

**I.E.S Fuengirola N°1**

**DESARROLLO DE APLICACIONES MULTIPLATAFORMA**

**PROYECTO INTEGRADO**

# **Halaman Tower**

**Departamento de Informática**

**Manual Técnico**

**Autor:**

**Juan Carlos Lucena Martinez**

## ÍNDICE DE CONTENIDOS

<b>1 - Sobre éste proyecto</b>	<b>2</b>
1.1 - Control de versiones	2
1.2 - Licencia de uso	2
<b>2 - Análisis del problema</b>	<b>2</b>
2.1 - Introducción al problema	2
2.2 - Antecedentes	2
2.3 - Objetivos	2
2.4 - Requisitos	2
2.4.1 - Funcionales	2
2.4.2 - No funcionales	2
2.5 - Recursos	2
2.5.1 - Software	2
2.5.2 - Hardware	2
<b>3 - Diseño de la solución software</b>	<b>2</b>
3.1 - Modelados	2
3.1.1 - Casos de uso	2
3.1.2 - [ Interacción ]	2
3.1.3 - [ Estado ]	2
3.1.4 - [ Actividad ]	2
3.2 - Prototipado gráfico	2
3.2.1 - [ Escritorio ]	2
3.2.2 - [ Tablets / Smartphones ]	2
3.3 - Base de datos	2
3.3.1 - Diseño Conceptual (ER)	2
3.3.2 - Diseño lógico (tablas normalizadas)	2
<b>4 - Implementación</b>	<b>2</b>
4.1 - Codificación	2
4.1.1 - [ Usabilidad ]	2
4.1.2 - Backend	2
4.1.3 - Frontend	2
4.2 - [ Pruebas ]	2
<b>5 - Documentación</b>	<b>2</b>
5.1 - Empaquetado / Distribución	2
5.2 - Instalación	2
5.3 - Manual de Usuario / Referencia	2
<b>6 - Conclusiones</b>	<b>2</b>
<b>7 - Bibliografía</b>	<b>2</b>

## 1 - Sobre éste proyecto

## 1.1 - Control de versiones

El control de versiones utilizado será git junto a github

## 1.2 - Licencia de uso

Tendría una licencia de uso Creative Commons (CC), del tipo CC BY-SA-NC, esto implica que otros usuarios puedan usar, modificar y distribuir el trabajo, siempre que den crédito, las obras derivadas han de compartir la misma licencia y finalmente el impedimento de compartir y modificar el trabajo con fines comerciales

# 2 - Análisis del problema

## 2.1 - Introducción al problema

En la actualidad, muchos juegos carecen de una experiencia compacta que combine simplicidad, desafío estratégico y rejugabilidad. Halaman Tower surge para atender esta necesidad, ofreciendo un RPG por turnos con mecánicas de rogue-like, donde el jugador se enfrenta a una serie ininterrumpida de enemigos, eligiendo cómo mejorar sus estadísticas y buscando maximizar su puntuación. Este videojuego busca brindar una experiencia rápida, estratégica y competitiva, ideal para quienes buscan partidas dinámicas pero con profundidad táctica. Además, incorpora un sistema de registro en línea que fomenta la competitividad al permitir a los jugadores comparar sus resultados, integrando de forma innovadora componentes locales y en línea en un formato accesible y rejugable.

## 2.2 – Antecedentes

En el género de los videojuegos RPG y rogue-like, han existido numerosos títulos que han explorado mecánicas de combate por turnos y progresión aleatoria. Juegos como Slay the Spire, Darkest Dungeon y Hades han demostrado el atractivo de estos sistemas, combinando estrategia y rejugabilidad con un diseño desafiante. Sin embargo, muchos de estos juegos requieren sesiones de juego extensas o presentan una curva de aprendizaje elevada, lo que puede desalentar a ciertos jugadores. Halaman Tower busca ofrecer una alternativa accesible y dinámica, manteniendo la profundidad táctica y la aleatoriedad característica del género, pero con partidas más rápidas y un sistema de progresión sencillo pero desafiante.

## 2.3 – Objetivos

Un juego capaz de entretener al público, ofreciendo una posibilidad al estilo de juego de cada persona, con capacidad de aprendizaje adaptándose a pequeñas mecánicas que hará mejor o peor al personaje en algunos aspectos, anteponiendo siempre la diversión del jugador

## 2.4 - Requisitos

### 2.4.1 – Funcionales

- Creación de cuenta y uso de la misma.

- Implementación de un combate por turnos.
- Generación de enemigos de forma aleatoria (entre los existentes en el juego).
- Permitir la mejora de los atributos seleccionados del personaje.
- Subida de nivel.
- Ser capaz de registrar puntuaciones y mostrarse, así como las mejores puntuaciones.

### 2.4.2 - No funcionales

- Creación de una interfaz sencilla
- Evitar problemas de rendimiento, buscando que funcione sin necesidad de dispositivos potentes.
- Permitir que el juego pueda escalar para que sea actualizable en el futuro.
- Conexión con la base de dato segura.

## 2.5 - Recursos

### 2.5.1 – Software

- Visual Studio Code: Editor de código para escribir scripts en C# y editar configuraciones.
- Unity Editor: Entorno de desarrollo integrado (IDE) para crear el juego.

#### **Software online:**

- PythonAnywhere: Servicio web utilizado para alojar el servidor y la base de datos.
- Piskel: Herramienta en línea para la creación de sprites y pixel art.
- MyEdit: Plataforma web para edición de imágenes y audio.
- Suno: Servicio web utilizado para generar música.

### 2.5.2 - Hardware

Se estima que un ordenador con 8GB de RAM Cerca de 2GB de espacio en disco. Resolución de pantalla 1280x720 o mismas proporciones. Intel core i5 o superior. Windows 10. Las estimaciones pueden variar y seguramente serán menores una vez el juego este completo en su versin 1.0

## 3 - Diseño de la solución software

### 3.1 - Modelados

#### 3.1.1 - Casos de uso

Actor: Jugador.

- Iniciar partida.
- Seleccionar habilidad.
- Realizar acción.
- Avazar stage.
- Guardar puntuación.

### 3.1.2 - [ Interacción ]

- Se elige una habilidad.
- El juego determina el tipo de habilidad.
- Interactúa con el jugador o enemigo según la habilidad.
- Se actualiza la vida modificada.

### 3.1.3 - [ Estado ]

Vivo: el personaje / enemigo tiene vida y puede realizar acciones.  
Muerto: el personaje/ enemigo ha reducido su vida a 0.

### 3.1.4 - [ Actividad ]

1. Inicia el turno del jugador.
2. El jugador usa una habilidad.
3. Se ejecuta la acción y los cambios a la salud que impliquen.
4. Inicia el turno del enemigo.
- 5 Se repite el ciclo hasta que uno caiga

## 3.2 - Prototipado gráfico

### 3.2.1 - [ Escritorio ]



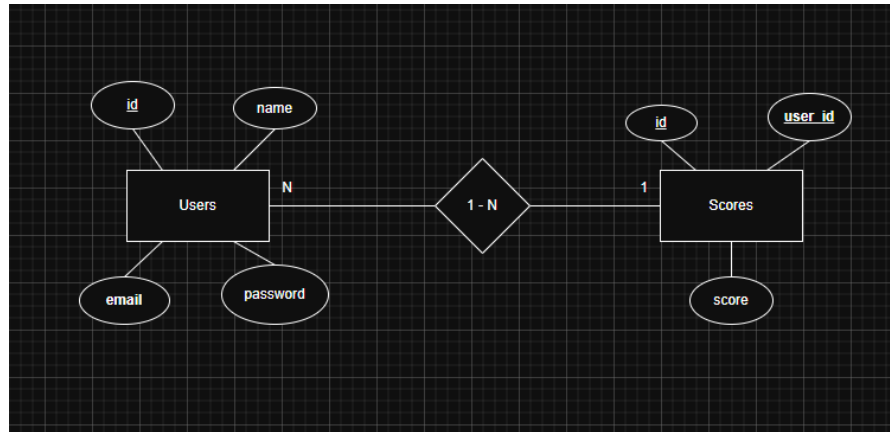
Personaje controlado por el jugador y enemigo controlado por la cpu, un log que indica lo sucedido y que a su vez contiene las habilidades que puede usar el jugador

### 3.2.2 - [ Tablets / Smartphones ]

No aplica

## 3.3 - Base de datos

### 3.3.1 - Diseño Conceptual (ER)



### 3.3.2 - Diseño lógico (tablas normalizadas)

Users ( id INT PRIMARY KEY AUTO\_INCREMENT, name VARCHAR(50) NOT NULL, email VARCHAR(100) UNIQUE NOT NULL, password VARCHAR(255) NOT NULL )

Scores ( id INT PRIMARY KEY AUTO\_INCREMENT, user\_id INT NOT NULL, score INT NOT NULL, FOREIGN KEY (user\_id) REFERENCES Users(id) ON DELETE CASCADE )

## 4 - Implementación

### 4.1 - Codificación

#### 4.1.1 - [ Usabilidad ]

Sencilla e intuitiva de usar, permitiendo su uso a cualquier persona

#### 4.1.2 – Backend

Gestión de los turnos de los personajes, el uso de las habilidades seleccionadas en cada turno y el desenlace de la misma.

Guardado del score en los servidores de Python Anywhere mediante una API REST, además de poder consultarse

#### 4.1.3 – Frontend

La vida y niveles de los personajes, el log de sucesos y durante el turno del jugador, las habilidades que este puede usar.

### 4.2 - [ Pruebas ]

Realizadas en el motor de unity hub para verificar funcionamiento completo y postman

para llamadas a la API REST en etapas tempranas

## 5 - Documentación

### 5.1 - Empaquetado / Distribución

Será usado unity build settings, lo que permitirá crearse en distintos formatos, entre estos incluye para ordenador (ejecutable windows, mac y linux), dispositivo móvil y WebGL y más, actualmente solo será creada la versión de windows, queriendo una versión WebGL a futuro para plataformas de distribución de juegos indie como "itch.io".

### 5.2 – Instalación

No es requerida, debido al formato de unity y al tamaño del juego, ejecutar el programa sería suficiente

### 5.3 - Manual de Usuario / Referencia

Ejecute el archivo HalamanTower.exe

Solo se requiere de ratón.

Seleccione la habilidad que desee, tras ejecutarse su acción, el enemigo realizará la suya.

Tras eso, se repetirá el ciclo hasta que uno venza, en caso de ganar el usuario, su personaje se fortalecerá y avanzará a la siguiente dificultad, el objetivo es llegar lo mas lejos posible sin perecer.

## 6 – Conclusiones

Halaman Tower ha sido diseñado con la intención de ofrecer una experiencia de juego entretenida, desafiante y accesible. La combinación de mecánicas de RPG y rogue-like en un formato compacto permite que los jugadores disfruten de partidas rápidas pero estratégicas, lo que mejora la rejugabilidad.

Posee buenas capacidades de escalabilidad para un crecimiento futuro.

## 7 – Bibliografía

Documentación oficial de Unity:

Unity Technologies. (2024). Unity Manual. <https://docs.unity3d.com/Manual/>.

Canal de Youtube utilizado como apoyo en la creación:

<https://www.youtube.com/@DanielWayota>

Documentación oficial Python Anywhere:

<https://help.pythonanywhere.com/pages/>