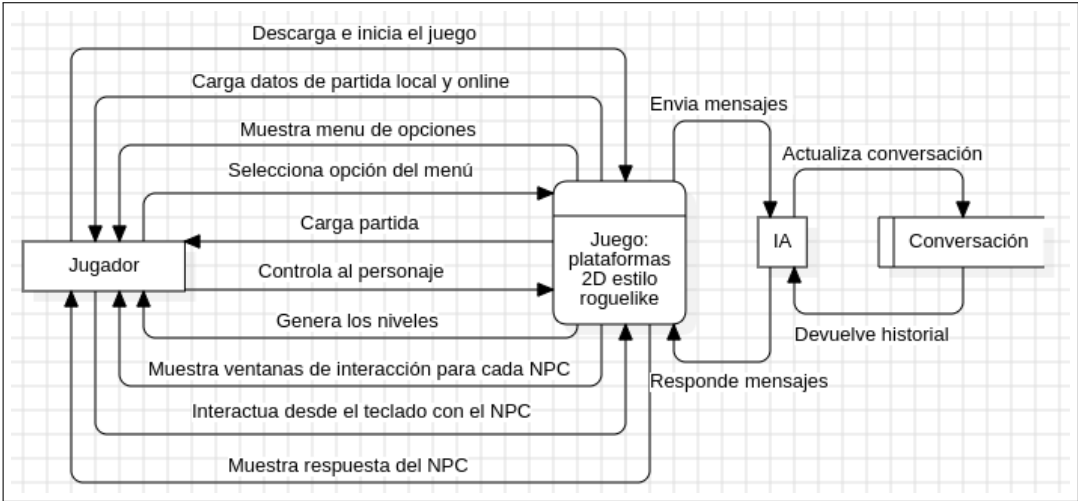


Figura 2: Diagrama de contexto

## 8.2. Diagrama de flujo de datos

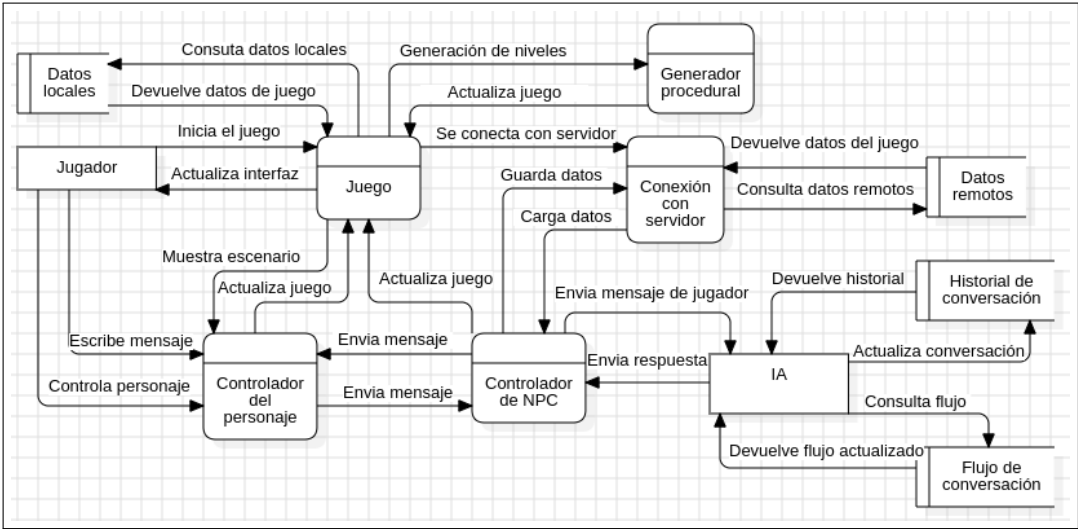


Figura 3: Diagrama de flujo de datos

8.3. Diagrama de casos de uso

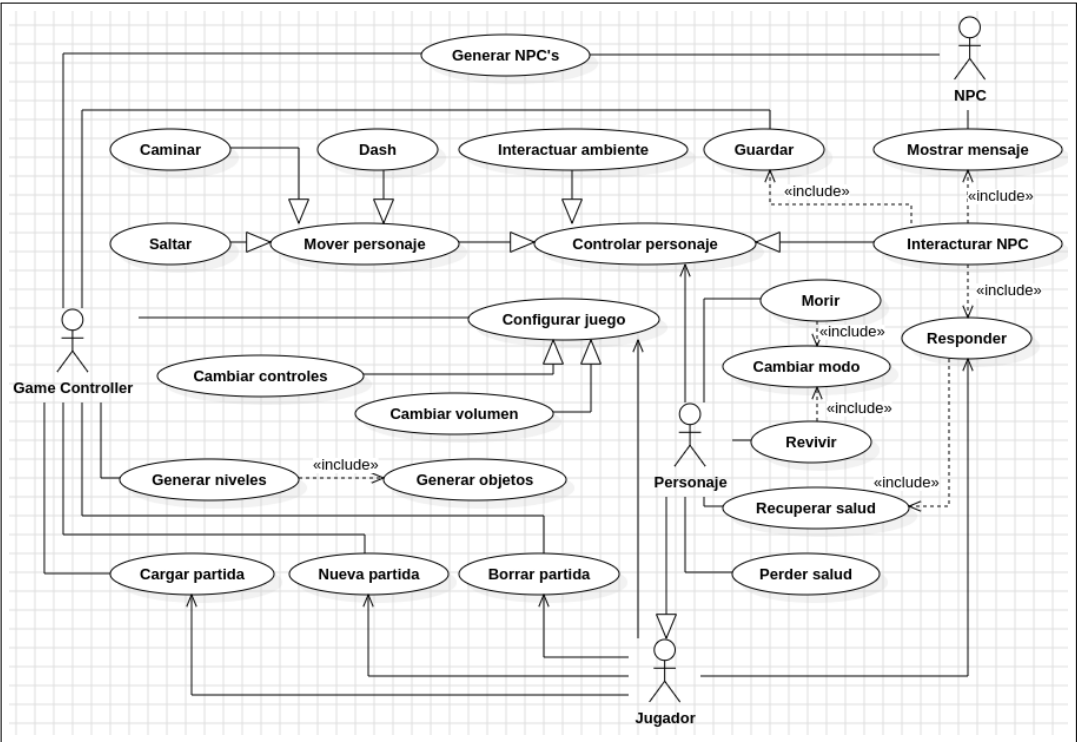


Figura 4: Diagrama de casos de uso

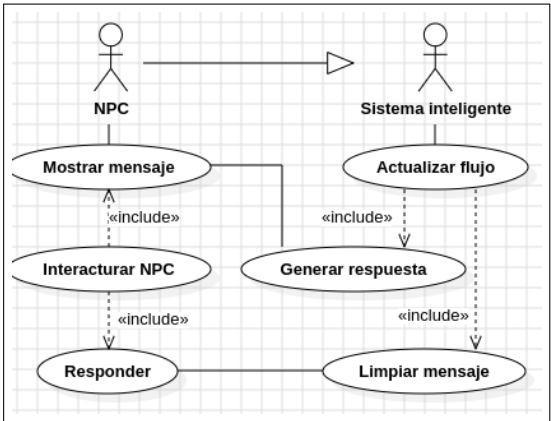


Figura 5: Diagrama de casos de uso

8.3.1. Especificaciones de casos de uso del jugador(personaje)



Figura 6: Caso de uso: Nueva partida

CASO DE USO	Nueva partida	
ACTOR	Jugador	
DESCRIPCIÓN	El jugador inicia una nueva partida en un ‘slot’ disponible.	
PRECONDICIÓN	Ninguna	
FLUJO BÁSICO	ACTOR	SISTEMA
	1. Presiona el botón de ‘Nueva partida’ en el menú principal. 3. Selecciona un ‘slot’ disponible.	2. Muestra ‘slots’ de almacenamiento. 4. Crea datos de partida e inicia el juego.
FLUJO ALTERNO	ACTOR	SISTEMA
1. Sin ‘slots’ de almacenamiento: Cancelar.	1. Presiona el botón de ‘Nueva partida’ en el menú principal. 3. Selecciona un ‘slot’ no disponible. 5. Selecciona la opción de cancelar.	2. Muestra ‘slots’ de almacenamiento. 4. Mensaje de advertencia para sobrescribir datos o cancelar. 6. Regresa al menú principal.
FLUJO ALTERNO	ACTOR	SISTEMA
2. Sin ‘slots’ de almacenamiento: Cancelar.	1. Presiona el botón de ‘Nueva partida’ en el menú principal. 3. Selecciona un ‘slot’ no disponible. 5. Selecciona la opción de sobrescribir.	2. Muestra ‘slots’ de almacenamiento. 4. Mensaje de advertencia para sobrescribir datos o cancelar. 6. Borra datos de partida pasada, crea datos de partida e inicia el juego.
POSTCONDICIÓN	Ninguna	

Cuadro 4: Descripción del caso de uso: Nueva partida.



Figura 7: Caso de uso: Cargar partida

CASO DE USO	Cargar partida	
ACTOR	Jugador	
DESCRIPCIÓN	El jugador carga los datos de la partida guardada e inicia el juego.	
PRECONDICIÓN	Nueva partida	
FLUJO BÁSICO	ACTOR	SISTEMA
	1. Selecciona el 'slot' de partida.	2. Carga datos de partida, e inicia el juego.
POSTCONDICIÓN	Ninguna	

Cuadro 5: Descripción del caso de uso: Cargar partida



Figura 8: Caso de uso: Borrar partida.

CASO DE USO	Borrar partida	
ACTOR	Jugador	
DESCRIPCIÓN	El jugador elimina los datos guardados de una partida.	
PRECONDICIÓN	Nueva partida	
FLUJO BÁSICO	ACTOR	SISTEMA
	1. Selecciona el botón de eliminar 'slot' de partida. 3. Selecciona eliminar los datos.	2. Mensaje de advertencia para eliminar o cancelar. 4. Elimina permanente los datos de partida del 'slot'.
FLUJO ALTERNO	ACTOR	SISTEMA
1. Cancelar.	1. Selecciona el botón de eliminar 'slot' de partida. 3. Selecciona cancelar.	2. Mensaje de advertencia para eliminar o cancelar. 4. Regresa al menú principal.
POSTCONDICIÓN	Ninguna	

Cuadro 6: Descripción del caso de uso: Eliminar partida

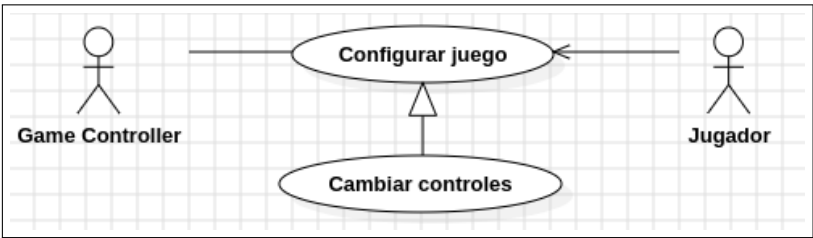


Figura 9: Caso de uso: Cambiar controles

<b>CASO DE USO</b>	Cambiar controles	
<b>ACTOR</b>	Jugador	
<b>DESCRIPCIÓN</b>	El jugador configura los controles para las acciones del personaje.	
<b>PRECONDICIÓN</b>	Ninguna	
<b>FLUJO BÁSICO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. Selecciona el menú de configuración. 3. Selecciona el sub menú de controles. 5. Selecciona una acción. 7. Presiona un botón en el tiempo dado.	2. Muestra configuración de audio y de controles. 4. Muestra un listado de acciones y su botón/tecla asignada. 6. Inicia un contador esperando respuesta del jugador. 8. Configura la acción al botón seleccionado.
<b>FLUJO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
1. Omisión.	1. Selecciona el menú de configuración. 3. Selecciona el sub menú de controles. 5. Selecciona una acción. 7. No presiona un botón en el tiempo dado.	2. Muestra configuración de audio y de controles. 4. Muestra un listado de acciones y su botón/tecla asignada. 6. Inicia un contador esperando respuesta del jugador. 8. Mantiene la configuración actual de la acción.
<b>POSTCONDICIÓN</b>	Ninguna	

Cuadro 7: Descripción del caso de uso: Cambiar controles



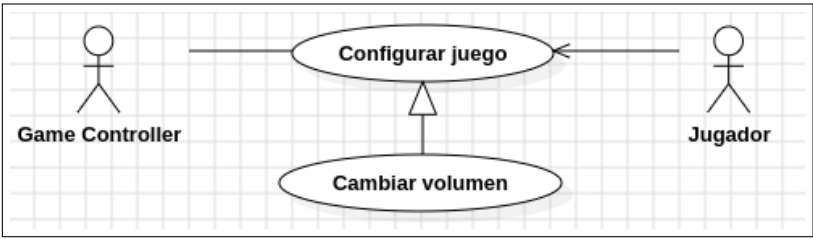


Figura 10: Caso de uso: Cambiar volumen

CASO DE USO	Cambiar volumen	
ACTOR	Jugador	
DESCRIPCIÓN	Se configuran los volúmenes del juego	
PRECONDICIÓN	Ninguna	
FLUJO BÁSICO	ACTOR	SISTEMA
	1. Selecciona el menú de configuración. 3. Selecciona el sub menú de volumen. 5. Selecciona y desplaza la barra de volumen específica.	2. Muestra configuración de audio y de controles. 4. Muestra barras deslizables de volumen. 6. Actualiza el volumen seleccionado.
POSTCONDICIÓN	Ninguna	

Cuadro 8: Descripción del caso de uso: Cambiar volumen

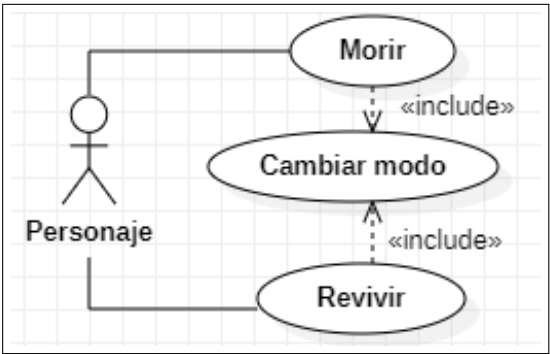


Figura 11: Caso de uso: Cambiar modo

<b>CASO DE USO</b>	Cambiar modo	
<b>ACTOR</b>	Personaje (Jugador)	
<b>DESCRIPCIÓN</b>	El personaje tiene un segundo intento después de morir y lo recupera cuando revive	
<b>PRECONDICIÓN</b>	Morir o Revivir	
<b>FLUJO BÁSICO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. El personaje muere. 2. El personaje encuentra el lugar donde falleció.	2. El personaje reaparece con el ultimo NPC con el que interactuó (Sus estadísticas están determinadas por el flujo conversacional del NPC) en modo 'after'. 3. El personaje recupera sus estadísticas iniciales y continua el juego.
<b>FLUJO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
1. El jugador fallece en modo 'after'.	1. El personaje muere. 2. El personaje muere nuevamente.	2. El personaje reaparece con el ultimo NPC con el que interactuó en modo 'after'. 3. El personaje reaparece en el 'respawn'.
<b>FLUJO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
1. El jugador no fallece en modo 'after' y no encuentra donde falleció.	1. El personaje muere. 2. El personaje sigue jugando en modo 'after'.	2. El personaje reaparece con el ultimo NPC con el que interactuó en modo 'after'. 3. El personaje se mantiene en modo 'after'.
<b>POSTCONDICIÓN</b>	Revivir o Morir	

Cuadro 9: Descripción del caso de uso: Cambiar modo

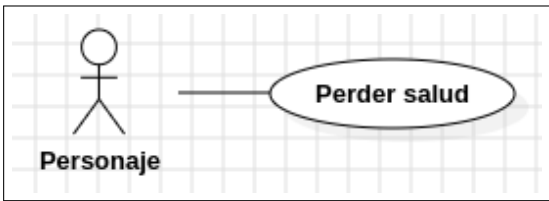


Figura 12: Caso de uso: Perder salud

CASO DE USO	Perder salud	
ACTOR	Personaje (Jugador)	
DESCRIPCIÓN	El jugador pierde puntos de vida al sufrir daño.	
PRECONDICIÓN	Ninguna.	
FLUJO BÁSICO	ACTOR	SISTEMA
	1. Choca con un obstáculo.	2. Se disminuye un punto de vida y el personaje regresa al inicio de la habitación.
FLUJO ALTERNO	ACTOR	SISTEMA
1. Vida mínima.	1. Choca con un obstáculo.	2. Fallece y cambia a modo 'after'.
POSTCONDICIÓN	Morir	

Cuadro 10: Descripción del caso de uso:

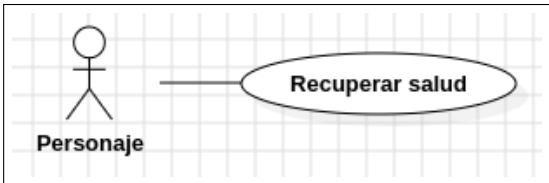


Figura 13: Caso de uso: Recuperar salud

<b>CASO DE USO</b>	Recuperar salud	
<b>ACTOR</b>	Personaje (Jugador)	
<b>DESCRIPCIÓN</b>	Aumenta los puntos de vida de personaje cada que termina una habitación.	
<b>PRECONDICIÓN</b>	Ninguna	
<b>FLUJO BÁSICO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. El jugador termina una habitación.	2. Adquiere un punto de vida.
<b>FLUJO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
1. Vida máxima.	1. El jugador termina una habitación.	2. No recibe puntos adicionales de vida.
<b>POSTCONDICIÓN</b>	Ninguna.	

Cuadro 11: Descripción del caso de uso: Recuperar salud

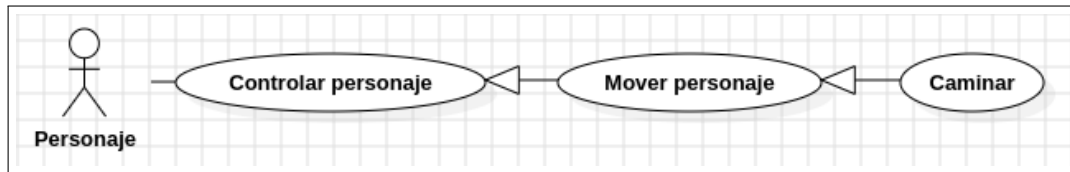


Figura 14: Caso de uso: Caminar

<b>CASO DE USO</b>	Caminar	
<b>ACTOR</b>	Personaje (Jugador)	
<b>DESCRIPCIÓN</b>	El jugador se mueve en el eje horizontal.	
<b>PRECONDICIÓN</b>	Ninguna.	
<b>FLUJO BÁSICO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. El jugador presiona las teclas de movimiento horizontal.	2. El personaje se mueve en la dirección seleccionada.
<b>FLUJO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
1. Colisión.	1. El jugador presiona las teclas de movimiento horizontal. 3. El personaje alcanza un muro.	2. El personaje se mueve en la dirección seleccionada. 4. El personaje colisiona y detiene el movimiento.
<b>POSTCONDICIÓN</b>	Ninguna	

Cuadro 12: Descripción del caso de uso: Caminar

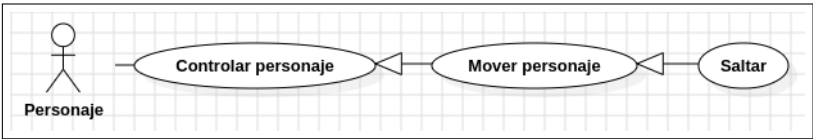


Figura 15: Caso de uso: Saltar

CASO DE USO	Saltar	
ACTOR	Personaje (Jugador)	
DESCRIPCIÓN	El personaje realiza un salto vertical mediante un impulso.	
PRECONDICIÓN	Ninguna.	
FLUJO BÁSICO	ACTOR	SISTEMA
	1. El personaje esta tocando el suelo y presiona la tecla para saltar.	2. Cambio de posición vertical mediante un impulso (curva parabólica).
FLUJO ALTERNO	ACTOR	SISTEMA
1. Colisión.	1. El personaje esta tocando el suelo y presiona la tecla para saltar. 3. El personaje alcanza un muro(techo).	2. Cambio de posición vertical mediante un impulso (curva parabólica). 4. Se cancela la velocidad vertical del personaje para empezar su descenso.
FLUJO ALTERNO	ACTOR	SISTEMA
2. Permision de salto.	1. El personaje se encuentra en el aire. 3. Toca una superficie de piso.	2. Se inhabilita el salto. 4. Se habilita el salto.
POSTCONDICIÓN	Ninguna	

Cuadro 13: Descripción del caso de uso: Saltar

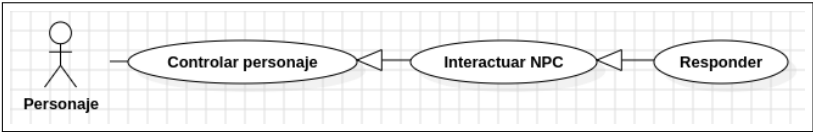


Figura 16: Caso de uso: Responder

<b>CASO DE USO</b>	Responder	
<b>ACTOR</b>	personaje (Jugador)	
<b>DESCRIPCIÓN</b>	El jugador responde el mensaje del NPC desde el teclado.	
<b>PRECONDICIÓN</b>	NPC: mostrar mensaje	
<b>FLUJO BÁSICO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
	1. El jugador escribe el mensaje y lo envía.	2. Se envía el mensaje al sistema inteligente e imprime una respuesta.
<b>FLUJO ALTERNO</b>	<b>ACTOR</b>	<b>SISTEMA</b>
1. Cancelación.	1. El jugador escribe el mensaje y cierra la ventana de dialogo.	2. Cierra la ventana de dialogo y regresa al juego.
<b>POSTCONDICIÓN</b>	Ninguna.	

Cuadro 14: Descripción del caso de uso: Responder

## 8.4. Diagrama de clases

Con base en el genero del videojuego, se requiere de diferentes clases que complementen la funcionalidad lógica. Elementos como la cámara, 'sprites', colisiones, sonidos, partículas, animaciones, cinemáticas, etc. pueden ser añadidas directo desde el IDE de Unity. Por esta razón, el diagrama de clases de la figura 17 omite esos aspectos de construcción (constructores, 'getters' y 'setters' incluidos) y busca representar la estructura y relación entre clases.

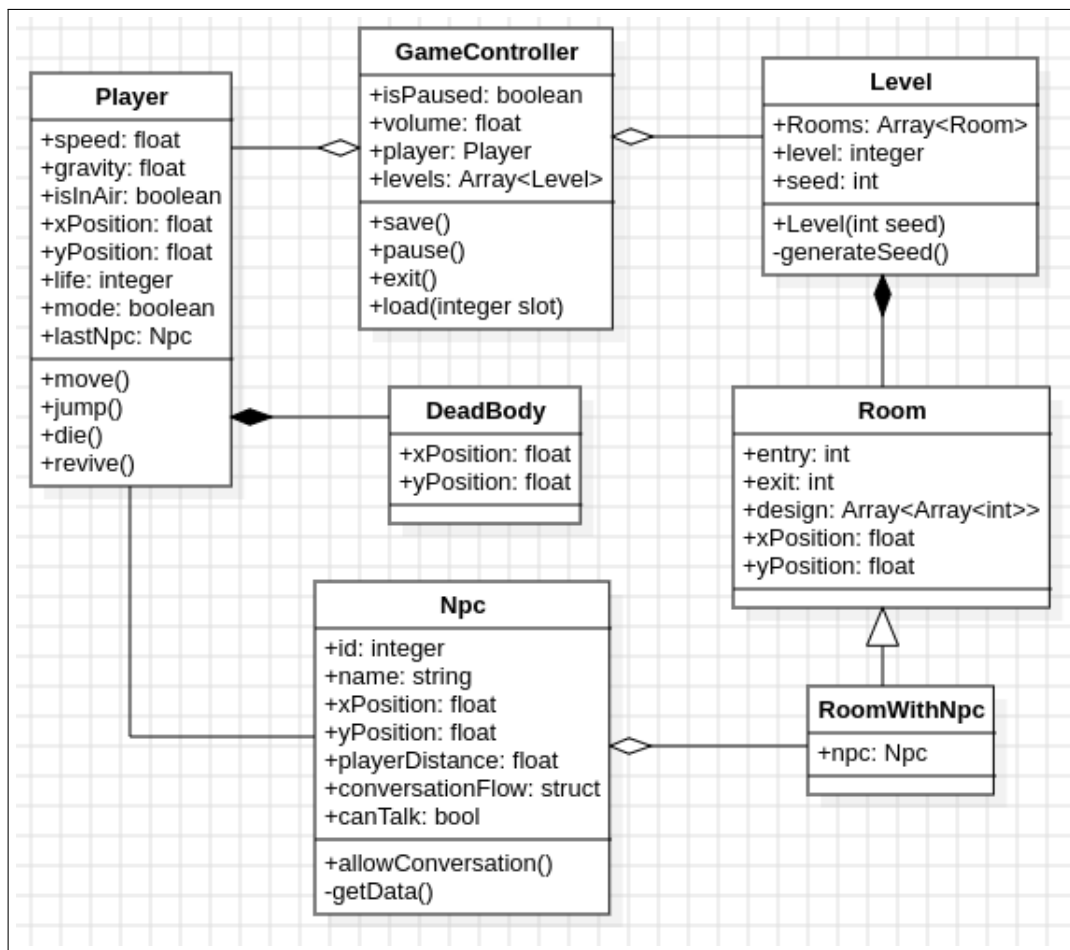


Figura 17: Diagrama de clases

## 9. Prototipo



## 10. Conclusiones

En esta primera etapa del proyecto, se logró establecer la estructura básica fundamental para el desarrollo del videojuego. Se alcanzaron varios hitos importantes que sientan las bases para las etapas subsiguientes:

- Se implementó con éxito el controlador del personaje, que incluye animaciones fluidas, efectos visuales y un sistema de movimiento que responde de manera intuitiva a las acciones del jugador.
- Se desarrolló un controlador funcional para los NPCs, permitiendo su generación dentro de los niveles y la interacción no procedural con el jugador. Esto establece las primeras interacciones y dinámicas básicas entre el personaje principal y los personajes no jugables.
- Se diseñó y probó un modelo inicial para la construcción de habitaciones, lo que facilita la creación y la implementación de diversos diseños de habitaciones que enriquecerán la experiencia del juego.
- Se implementó un generador de niveles que utiliza semillas o claves únicas para generar entornos y paisajes de manera procedural, asegurando así experiencias de juego únicas y variadas en cada partida.
- Se avanzó significativamente en la creación del entorno visual y artístico del juego, sentando las bases estéticas y de diseño que capturan la esencia y la atmósfera deseada para el juego final.

Para las próximas etapas del desarrollo, se planea enfocarse en la implementación de controladores adicionales para la configuración del juego, permitiendo ajustes dinámicos y mejoras en la jugabilidad. Además, se buscará expandir la variedad de habitaciones disponibles, diseñar nuevos niveles desafiantes, introducir personajes adicionales con comportamientos distintivos y mejorar continuamente los diseños gráficos para ofrecer una experiencia visualmente cautivadora y cohesiva.

Estas metas futuras no solo enriquecerán el contenido y la complejidad del juego, sino que también asegurarán que el proyecto avance hacia su objetivo final de crear un videojuego *rogue lite* innovador y atractivo para los jugadores.

## 11. Referencias Bibliográficas

### Referencias

- [1] Supergiant Games, *Hades*. Supergiant Games, 2019.
- [2] Maddy Makes Games, *Celeste*. Maddy Makes Games, 2018.
- [3] Team Cherry, *Hollow Knight*. Team Cherry, 2017.
- [4] Toge Productions, *Coffee Talk*. Toge Productions, 2020.
- [5] Character Technologies, Inc. “character.ai”, 2022. [En línea]. Disponible: <https://character.ai>. [Accedido: 11-Junio-2024].
- [6] Secretaria de salud, “Depresión y ansiedad, principales problemas de salud mental en estos segmentos poblacionales”, Gobierno de México. [En línea]. Disponible: <https://www.gob.mx/salud/prensas>. [Accedido: 23-Junio-2024].
- [7] Gonzáles Mario, “Cómo se aplica la Inteligencia Artificial en los videojuegos”, Instituto de ingeniería del conocimiento. [En línea]. Disponible: <https://www.iic.uam.es/noticias/como-aplica-inteligencia-artificial-en-videojuegos/>. [Accedido: 23-junio-2024].
- [8] Hernández Cabezas Ariadna, “Videojuegos como tratamiento terapéutico”, [En línea]. Disponible: [https://repositori.uji.es/xmlui/bitstream/handle/10234/191101/TFM\\_2020\\_HernandezCabezas\\_Ariadna.pdf?sequence=1&isAllowed=y](https://repositori.uji.es/xmlui/bitstream/handle/10234/191101/TFM_2020_HernandezCabezas_Ariadna.pdf?sequence=1&isAllowed=y) [Accedido: 23-junio-2024].
- [9] Salinaz Benito, “UDLAP”, catarina. [En línea]. Disponible: [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lnd/benitez\\_salinas\\_ve/capitulo1.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lnd/benitez_salinas_ve/capitulo1.pdf). [Accedido: 20-Junio-2024].
- [10] Plarium, “Roguelike”. [En línea]. Disponible: <https://plarium.com/es/glossary/roguelike-games/>. [Accedido: 20-Junio-2024].
- [11] Google Cloud, “¿Qué es el procesamiento del lenguaje natural?”. [En línea]. Disponible: <https://cloud.google.com/learn/what-is-natural-language-processing>. [Accedido: 20-Junio-2024].
- [12] Real Academia Española, “polisémico, ca”. [En línea]. Disponible: <https://www.rae.es/diccionario-estudiante/polismico>. [Accedido: 20-Junio-2024].

- [13] Real Academia Española, “homónimo, ma”. [En línea]. Disponible: <https://dle.rae.es/homnimo>. [Accedido: 20-Junio-2024].
- [14] IMB, “¿Qué es el procesamiento del lenguaje natural (NLP)?”. [En línea]. Disponible: <https://www.ibm.com/mx-es/topics/natural-language-processing>. [Accedido: 20-Junio-2024].
- [15] Prompts para IA, “Procesamiento del lenguaje natural y detección de emociones”, 2024. [En línea]. Disponible: <https://prompt.uno/vision-por-computadora/procesamiento-del-lenguaje-natural-y-deteccion-de-emociones/>. [Accedido: 20-Junio-2024].
- [16] Universidad Europea de Madrid, S.L.U, “Análisis léxico”, Cartagena99. [En línea]. Disponible: [https://www.cartagena99.com/recursos/alumnos/apuntes/ININF2\\_M4\\_U2\\_T1.pdf](https://www.cartagena99.com/recursos/alumnos/apuntes/ININF2_M4_U2_T1.pdf). [Accedido: 20-Junio-2024].
- [17] Oracle, “¿Qué es el machine learning?”. [En línea]. Disponible: <https://www.oracle.com/mx/artificial-intelligence/machine-learning/what-is-machine-learning/>. [Accedido: 20-Junio-2024].
- [18] Organización Mundial de la Salud, “Salud mental: fortalecer nuestra respuesta”, 17-Junio-2022. [En línea]. Disponible: <https://www.who.int/es/news-room/fact-sheets/detail/mental-health-strengthening-our-response>. [Accedido: 20-Junio-2024].
- [19] Unicef, “Cómo reconocer nuestras emociones”. [En línea]. Disponible: <https://www.unicef.org/lac/crianza/seguridad-proteccion/como-reconocer-nuestras-emociones>. [Accedido: 20-Junio-2024].
- [20] Arancibia Psicología, “La relación entre las emociones y la salud mental”. [En línea]. Disponible: <https://www.arancibia-psicologia.com>. [Accedido: 20-Junio-2024].
- [21] Clinica Universidad de Navarra, “Terapia”. [En línea]. Disponible: <https://www.cun.es/diccionario-medico/terminos/terapia>. [Accedido: 22-Junio-2024].
- [22] Definición, “Psicología”. [En línea]. Disponible: <https://definicion.de/psicologia/>. [Accedido: 22-Junio-2024].
- [23] Mayo Clinic, “Psicoterapia”. [En línea]. Disponible: <https://www.mayoclinic.org/es/tests-procedures/psychotherapy/about/pac-20384616>. [Accedido: 22-Junio-2024].

- [24] Unity Technologies. “Unity para principiantes — Unity”. Unity. [En línea]. Disponible: <https://unity.com/es/learn/get-started>. [Accedido: 13-Junio-2024].
- [25] E. Canle Fernández, “¿Qué lenguaje de programación usa Unity? – Tokio School”. Tokio School. [En línea]. Disponible: <https://www.tokioschool.com/noticias/lenguaje-unity>. [Accedido: 13-Junio-2024].
- [26] UNAM, “PROLOG”. [En línea]. Disponible: <https://virtual.cuautitlan.unam.mx/intar>. [Accedido: 13-Junio-2024].
- [27] MongoDB, “MongoDB: The Developer Data Platform”. MongoDB. [En línea]. Disponible: <https://www.mongodb.com>. [Accedido: 13-Junio-2024].
- [28] MKLab, “StarUML”. StarUML. [En línea]. Disponible: <https://www.pixilart.com>. [Accedido: 13-Junio-2024].
- [29] Pixilart, “Pixilart”. Pixilart LLC. [En línea]. Disponible: <https://staruml.io>. [Accedido: 13-Junio-2024].