



Ejercicio de Modelación de Base de Datos del reto

Integrantes:

Luis Emilio Veledíaz Flores -

Valentina Castilla Melgoza

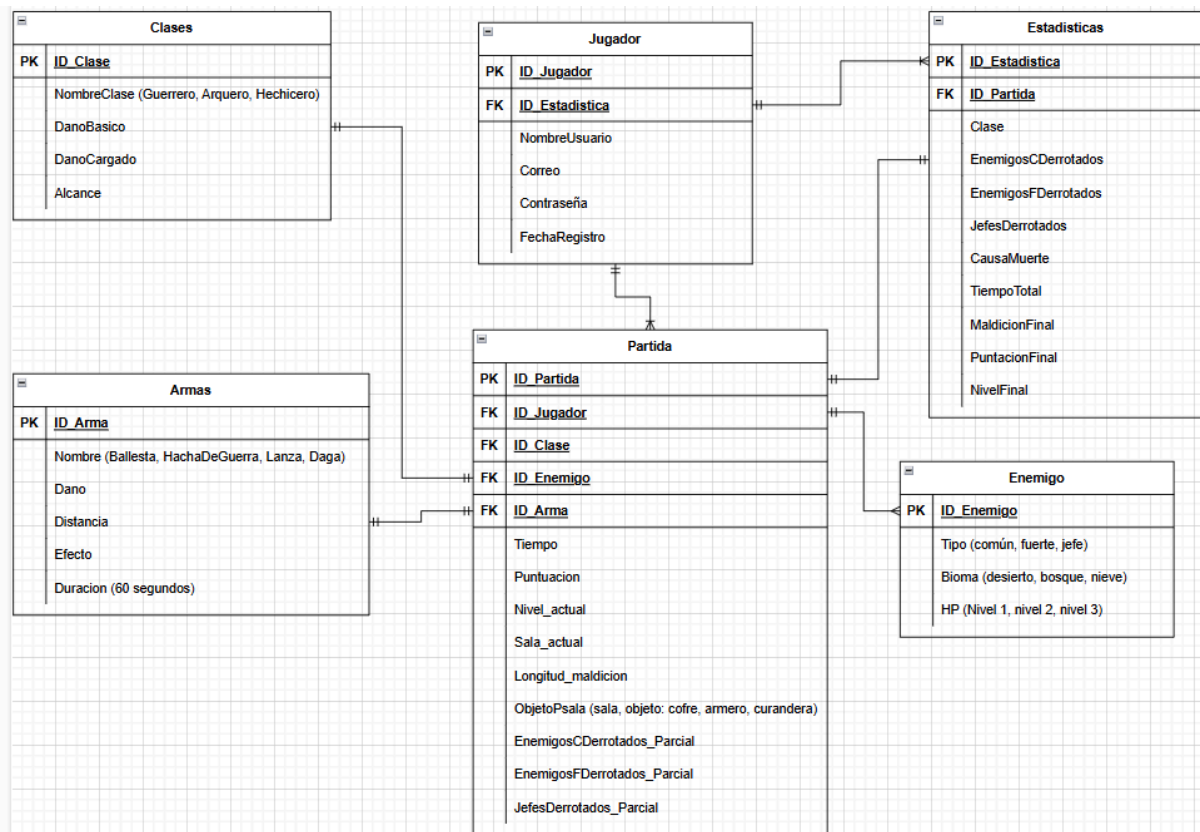
Diego de la Vega Saishio

Campus Santa Fe

Construcción de software y toma de decisiones (Gpo 401)

Esteban Castillo Juarez

Viernes 14 de marzo del 2025



Link al modelo:

<https://drive.google.com/file/d/1aFsmwACIHLmOv1gcKs2nngAER2xKR1CO/view?usp=sharing>

Justificación del diagrama Entidad-Relación

- 1) Jugador: En esta tabla se registra el usuario, con su contraseña
 - a) Primary Key: ID_Jugador INT PRIMARY KEY AUTO_INCREMENT
 - b) Foreign Key: ID_Estadistica
 - i) Relación 1 a N: La justificación de esta relación es que cada usuario tiene un set de estadísticas, pues pueden tener varias partidas creadas.
 - 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: En esta tabla se lleva el autoguardado de la partida al pasar un checkpoint
(uso de triggers)

- 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.
 - iii) ID_Enemigo: Relación 1:N porque en una partida hay 15 enemigos comunes, 3 fuertes y 1 jefe, y se necesitan guardar estos valores después de cada checkpoint . Este número después se restara del total máximo por nivel en JavaScript, para generar aleatoriamente los enemigos restantes al cargar la partida nuevamente.
 - iv) ID_Arma: Relación 1:1, más que guardar el arma que se encontró, para lo que servirá esta tabla es para guardar el número de armas que se encontraron en el nivel, para mandar el dato y desde JavaScript poder generar los restantes al cargar la partida.
- 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 (Uso de trigger para incrementar al pasar de nivel).
 - iv) Sala_actual: TINYINT (Uso de trigger para incrementar al pasar de sala).
 - v) Longitud_maldicion: DECIMAL(5,2)
 - vi) ObjetoPsala: TINYINT (guarda el número de cofres, armeros y curanderas generadas por sala, para generar aleatoriamente lo restante desde JS al cargar la partida).
 - vii) EnemigosCDerrotados_Parcial: TINYINT (Uso de trigger para sumar los valores acumulativos de enemigos comunes derrotados por nivel)
 - viii) EnemigosFDerrotados_Parcial: TINYINT (Uso de trigger para sumar los valores acumulativos de enemigos fuertes derrotados por nivel)

- ix) JefesDerrotados_Parcial: TINYINT (Uso de trigger para sumar los valores acumulativos de jefes derrotados por nivel)
- 3) Estadísticas: En esta tabla se guardara el histórico acumulativo de los valores por partida guardada. No tiene la misma función de partida, porque jalará los resultados globales para mostrárselos al jugador.
- a) Primary Key: ID_Estadistica INT PRIMARY KEY AUTO_INCREMENT
 - b) Foreign Key:
 - i) ID_Partida: Relación 1:1 porque se generará un histórico de estadísticas por partida completada (ya sea por morir o por ganar).
 - c) Atributos:
 - i) Clase: CHAR (Aquí solo se guarda la clase que se usó en la partida)
 - ii) EnemigosCDerrotados: TINYINT (se suman los enemigos comunes totales derrotados al final de la partida)
 - iii) EnemigosFDerrotados: TINYINT (se suman los enemigos fuertes totales derrotados al final de la partida)
 - iv) JefesDerrotados: TINYINT (se suman los jefes totales derrotados al final de la partida)
 - v) CausaMuerte: VARCHAR(50) (Maldicion, vida)
 - vi) TiempoTotal: TIME (tiempo total de la partida, sumado a partir de atributo Tiempo en Partida)
 - vii) MaldicionFinal: DECIMAL(5,2)
 - viii) PuntuacionFinal: TINYINT (sumado a partir de tabla partida)
 - ix) NivelFinal: TINYINT (3 si ganó el juego, menor a 3 si no)
- 4) Enemigo: Tabla que guarda los tipos de enemigos, los biomas a los que corresponden, su vida base por nivel y su alcance de daño.
- a) Primary Key: ID_Enemigo INT PRIMARY KEY AUTO_INCREMENT
 - b) Atributos:
 - i) Tipo: VARCHAR(20) (común, fuerte, jefe)
 - ii) 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
 - iii) 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
 - iv) 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.
- 5) 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
- 6) Armas: Tabla que guarda las armas secundarias y sus daños para actualizarlo en el jugador, a partir de su conexión 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) Dano: TINYINT NOT NULL
(1) Daño (Ballesta, HachaDeGuerra, Lanza, Daga): 40, 45, 40, mismo que la clase (extraído de la tabla Clases)
 - iii) Distancia: TINYINT NOT NULL
(1) Distancia (Ballesta, HachaDeGuerra, Lanza, Daga): 30,0,10,0
 - iv) Efecto: VARCHAR(20)
(1) Solo la hacha de guerra: Reduce tiempo de ataque a la mitad
 - v) Duracion: TINYINT DEFAULT 60 (no se puede cambiar)