

Zero Hour: Undead

Proyecto Integrado – Desarrollo de Aplicaciones Multiplataforma

Gonzalo Castillo Muñoz

Centro educativo: Escuela Profesional Vedruna Sevilla

Curso académico: 2024–2025

Descripción rápida:

Zero Hour: Undead es un juego de supervivencia en primera persona inspirado en el modo Zombies de Call of Duty, desarrollado como parte del proyecto final del ciclo superior de DAM. Este documento recopila la documentación técnica, funcional y visual del proyecto.

Índice de la Documentación

1. Portada / Landing page
2. Índice
3. Diagrama de Casos de Uso
4. Diagrama de Clases
5. Diagrama Entidad-Relación
6. Diagrama de Componentes
7. Casos de Prueba

Diagrama de Casos de Uso

A continuación, se presenta el diagrama de casos de uso que representa las distintas interacciones que el **jugador** puede realizar dentro del videojuego **Zero Hour: Undead**.

Actor principal: Jugador

Sistema: Zero Hour: Undead (el videojuego)

Funcionalidades representadas

- Registrarse en el sistema
- Iniciar sesión
- Iniciar una partida
- Recargar armas
- Comprar armas
- Mejorar armas mediante la máquina Pack-a-Punch
- Abrir puertas para desbloquear nuevas zonas

- Recoger power-ups
- Visualizar el marcador de puntuación
- Morir (cuando el jugador pierde)

Diagrama:

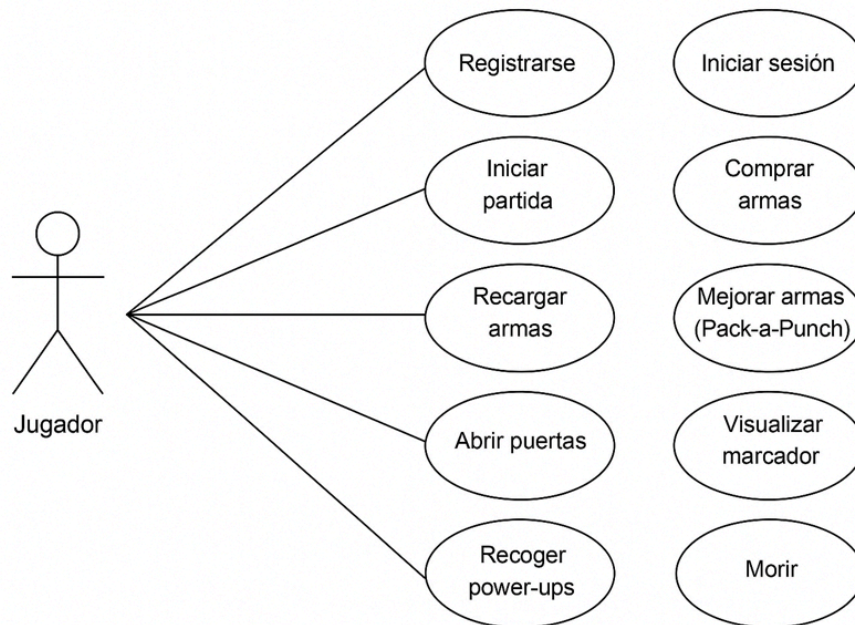


Diagrama de clases

El siguiente diagrama de clases representa la estructura principal del videojuego.

Este diagrama resume las **clases clave**, sus **atributos y métodos más relevantes**, y las **relaciones entre ellas**, organizadas por módulos funcionales.

🎯 Objetivo del diagrama

Reflejar la **arquitectura del sistema** para entender cómo están organizadas las funcionalidades principales del juego, desde la interacción del jugador, el combate, la IA enemiga, el sistema de oleadas, las armas, el entorno y la conexión con la API externa.

Módulos del sistema

Módulo	Clases principales	Funcionalidad principal
Jugador	<code>PlayerMovement</code> , <code>CameraLook</code> , <code>PlayerInteractions</code>	Movimiento, control de cámara y gestión de colisiones/daño
Zombis	<code>AI</code> , <code>ZombieAnimationController</code>	Inteligencia artificial, seguimiento, animaciones y muerte de enemigos

Armas	WeaponLogic , WeaponSwitch , WeaponManager , WeaponBuyZone	Disparo, cambio de arma, compra y gestión del inventario armado
UI	Menu , MenuNavegacion , GameManager (UI)	Menús, panel de pausa, mensajes interactivos, puntuación y final de partida
Mundo	DoorZone	Desbloqueo de zonas y activación de nuevos puntos de spawn
Gestor global	GameManager	Control de salud, rondas, spawn de zombis, puntuación, eventos especiales (insta-kill)
API	AuthManager , EnviarSesion	Registro/login del jugador y guardado de la partida al morir mediante una API externa

El diagrama UML incluye:

- Clases con sus atributos y métodos clave.
- Relaciones entre clases (uso, herencia, composición).
- Organización visual por colores, agrupando las clases por módulo funcional.

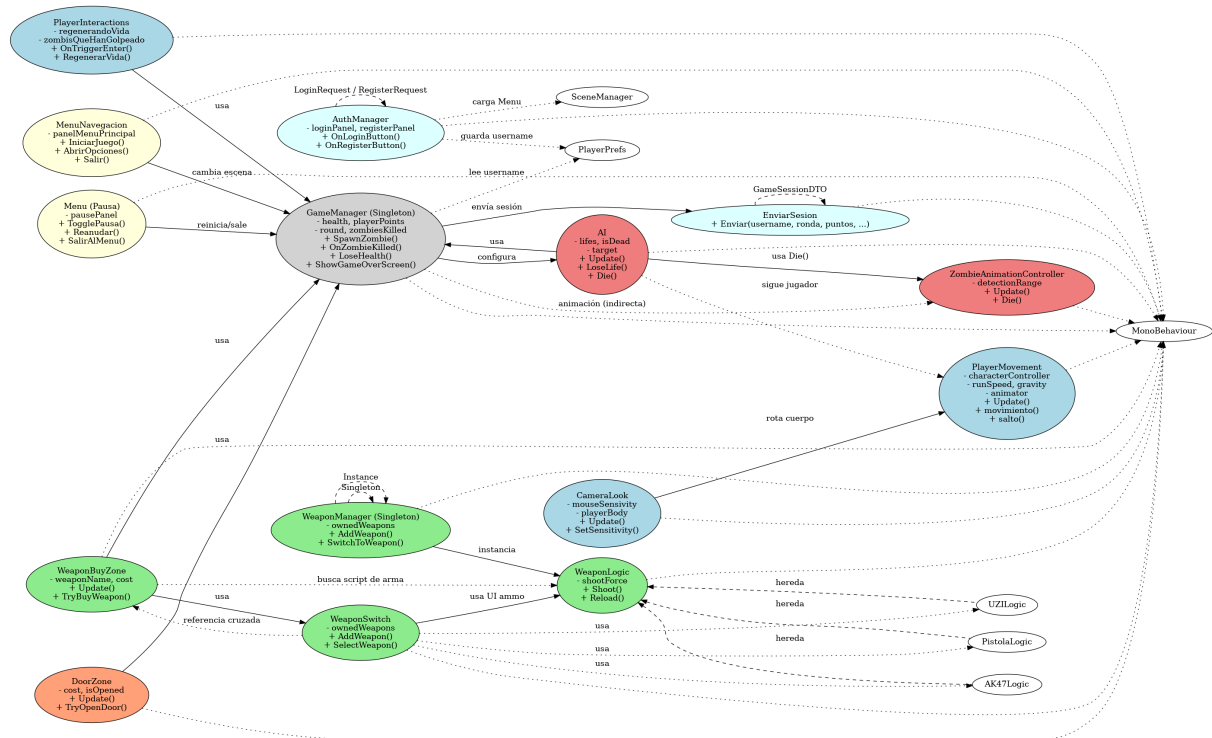


Diagrama Entidad-Relación

Este diagrama representa las entidades principales del sistema y sus relaciones, tal y como están estructuradas en la base de datos MySQL utilizada por el backend del proyecto. Su objetivo es ilustrar cómo se organizan y relacionan los datos que se generan durante el uso del videojuego.

◆ Descripción de las entidades:

- **users**

- user_id: INT (PK)
- username: VARCHAR(255)
- password: VARCHAR(255)
- email: VARCHAR(255)
- fecha_registro: DATETIME

- **game_sessions**

- id: BIGINT (PK)
- ronda: INT
- puntos: INT
- mapa: VARCHAR(255)
- duracion: VARCHAR(255)
- fecha: DATETIME
- user_id: INT
- username: VARCHAR(255)

- **armas**

- id: INT (PK)
- nombre: VARCHAR(100)
- daño: INT

- **enemigos**

- id: INT (PK)
- nombre: VARCHAR(100)
- vida: INT
- incremento_vida: INT

Relaciones entre entidades:

- **users – game_sessions**

Relación 1:N.

Un usuario puede tener múltiples partidas registradas en el sistema.

- **users – armas**

Relación 1:2.

Un jugador puede llevar hasta un máximo de dos armas al mismo tiempo.

- **game_sessions – enemigos**

Relación 1:N.

Cada sesión de juego puede involucrar múltiples enemigos, que escalan en dificultad conforme avanzan las rondas.

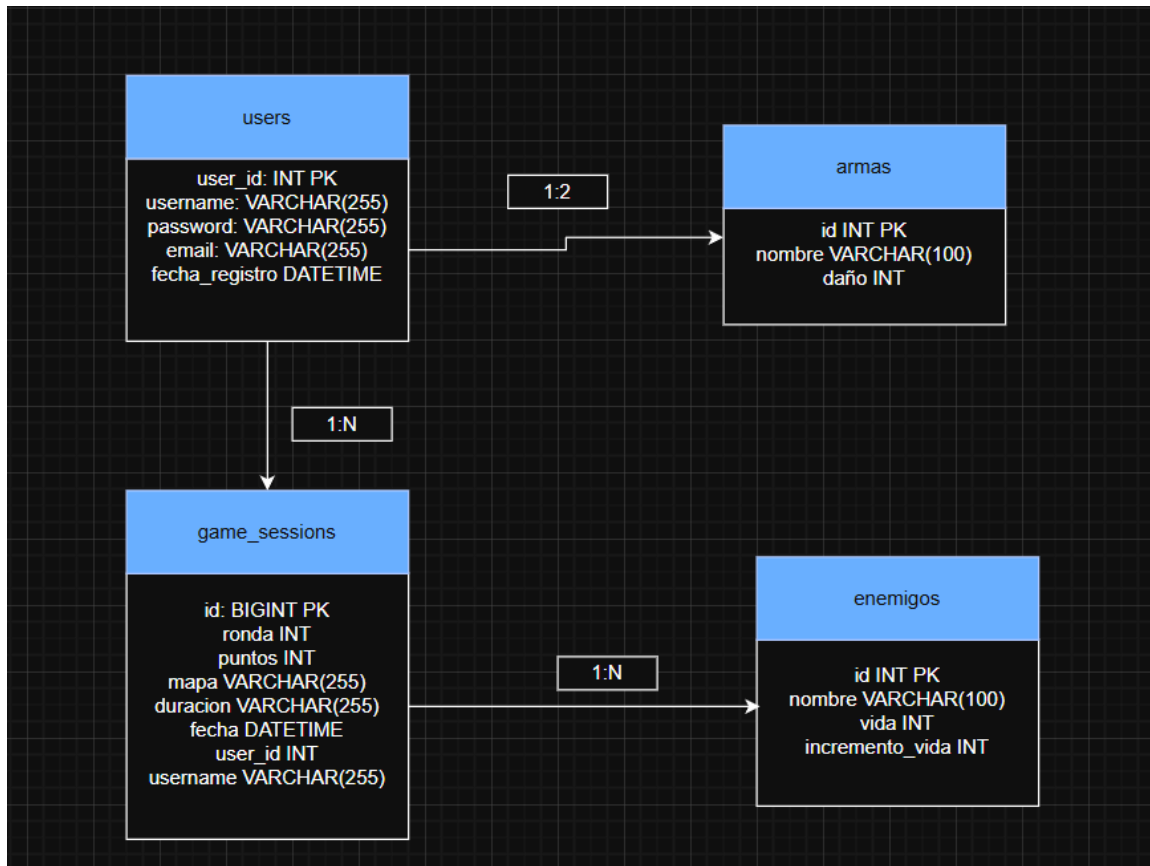


Diagrama de Componentes

El siguiente diagrama muestra la arquitectura modular del proyecto, dividiendo sus partes principales y cómo se comunican entre sí.

El proyecto está dividido principalmente en:

Frontend (Unity)

- **Sistema de UI**: Maneja menús, login, registro, HUD de partida, etc.
- **Controlador de Juego**: Gestiona la lógica general como oleadas, puntos y eventos.
- **Jugador**: Controla los scripts de movimiento, interacción y combate.
- **IA Zombis**: Componente autónomo de los enemigos, con lógica de seguimiento y ataque.
- **Manejador de Sonido**: Administra música de fondo y efectos de sonido.

Comunicación (API REST)

- **Sistema de Autenticación**: Permite registrar e iniciar sesión con credenciales.
- **Sistema de Partidas**: Permite registrar puntuaciones, mapas y duración de partidas.

Backend (API Java Spring Boot)

- **Controladores REST:** Exponen endpoints para registro, login y envío de partidas.
- **Servicios:** Contienen la lógica de negocio (validaciones, gestión de usuarios y partidas).
- **Modelo:** Entidades `User` , `GameSession` , etc.
- **Base de Datos:** MySQL con tablas `users` , `game_sessions` , `armas` , `enemigos` .

Diagrama

Asegúrate de subir la imagen a Notion o enlazarla correctamente.

Casos de prueba

ID	Funcionalidad	Descripción	Datos de Entrada	Resultado Esperado	Resultado Obtenido
TC-01	Registro de usuario	El jugador se registra con datos válidos	Nombre, Email, Contraseña	Usuario registrado correctamente	Usuario registrado correctamente
TC-02	Registro con campos vacíos	El jugador intenta registrarse sin rellenar todos los campos	"" , "" , ""	Error: campos requeridos	Error al registrarse
TC-03	Login correcto	Inicia sesión con usuario y contraseña correctos	Usuario registrado	Acceso al menú principal	Acceso al menú principal
TC-04	Login incorrecto	Usuario/contraseña incorrectos	Usuario válido, contraseña inválida	Error de autenticación	Error al iniciar sesión
TC-05	Iniciar partida	Click en botón de jugar	Click desde menú	Escena de juego cargada	Escena de juego cargada
TC-06	Disparo con munición	El jugador pulsa disparar con munición	Click izquierdo + munición disponible	Se dispara y se escucha el sonido	Se dispara y se escucha el sonido
TC-07	Disparo sin munición	Disparo con munición agotada	Click izquierdo + 0 balas	No se dispara	No se dispara
TC-08	Compra de arma correcta	Comprar arma teniendo puntos suficientes	1000 puntos, arma cuesta 750	Arma equipada, puntos restados	Arma equipada, puntos restados
TC-09	Compra sin puntos	Comprar arma sin puntos suficientes	200 puntos, arma cuesta 750	Compra rechazada	Compra rechazada
TC-10	Abrir puerta	Desbloquear zona con puntos	1000 puntos, puerta cuesta	Puerta se abre, puntos	Puerta se abre, puntos

		suficientes	500	restados	restados
TC-11	Abrir puerta sin puntos	Intentar abrir sin puntos	0 puntos	No se abre la puerta	No se abre la puerta
TC-12	Pack-a-Punch con puntos	Mejorar arma con puntos suficientes	Arma equipada, 5000 puntos	Arma mejorada con nuevas estadísticas	Arma mejorada con nuevas estadísticas
TC-13	Pack-a-Punch sin puntos	Intentar mejorar sin puntos suficientes	Arma equipada, 1000 puntos	Mejora rechazada	Mejora rechazada
TC-14	Guardar puntuación al morir	Se guarda puntuación tras perder	Ronda 12, puntos 2000	Guardado exitoso en la API	Guardado exitoso en la API
TC-15	API caída al guardar	El servidor no está disponible al intentar guardar	Ronda 7, 1500 puntos	Error 500 mostrado en consola	Error 500 mostrado en consola
TC-16	Easter Egg activado	El jugador activa el easter egg del nivel	Secuencia específica	Activación del evento oculto	Activación del evento oculto
TC-17	Easter Egg incompleto	Intenta activarlo sin cumplir condiciones	Secuencia incompleta	Nada ocurre	Nada ocurre
TC-18	Movimiento básico	Moverse usando teclas WASD	Presionar W, A, S, D	Jugador se desplaza	Jugador se desplaza
TC-19	Inputs bloqueados	Se intenta mover o disparar mientras el juego está pausado	Input durante pausa	Movimiento y disparo bloqueados	Movimiento y disparo bloqueados