

## **Base de Datos para Videojuego de Rol (RPG)**

### **INTRODUCCIÓN**

La temática de esta base de datos gira en torno a la creación y diseño de un emocionante videojuego de rol (RPG, por sus siglas en inglés). Los RPGs ofrecen una experiencia en la que los jugadores pueden sumergirse en mundos de fantasía, interactuar con personajes, embarcarse en misiones épicas y enfrentarse a desafíos emocionantes. Esta base de datos está diseñada para capturar todos los aspectos esenciales de esta experiencia, desde la creación de personajes hasta la gestión de inventarios y la ejecución de misiones.

### **OBJETIVO**

El objetivo de esta base de datos es brindar una plataforma sólida para la creación y gestión de un emocionante videojuego de rol. Proporciona la estructura necesaria para que los jugadores interactúen con el mundo del juego, desarrollen sus personajes, recolecten objetos, completen misiones y exploren un universo lleno de fantasía y desafíos.

### **SITUACION PROBLEMÁTICA**

Imagina que estás desarrollando un videojuego de rol (RPG) ambicioso y emocionante. Tienes un equipo de diseñadores de juegos que han creado un vasto mundo de fantasía, lleno de personajes, criaturas misteriosas y desafíos intrigantes. Sin embargo, te enfrentas a varios desafíos clave que necesitas resolver para que el juego sea un éxito:

**Gestión de Personajes:** ¿Cómo puedes permitir que los jugadores creen y gestionen sus propios personajes únicos en el juego? ¿Cómo se pueden rastrear sus niveles, experiencia, inventario y otros atributos?

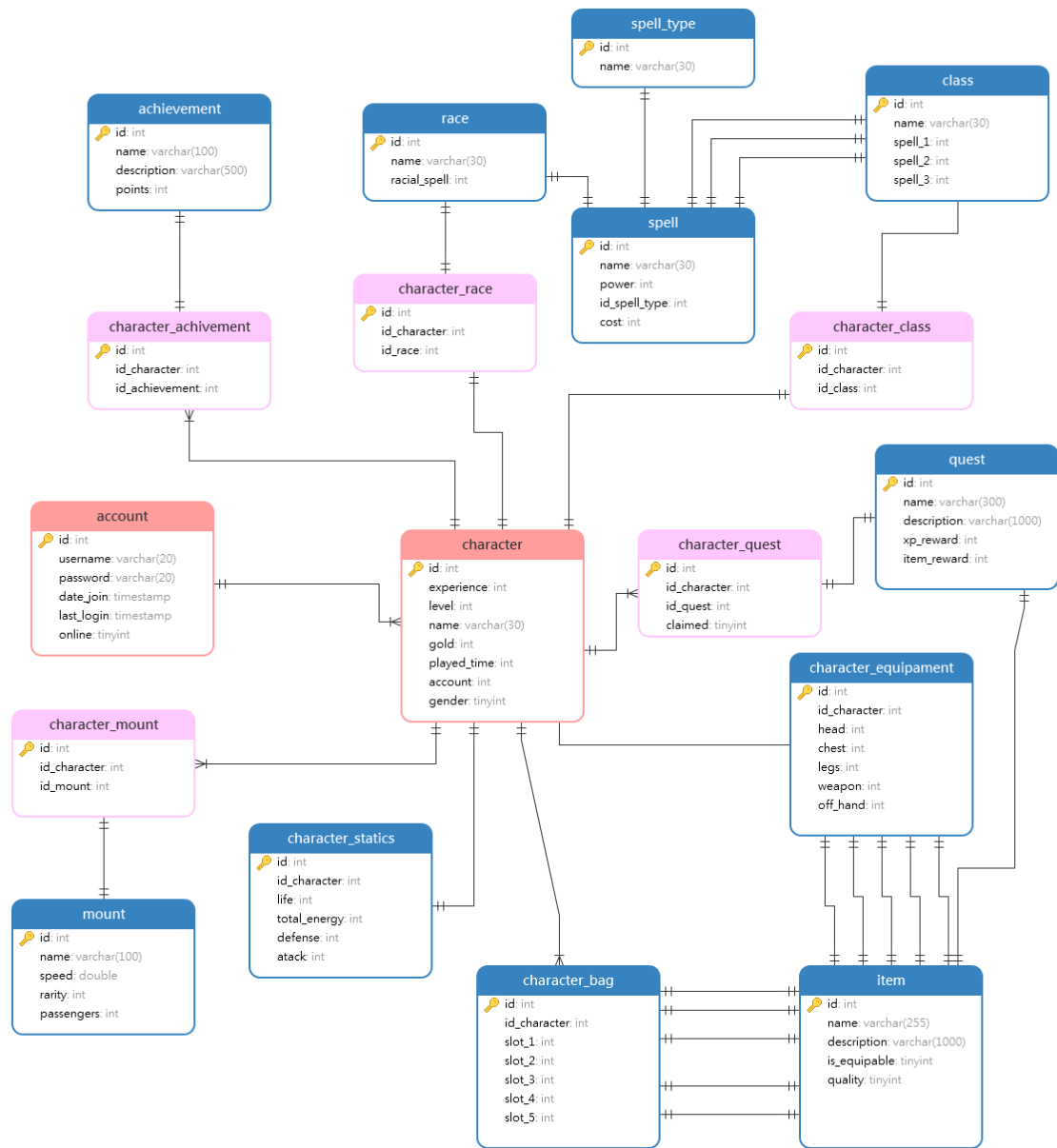
**Misiones y Logros:** ¿Cómo puedes diseñar un sistema que permita a los jugadores embarcarse en misiones emocionantes, rastrear su progreso y recompensar sus logros? ¿Cómo se registran las misiones completadas y los logros desbloqueados?

**Inventario y Equipamiento:** ¿Cómo puedes implementar un sistema de inventario que permita a los jugadores recolectar y gestionar una variedad de objetos y equipamiento en el juego? ¿Cómo se asegura que los objetos estén disponibles para su uso en el momento adecuado?

**Interacción con Personajes y Razas:** ¿Cómo se pueden gestionar las relaciones y las interacciones entre los jugadores y los personajes no jugables (PNJ) en el juego? ¿Cómo se pueden incorporar las diferentes razas de manera coherente en la trama y el juego?

**Estadísticas y Habilidades:** ¿Cómo se pueden rastrear y equilibrar las estadísticas y habilidades de los personajes para garantizar una experiencia de juego justa y emocionante?

### DIAGRAMA ER



DESCRIPCION DE CADA ENTIDAD:

TABLA: **ACCOUNT**

Nombre	Abv	Tipo	Clave	Opción/Restricción
id	id_account	int	primary key	AUTO INCREMENT
username	user	varchar(20)	index	NOT NULL, UNIQUE
password	pass	varchar(20)		NOT NULL
date_join	join	timestamp		NOT NULL
last_login	last	timestamp		NOT NULL
online	online	tinyint		NOT NULL

**La tabla account en la cual se indicarán las diferentes cuentas de usuarios del juego. Tenemos diferentes campos para definir la tabla los cuales van discriminados de la siguiente forma:**

**id\_account:** Clave principal de la tabla e id de la cuenta del jugador.

**username:** Usuario del personaje.

**password:** Password de la cuenta.

**date\_join:** Fecha de creación de la cuenta.

**last\_login:** Ultimo registro de logeo de la cuenta.

**online:** Establece si esta online la cuenta.

TABLA: **CHARACTER**

Nombre	Abv	Tipo	Clave	Opción/Restricción
id	id	int	primary key	AUTO INCREMENT
experience	xp	int		NOT NULL, Default 0
level	level	int		NOT NULL, Default 1
name	name	varchar(30)	index	NOT NULL, UNIQUE
gold	gold	int		NOT NULL, Default 0
played_time	time	int		NOT NULL, Default 0
account	acc	int	FK (account)	NOT NULL
gender	gender	tinyint		NOT NULL

**La tabla characters, es la tabla principal de los personajes la cual ira enlazada a diferentes tablas mediante llaves foráneas las cuales definen la estructura completa de un personaje.**

Tenemos diferentes campos para definir la tabla los cuales van discriminados de la siguiente forma:

**id\_character:** Clave principal de la tabla e id del personaje creado.

**experience:** En este campo va definida la experiencia actual del personaje.

**level:** Nivel actual del personaje, este irá incrementando de acuerdo a la xp actual.

**name:** Nombre del personaje y también puede usarse como índice y único.

**gold:** Oro del personaje.

**played\_time:** Tiempo total de juego del personaje calculado en segundos.

**account:** Clave foránea de la tabla account, en la cual se discriminan las cuentas de los usuarios.

**gender:** El juego contará con dos géneros los cuales se usarán 0 y 1 para determinar el género  
0 Macho, 1 Hembra.

**TABLA: CHARACTER\_CLASS**

Nombre	Abv	Tipo	Clave	Opción/Restricción
id	Id	int	primary key	AUTO INCREMENT
id_character	char	int	FK (character)	NOT NULL
id_class	sp_1	int	FK (class)	NOT NULL

*La tabla character\_class, es la tabla en la normalización de la tabla character y class, es la relacion entre las dos.*

**id:** Clave principal de la tabla.

**id\_character:** Es la clave foránea de la tabla characters.

**id\_class:** Es la clave foránea de la tabla class.

**TABLA: CLASS**

Nombre	Abv	Tipo	Clave	Opción/Restricción
id	Id_class	int	primary key	AUTO INCREMENT
name	classname	varchar(30)	index	NOT NULL
spell_1	sp_1	int	FK (spell)	NOT NULL
spell_2	sp_2	int	FK (spell)	NOT NULL
spell_3	sp_3	int	FK (spell)	NOT NULL

*La tabla character\_class, es la tabla en la cual se definen las clases de los personajes, asi como también se definen los spells que tienen las clases.*

**id:** Clave principal de la tabla e id la clase.

**name:** En este campo va definida el nombre de la clase.

**spell\_1, 2, 3:** Spells las cuales dispone el personaje.

**TABLA: CHARACTER\_RACE**

Nombre	Abv	Tipo	Clave	Opción/Restricción
id	id	int	primary key	AUTO INCREMENT
id_character	character	int	FK (character)	NOT NULL
id_race	race	int	FK (race)	NOT NULL

*La tabla character\_race, es la normalización de la tabla character y race, la cual une la relacion entre las dos para identificar la raza del personaje.*

**id:** Clave principal de la tabla.

**id\_character:** Clave foránea de la tabla character.

**id\_race:** Clave foránea de la tabla race.

**TABLA: RACE**

Nombre	Abv	Tipo	Clave	Opción/Restricción
id	id	int	primary key	AUTO INCREMENT
name	name	varchar(30)		NOT NULL
racial_spell	rsPELL	int	FK (spell)	NOT NULL

*La tabla race, comprende las razas disponibles en el juego.*

**id:** Clave principal de la tabla e id de la raza.

**name:** Nombre de la raza.

**racial\_spell:** Este campo es una clave foránea de la tabla spell, cada raza tiene un poder único.

**TABLA: SPELL**

Nombre	Abv	Tipo	Clave	Opción/Restricción
id	Id	int	primary key	AUTO INCREMENT
name	spellname	varchar(30)	index	NOT NULL
power	power	int		NOT NULL
id_spell_type	spelltype	int	FK (spell_type)	NOT NULL
cost	cost	int		NOT NULL

*La tabla spell comprende todas las spells del juego*

**id:** Clave principal de la tabla e id del spell.

**name:** En este campo va definido el nombre del spell.

**power:** Poder del la spell.

**id\_spell\_type:** Clave foránea del spell type , ya que cada spell tiene un tipo.

**cost:** Costo de energía del spell.

**TABLA: SPELL\_TYPE**

Nombre	Abv	Tipo	Clave	Opción/Restricción
id	Id	int	primary key	AUTO INCREMENT
name	spellname	varchar(30)	index	NOT NULL

*La tabla spell\_type comprende el tipo de spells del juego ya que hay sombra, sagrado, veneno etc*

**id:** Clave principal de la tabla e id del tipo de spell.

**name:** En este campo va definido el nombre del tipo spell.

**TABLA: CHARACTER\_ACHIEVENT**

Nombre	Abv	Tipo	Clave	Opción/Restricción
id	char_ach	int	primary key	AUTO INCREMENT
id_character	id_char	int	FK (character)	NOT NULL
id_achievment	Id_ach	int	FK (achievment)	NOT NULL

*La tabla character\_achievment es la normalización para los logros obtenidos por el jugador.*

**id:** Clave principal de la tabla e id del logro obtenido.

**id\_character:** Clave foránea referenciando al ID del personaje.

**id\_achievment:** Clave foránea referenciando el logro conseguido.

**TABLA: ACHIEVENT**

Nombre	Abv	Tipo	Clave	Opción/Restricción
id	ach	int	primary key	AUTO INCREMENT
name_achievment	name	Varchar(100)	index	NOT NULL
description_achievment	description	Varchar(500)		NOT NULL
points_achievment	points	int		NOT NULL, Default 0

*La tabla achievent es el listado de logros conseguibles en el juego.*

**id:** Clave principal de la tabla e id del logro.

**name\_achievment:** Nombre del logro.

**description\_achievment:** Una descripción completa del logro.

**points\_achievment:** Puntos que se obtienen al obtener el logro.

TABLA: **CHARACTER QUEST**

Nombre	Abv	Tipo	Clave	Opción/Restricción
id	charquest	int	primary key	AUTO INCREMENT
id_character	character	int	FK(character)	NOT NULL
id_quest	Id_quest	int	FK(quest)	NOT NULL
claimed	claimed	tinyint		NOT NULL, Default 0

*La tabla character\_quest es la normalización las misiones tomadas y obtenidas por el jugador.*

**id:** Clave principal de la tabla e id del la misión del jugador.

**id\_character:** Clave foránea del personaje que obtiene la mision.

**id\_quest:** Clave foránea del listado de misiones del juego.

**claimed:** Declaración de si la misión está completa o no.

TABLA: **QUEST**

Nombre	Abv	Tipo	Clave	Opción/Restricción
id	quest	int	primary key	AUTO INCREMENT
name	name	Varchar(300)	index	NOT NULL
description	descr	Varchar(1000)		NOT NULL
xp_reward	xp	int		NOT NULL, Default 0
item_reward	item	int		

*La tabla quest es el listado de las misiones del juego.*

**id:** Clave principal de la tabla e id de la misión.

**name:** Nombre de la misión.

**description:** Descripción completa de la misión.

**xp\_reward:** Cantidad de experiencia que se obtiene con la misión.

**item\_rewards:** Clave foránea de la tabla item, campo en el cual dice si la quest tiene un item de recompensa.

TABLA: **CHARACTER\_BAG**

Nombre	Abv	Tipo	Clave	Opción/Restricción
id	bag	int	primary key	AUTO INCREMENT
id_character	character	int	FK(character)	NOT NULL
slot_1	slot_1	int	FK(item)	
slot_2	slot_2	int	FK(item)	
slot_3	slot_3	int	FK(item)	

slot_4	slot_4	int	FK(ítem)	
slot_5	slot_5	int	FK(ítem)	

La tabla character\_bag es la tabla que indica cuales ítems tienen las bolsas de los jugadores.

**id:** Clave principal de la tabla.

**id\_character:** Clave foránea que referencia el personaje al que pertenece la bolsa.

**slot1,2,3,4,5:** Clave foránea de la tabla item donde se define el ítem de la lista de ítems del juego que el player tiene en la bolsa.

**TABLA: CHARACTER\_EQUIPMENT**

Nombre	Abv	Tipo	Clave	Opción/Restricción
id	eq	int	primary key	AUTO INCREMENT
id_character	character	int	FK(character)	NOT NULL
head	head	int	FK(ítem)	
chest	chest	int	FK(ítem)	
legs	legs	int	FK(ítem)	
weapon	weapon	int	FK(ítem)	
off_hand	off_hand	int	FK(ítem)	

La tabla character\_equipament es la tabla que indica cuales son los ítems equipados al personaje.

**id:** Clave principal de la tabla.

**id\_character:** Clave foránea que referencia el personaje al que pertenece la bolsa.

**head, chest, legs, weapon, off\_hand:** Clave foránea de la tabla item donde se define el ítem de la lista de ítems del juego que el player tiene equipado.

**TABLA: ITEM**

Nombre	Abv	Tipo	Clave	Opción/Restricción
id	item	int	primary key	AUTO INCREMENT
name	name	Varchar(255)	index	NOT NULL
description	descr	Varchar(1000)		NOT NULL
is_equipable	equipable	tinyint		NOT NULL, Default 0
quality	legs	tinyint		NOT NULL, Default 0

La tabla item es la tabla que indica todos los ítems disponibles del juego.

**id:** Clave principal de la tabla e id del item.

**name:** Nombre del item.

**description:** Descripción completa del item.

**is\_equipable:** Esto indica si este item es equipable.

**quality:** Esta tabla indica la calidad del item, va de 0 a 4.

**TABLA: CHARACTER\_STATICS**



Nombre	Abv	Tipo	Clave	Opción/Restricción
id	stats	int	primary key	AUTO INCREMENT
id_character	character	int	FK(character)	NOT NULL
life	life	int		NOT NULL, Default 1
total_energy	energy	int		NOT NULL, Default 1
defense	defense	int		NOT NULL, Default 1
atack	atack	int		NOT NULL, Default 1

La tabla character\_stats es la tabla que indica las estadísticas de los jugadores.

**id:** Clave principal de la tabla.

**id\_character:** Clave foránea que referencia el personaje al que pertenece las estadísticas.

**life:** Vida del personaje.

**total\_energy:** Energía total del personaje para lanzar hechizos.

**defense:** Defensa del personaje, esto reduce el daño a la vida de los hechizos.

**atack:** Ataque del personaje, afecta el daño que producen los hechizos.

### TABLA: CHARACTER\_MOUNT

Nombre	Abv	Tipo	Clave	Opción/Restricción
id	char_mount	int	primary key	AUTO INCREMENT
id_character	character	int	FK(character)	NOT NULL
id_mount	mount	int	FK(mount)	NOT NULL

La tabla character\_mount es la normalización de la relación entre las monturas que tiene el jugador y el personaje.

**id:** Clave principal de la tabla.

**id\_character:** Clave foránea que referencia el personaje al que pertenece la montura.

**id\_mount:** Clave foránea que referencia a la montura.

### TABLA: MOUNT

Nombre	Abv	Tipo	Clave	Opción/Restricción
id	mount	int	primary key	AUTO INCREMENT
name	name	Varchar(100)	index	NOT NULL
speed	speed	double		NOT NULL
rarity	rarity	int		NOT NULL, Default 0
passengers	passengers	int		NOT NULL, Default 1

La tabla mount son las monturas disponibles en el juego.

**id:** Clave principal de la tabla e id de la bolsa.

**name:** Nombre de la montura.

**speed:** Velocidad de la montura.

**rarity:** Rareza de la montura va desde 0 a 3.

**passengers:** Cantidad de pasajeros que permite llevar la montura, máximo 5.

## SCRIPTS GENERADOS PARA EL ANALISIS

### Script 1: Cuentas Creadas en los Últimos Dos Meses

```
SELECT COUNT(*) AS cuentas_creadas
FROM sql_coder_rpg.account
WHERE date_join >= DATE_SUB(NOW(), INTERVAL 2 MONTH);
```

### Script 2: Nivel Promedio de Personajes por Clase

```
SELECT c.name AS clase, AVG(ch.level) AS nivel_promedio
FROM sql_coder_rpg.character ch
INNER JOIN sql_coder_rpg.character_class cc ON ch.id = cc.id_character
INNER JOIN sql_coder_rpg.class c ON cc.id_class = c.id
GROUP BY c.name;
```

### Script 3: Cantidad de Logros Obtenidos por Jugador

```
SELECT a.username AS jugador, COUNT(ca.id_achievment) AS logros_obtenidos
FROM sql_coder_rpg.account a
LEFT JOIN sql_coder_rpg.character ch ON a.id = ch.account
LEFT JOIN sql_coder_rpg.character_achievment ca ON ch.id = ca.id_character
GROUP BY a.username;
```

### Script 4: Rareza Promedio de Monturas

```
SELECT AVG(m.rarity) AS rareza_promedio  
FROM sql_coder_rpg.mount m;
```

## **HERRAMIENTAS Y TECNOLOGÍAS USADAS**

En el proyecto se utilizó las siguientes tecnologías:

- MSQL SERVER 8.0
- Para gestionar visualizar datos: MYSQL WORKBENCH, NAVICAT, SLYOG, MYSQLPROMP
- Para la creación de los script: VisualStudio Code

## **FUTURAS LINEAS**

La idea del proyecto es que escale para así poder crear una base de datos mas robusta , y poder darle escalabilidad al video juego la idea es que a futuro se implementen mas tablas mas relaciones como por ejemplo las zonas en el videojuego, los mapas, también la ubicación de los personajes actualmente , así como información mas especifica de los personajes agregándoles diferentes habilidades talentos, y muchas mas cosas que puedan hacer de base de datos para este juego algo mas explorable y así a futuro poderlo implementar en conjunto con las tecnologías para desarrollo de videojuegos.