



Ejercicio de Modelación de Base de Datos del reto

Integrantes:

Luis Emilio Veledíaz Flores

Valentina Castilla Melgoza

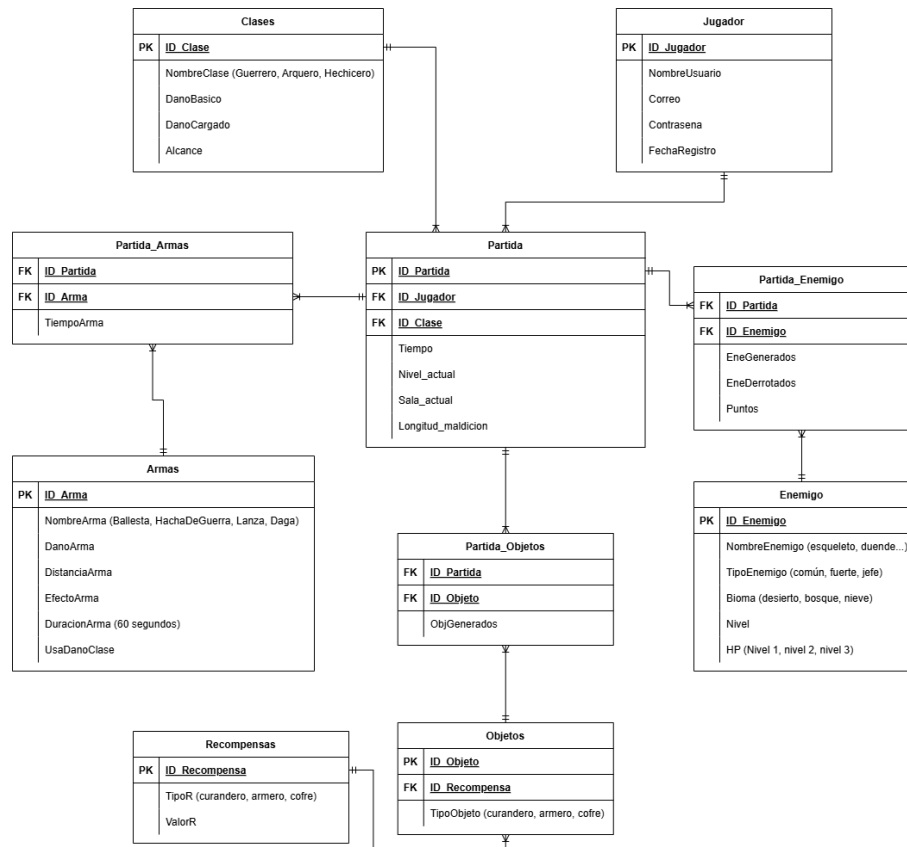
Diego de la Vega Saishio

Campus Santa Fe

Construcción de software y toma de decisiones (Gpo 401)

Esteban Castillo Juarez

Miércoles 2 de Abril del 2025



Link al modelo:

<https://drive.google.com/file/d/1Wi7SrRLY3zQUaV8rBgwyInCcUam2pNxM/view?usp=sharing>

Justificación del diagrama Entidad-Relación

- 1) Jugador: La tabla jugador registra a los usuarios del juego, junto con su información de acceso y registro.
 - a) Primary Key: ID_Jugador INT PRIMARY KEY AUTO_INCREMENT
 - b) Foreign Key:
 - i) Relación 1 a N con partida: La justificación de esta relación es que un jugador puede tener muchas partidas completadas o activas en diferentes pestañas.
 - c) Atributos:
 - i) NombreUsuario: Solo debe contener letras, números, guiones bajos o puntos, con una longitud entre 3 y 20 caracteres.
 - (1) RegEx: `^[a-zA-Z0-9._]{3,20}$`
 - (2) NOT NULL

- ii) Correo: Debe seguir el formato estándar de email ([usuario@dominio.ext](#)), Y permite letras, números, puntos, guiones y subdominios.
 - (1) RegEx: `^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$`
 - (2) NOT NULL
- iii) Contraseña: Mínimo de 8 caracteres y debe incluir al menos una letra mayúscula, una minúscula, un número y un carácter especial.
 - (1) RegEx: `^(?=.*[A-Z])(?=.*[a-z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$`
 - (2) NOT NULL
- iv) FechaRegistro: TIMESTAMP
 - (1) NOT NULL

2) Partida: Contiene toda la información del autoguardado de partidas, servirá para “checkpoints” después de cada sala y la continuidad para el jugador.

a) Primary Key: ID_Partida INT PRIMARY KEY AUTO_INCREMENT

b) Foreign Keys:

- i) ID_Jugador: Relación 1:N debido a que el usuario puede tener varias partidas iniciadas y sin terminar.
- ii) ID_Clase: Relación 1:1 porque por partida, el jugador solo puede tener una clase, elegida al comienzo de la misma.

c) Atributos:

- i) Tiempo: TIME
- ii) Puntuacion: TINYINT DEFAULT 0
 - (1) Este valor se calcula a partir de la suma de los enemigos eliminados con sus valores correspondientes (enemigos comunes: 1, enemigos fuertes: 3, jefes: 10).
- iii) Nivel_actual: TINYINT DEFAULT 1 (Uso de trigger para incrementar al pasar de nivel).
- iv) Sala_actual: TINYINT DEFAULT 1 (Uso de trigger para incrementar al pasar de sala).
- v) Longitud_maldicion: DECIMAL(5,2)

3) Partida_Armas: Esta es una tabla intermedia que permite vincular múltiples armas a una misma partida, permitiendo que también se registre cuándo se encontró cada una. El uso de esta relación N:N permite que la partida tenga múltiples armas encontradas a lo largo de la partida, sin necesidad de almacenar directamente en la tabla “Partida” un listado enorme. También podrá facilitar el manejo del inventario al reanudar la partida.

a) Foreign Keys:

- i) ID_Partida: INT NOT NULL → FOREIGN KEY (id_partida) REFERENCES Partida(id_partida)
- ii) ID_Arma: INT NOT NULL → FOREIGN KEY (id_armas) REFERENCES Armas(id_armas)

- 4) Partida_Enemigo: Tabla intermedia entre “Partida” y “Enemigo” que permite contar cuántos enemigos fueron generados, derrotados y cuantos puntos otorgan al jugador. Esta tabla reemplaza el enfoque anterior de guardar contadores sueltos por tipo. Aquí se almacena el seguimiento completo y detallado.
 - a) Foreign Keys:
 - i) ID_Partida: Hace referencia a la tabla partida. FOREIGN KEY INT NOT NULL
 - ii) ID_Enemigo: Hace referencia a la tabla enemigo. FOREIGN KEY INT NOT NULL
 - b) Atributos:
 - i) EneGenerados: Número de enemigos generados. TINYINT DEFAULT 0
 - ii) EneDerrotados: Número de enemigos derrotados. TINYINT DEFAULT 0
 - iii) Puntos: Creado a partir de un trigger que calcula los puntos con base en la puntuación que otorga cada enemigo.

- 5) Enemigo: Tabla que guarda el nombre de los enemigos, los tipos de enemigos, los biomas a los que corresponden, su nivel y HP dependiendo de su nivel.
 - a) Primary Key: ID_Enemigo INT PRIMARY KEY AUTO_INCREMENT
 - b) Atributos:
 - i) NombreEnemigo: VARCHAR(20) (esqueleto, duende, ...)
 - ii) TipoEnemigo: ENUM (común, fuerte, jefe)
 - iii) Bioma: VARCHAR(20) (desierto, bosque, nieve)
 - (1) Restricciones de enemigos por bioma:
 - (a) Desierto: Esqueletos
 - (b) Bosque: Duendes y lagartos
 - (c) Nieve: Minotauros y lobos
 - iv) Nivel: SMALLINT
 - v) HP: SMALLINT
 - (1) Restricciones de vida por nivel (comunes, fuertes, jefes):
 - (a) Nivel 1: 70, 100, 150
 - (b) Nivel 2: 200, 350, 500
 - (c) Nivel 3: 1000, 1500, 2000
 - vi) Nota: En principio no se guarda el alcance porque todos los enemigos tendrán un alcance de cuerpo a cuerpo, al menos en la fase inicial del proyecto.

- 6) Clases: En esta tabla se guarda la clase seleccionada desde el inicio de la partida para configurar el daño, daño cargado y alcance del jugador
 - a) Primary Key: ID_Clase INT PRIMARY KEY AUTO_INCREMENT

- b) Atributos:
 - i) NombreClase: VARCHAR(20) (Guerrero, Arquero, Hechicero)
 - ii) DanoBasico: TINYINT NOT NULL
 - (1) Daño básico (guerrero, arquero, hechicero): 35,15,20
 - iii) DanoCargado: TINYINT NOT NULL
 - (1) Daño cargado: 60,40,50
 - iv) Alcance: TINYINT NOT NULL
 - (1) Alcance: 0, 40, 20

7) Armas: Tabla que guarda las armas secundarias, sus daños, distancia, efecto y duración para actualizarlo en el jugador, se conecta a través de una tabla intermedia con la tabla Partida.

- a) Primary Key: ID_Arma INT PRIMARY KEY AUTO_INCREMENT
- b) Atributos:
 - i) Nombre: VARCHAR(20) (Ballesta, HachaDeGuerra, Lanza, Daga)
 - ii) DanoArma: TINYINT NOT NULL
 - (1) Daño (Ballesta, HachaDeGuerra, Lanza, Daga): 40, 45, 40, mismo que la clase (extraído de la tabla Clases)
 - iii) DistanciaArma: TINYINT NOT NULL
 - (1) Distancia (Ballesta, HachaDeGuerra, Lanza, Daga): 30,0,10,0
 - iv) EfectoArma: VARCHAR(20)
 - (1) Solo la hacha de guerra: Reduce tiempo de ataque a la mitad
 - v) DuracionArma: TINYINT DEFAULT 60 (no se puede cambiar)
 - vi) UsaDanoClase: BOOLEAN DEFAULT FALSE
 - (1) CONSTRAINT chk_danoArma CHECK (
 - (a) (usaDanoClase = FALSE AND danoArma IN (40, 45))
 - OR
 - (2) (usaDanoClase = TRUE AND danoArma IS NULL)
 - (3))

8) Objetos: Tabla que guarda los objetos encontrados por el jugador a lo largo de la partida. Guarda el tipo de objeto encontrado y la recompensa relacionada.

- a) Primary Key: ID_Objeto INT PRIMARY KEY AUTO_INCREMENT
- b) Foreign Keys:
 - i) ID_Recompensa Relación 1:N → INT NOT NULL FOREIGN KEY (id_recompensa) REFERENCES Recompensas(id_recompensa)
- c) Atributos:
 - i) TipoObjeto (curandero, armero, cofre) VARCHAR(20)

9) Partida_Objeto: Tabla intermedia que va a relacionar los objetos generados o encontrados en una partida determinada.

- a) Foreign Keys:
 - i) ID_Partida: Relación N:1 porque una partida puede tener varios objetos generados, pero cada registro se asocia a una sola partida.

- ii) ID_Objeto: Relación N:1 porque un objeto puede ser generado en muchas partidas, pero cada registro de esta tabla hace referencia a un solo tipo de objetos

b) Atributos:

- i) ObjGenerado: TINYINIT (NOT NULL)

10) Recompensas: Tiene toda la definición de todas las recompensas posibles en el juego. Esta tabla permite desacoplar los efectos de los objetos, facilitando ajustes de balance sin modificar entidades externas.

- a) Primary Key: ID_Recompensa INT AUTO_INCREMENT

b) Atributos:

- i) TipoR: ENUM('vida', 'arma', 'pocima')
- ii) ValorR: VARCHAR(20)