



# Tecnológico de Monterrey

## **Ejercicio de Modelación de Base de Datos del reto**

Juan Pablo Narchi A01781518

Efrén Chávez A01769533

Paolo Zesati A01784391

5 de mayo del 2025

Construcción de Software y Toma de Decisiones

## **Justificación del Esquema de Base de Datos para Juego de Plataformas Roguelike**

Este documento presenta la justificación del esquema entidad-relación propuesto para el juego de plataformas tipo roguelike descrito. El esquema ha sido diseñado para almacenar toda la información necesaria para el funcionamiento del juego, manteniendo la integridad de los datos y evitando la duplicación de información.

### **Entidades y su Justificación**

#### **1. Jugador**

Esta entidad almacena la información relacionada con cada usuario que juega al juego.

- **Justificación:** Es necesario mantener registro de los jugadores para guardar su progreso, experiencia acumulada y las mejoras permanentes que han desbloqueado. Esto permite implementar el sistema de progresión entre partidas que es característico de los juegos roguelike.
- **Uso en el juego:** Gestionar el sistema de experiencia y mejoras permanentes (vidas adicionales y reducción del tiempo de enfriamiento de la bola de fuego).

#### **2. Partida**

Representa cada sesión de juego iniciada por un jugador.

- **Justificación:** Es necesario registrar cada intento del jugador para calcular la experiencia ganada y actualizar las estadísticas del jugador. También permite implementar posibles funcionalidades futuras como tablas de clasificación.
- **Uso en el juego:** Seguimiento del progreso del jugador en cada intento, cálculo de experiencia ganada y registro del nivel máximo alcanzado.

#### **3. Nivel**

Contiene la información sobre cada uno de los tres niveles del juego más la sala del jefe final.

- **Justificación:** Cada nivel tiene características únicas como tema, dificultad, enemigos y obstáculos específicos que deben ser almacenados de forma estructurada.
- **Uso en el juego:** Configurar los distintos escenarios que el jugador deberá superar, estableciendo la dificultad progresiva y las mecánicas específicas de cada nivel.

#### 4. Enemigo

Almacena la información de los diferentes tipos de enemigos presentes en el juego.

- **Justificación:** Cada tipo de enemigo (Lobo, Mago, Jefe) tiene comportamientos, estadísticas y mecánicas distintas que deben ser definidas individualmente.
- **Uso en el juego:** Implementar la IA y comportamientos de los enemigos en cada nivel, estableciendo sus patrones de movimiento y ataques.

#### 5. Obstáculo

Representa los obstáculos como barriles rodantes y hechizos que el jugador debe evitar.

- **Justificación:** Los obstáculos son elementos fundamentales en la jugabilidad y varían en tipo y comportamiento según el nivel.
- **Uso en el juego:** Configurar la generación de barriles, su velocidad y comportamiento en cada nivel, así como los hechizos que lanzan los magos.

#### 6. MejoraPoder

Contiene información sobre las mejoras disponibles, tanto temporales como permanentes.

- **Justificación:** El sistema de mejoras es una parte central del diseño del juego, permitiendo la progresión del jugador tanto dentro de una partida como entre partidas.
- **Uso en el juego:** Gestionar las mejoras que el jugador puede adquirir con experiencia (permanentes) o encontrar durante los niveles (temporales).

#### Relaciones y su Justificación

### **1. Jugador-Partida (1:N)**

- **Justificación:** Un jugador puede iniciar múltiples partidas a lo largo del tiempo, pero cada partida pertenece a un único jugador.
- **Uso en el juego:** Relacionar cada intento de juego con el jugador correspondiente para actualizar su experiencia total.

### **2. Partida-Nivel (N:M, mediante tabla Partida\_Nivel)**

- **Justificación:** Una partida atraviesa múltiples niveles, y cada nivel puede ser parte de múltiples partidas diferentes. Esta relación muchos a muchos permite registrar el progreso del jugador a través de los niveles en cada partida.
- **Uso en el juego:** Seguimiento del progreso del jugador en cada nivel dentro de una partida específica.

### **3. Nivel-Enemigo (1:N)**

- **Justificación:** Un nivel contiene múltiples enemigos, pero cada enemigo pertenece a un nivel específico. Esto permite configurar la distribución de enemigos por nivel.
- **Uso en el juego:** Determinar qué enemigos aparecen en cada nivel y configurar su comportamiento según el contexto del nivel.

### **4. Nivel-Obstáculo (1:N)**

- **Justificación:** Un nivel contiene múltiples obstáculos, pero cada obstáculo pertenece a un nivel específico. Esto permite configurar la distribución y comportamiento de obstáculos por nivel.
- **Uso en el juego:** Determinar qué obstáculos aparecen en cada nivel y configurar su comportamiento y dificultad.

### **5. Jugador-MejoraPoder (N:M, mediante tabla Jugador\_MejoraPoder)**

- **Justificación:** Un jugador puede tener múltiples mejoras desbloqueadas, y cada mejora puede ser desbloqueada por múltiples jugadores. Esta relación permite implementar el sistema de progresión permanente.
- **Uso en el juego:** Seguimiento de las mejoras permanentes que cada jugador ha desbloqueado usando su experiencia acumulada.

## Modelo Entidad-Relación

