

# Proyecto: Plataforma de Videojuegos

## 1. Descripción de la temática de la base de datos y objetivo

La presente base de datos está diseñada para gestionar una plataforma de distribución de videojuegos similar a Steam. El objetivo es proporcionar una estructura organizada y eficiente para almacenar y gestionar información relacionada con videojuegos, usuarios, compras, reseñas, desarrolladores y bibliotecas.

El objetivo de este proyecto es diseñar una base de datos que facilite la gestión de una plataforma de distribución digital de videojuegos. Este modelo permite el seguimiento de ventas, almacenamiento seguro de datos de usuario, evaluación de juegos a través de reseñas, y gestión de relaciones con desarrolladores.

## 3. Situación Problemática

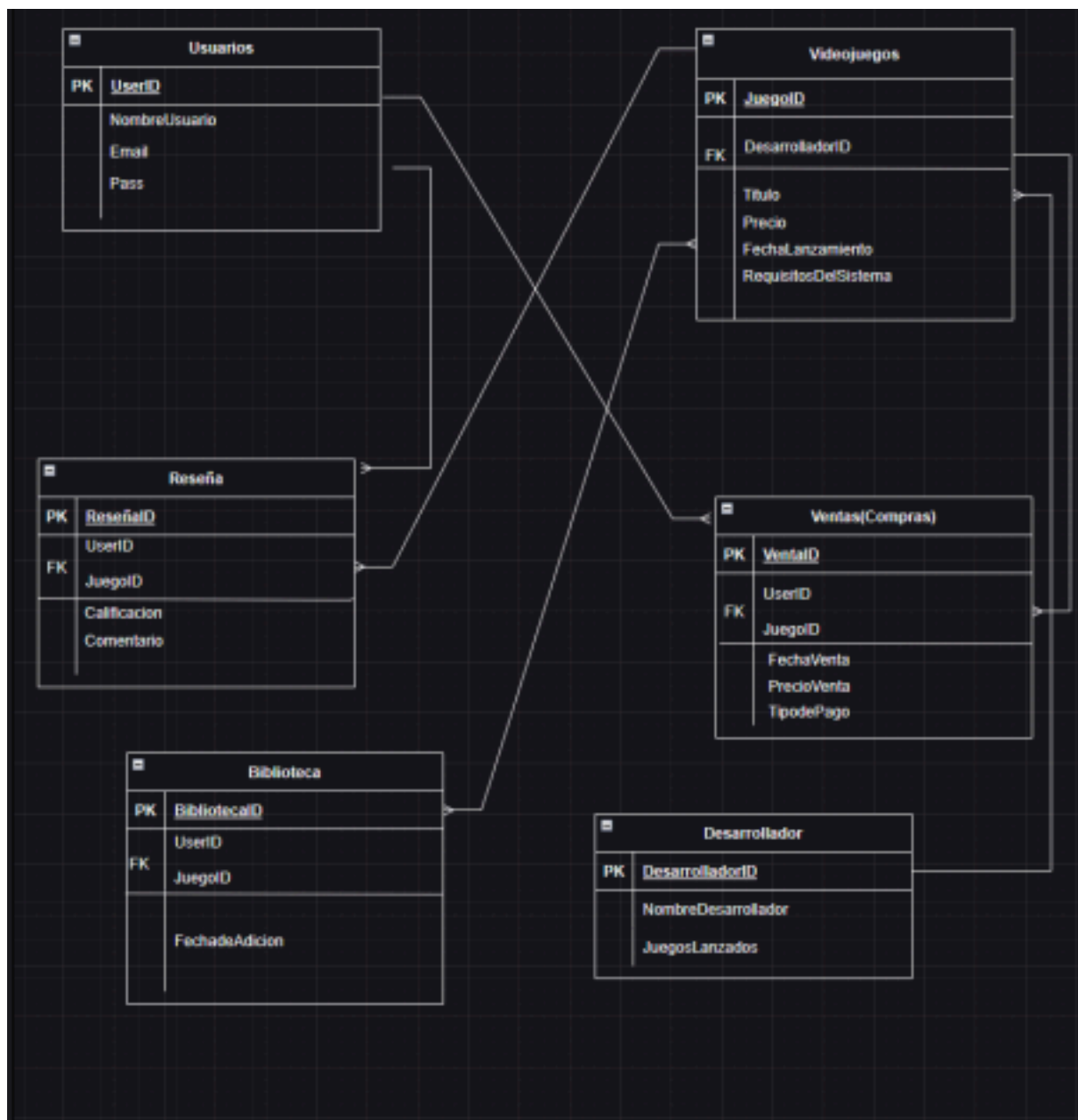
En la industria de los videojuegos, es esencial contar con un sistema robusto para gestionar grandes volúmenes de datos, incluyendo transacciones de compras, registros de usuarios y evaluaciones de juegos. La implementación de esta base de datos solucionará las brechas existentes en la gestión de estos datos, asegurando una operación eficiente y una experiencia de usuario optimizada.

## 4. Modelo de Negocio

El modelo de negocio descrito se enfoca en la organización y gestión de una plataforma de videojuegos. La base de datos diseñada permitirá a la plataforma:

- Almacenar y organizar información sobre videojuegos y sus detalles (títulos, fechas de lanzamiento, precios, etc.).
- Mantener un registro de los usuarios, sus datos personales y sus interacciones con la plataforma.
- Gestionar las transacciones de compras, facilitando un seguimiento eficiente de las ventas.
- Recopilar y almacenar reseñas de usuarios para cada videojuego, permitiendo evaluaciones y comentarios que mejoren la experiencia del cliente.
- Facilitar la relación entre la plataforma y los desarrolladores de videojuegos.

## 2. Diagrama de Entidad-Relación (E-R)



Se utilizarán las siguientes entidades y relaciones:

- Usuarios
- Desarrolladores
- Videojuegos
- Ventas
- Reseñas

Las relaciones serían:

- Un **Usuario** puede hacer varias **Ventas** (Compras).
- Un **Usuario** puede escribir varias **Reseñas**.
- Un **Videojuego** puede tener muchas **Ventas**.
- Un **Videojuego** puede tener muchas **Reseñas**.
- Un **Videojuego** pertenece a un **Desarrollador**.
- Muchos **Videojuegos** pueden tener muchas **Bibliotecas**.

### 3. Listado de tablas y descripción

**Tabla Desarrolladores:**

Campo	Abreviatura	Nombre Completo	Tipo de Dato	Tipo de Clave
desarrolladorID	DID	Desarrollador ID	INT	PRIMARY KEY
nombreDesarrollador	DID	Nombre Desarrollador	VARCHAR(100)	

**Tabla Videojuegos:**

Campo	Abreviatura	Nombre Completo	Tipo de Dato	Tipo de Clave
juegoID	JID	Juego ID	INT	PRIMARY KEY
titulo		Título	VARCHAR(100)	
desarrolladorID	DID	Desarrollador ID	INT	FOREIGN KEY

fechaLanzamiento	FechaLanz Fecha de Lanzamiento DATE
precio	Precio Precio DECIMAL(10 ,2)

**Tabla Usuarios:**

<b>Campo</b>	<b>Abreviatura</b>	<b>Nombre Completo</b>	<b>Tipo de Dato Tipo de Clave</b>
userID	UID	Usuario ID	INT PRIMARY KEY
nombreUsuario	NomUsuario	Nombre de Usuario	VARCHAR(50)
email	Email	Correo Electrónico	VARCHAR(100) UNIQUE
Password	Password	Contraseña	VARCHAR(100)

**Tabla Ventas:**

<b>Campo</b>	<b>Abreviatura</b>	<b>Nombre Completo</b>	<b>Tipo de Dato Tipo de Clave</b>
ventasID	VID	Ventas ID	INT PRIMARY KEY
userID	UID	Usuario ID	INT FOREIGN KEY
juegoID	JID	Juego ID	INT FOREIGN KEY

fechaCompra	FechaComp	Fecha de Compra	DATE
precioCompra	ra  PrecioComp  ra	Precio de Compra	DECIMAL(10,2)

tipo\_de\_pago TipoPago Tipo de Pago ENUM

**Tabla Reseña:**

Campo	Abreviatura	Nombre Completo	Tipo de Dato	Tipo de Clave
reseñalID	RID	Reseña ID	INT	PRIMARY KEY
userID	UID	Usuario ID	INT	FOREIGN KEY
juegoID	JID	Juego ID	INT	FOREIGN KEY
calificacion	Calificacion	Calificación	INT	
comentario	Comentario	Comentario	TEXT	

**Tabla Biblioteca:**

Campo	Abreviatura	Nombre Completo	Tipo de Dato	Tipo de Clave
BibliotecaID	BID	Biblioteca ID	INT	PRIMARY KEY
userID	UID	Usuario ID	INT	FOREIGN KEY
juegoID	JID	Juego ID	INT	FOREIGN KEY

fecha_de_adicion	FechaDeAdic Fecha de adición	DATE	
------------------	---------------------------------	------	--

## Vistas incluidas

### 1. Vista: vista\_usuarios\_compras

**Descripción:** Esta vista combina la información de los usuarios y las compras realizadas en la plataforma, mostrando qué videojuegos ha adquirido cada usuario junto con el precio de compra y la fecha.

**Objetivo:** Proveer un resumen detallado de las compras realizadas por cada usuario en la plataforma, permitiendo consultas rápidas para análisis de ventas, historial de compras y comportamiento del cliente.

#### Tablas involucradas:

**Usuarios:** Para obtener información del usuario (userID, nombreUsuario).

**Ventas:** Para enlazar usuarios con las compras realizadas (fechaCompra, precioCompra, juegoID).

**Videojuegos:** Para mostrar el título del videojuego adquirido.

### 2. Vista: vista\_resumen\_ventas

**Descripción:** Esta vista muestra un resumen general de las ventas realizadas en la plataforma, incluyendo el total de ingresos generados por cada videojuego, el número de ventas por título, y su precio promedio.

**Objetivo:** Ayudar a los administradores a identificar los videojuegos más vendidos, evaluar su rendimiento económico y planificar estrategias de venta.

#### Tablas involucradas:

**Ventas:** Para obtener la cantidad de ventas y los ingresos generados por cada videojuego.

**Videojuegos:** Para relacionar cada venta con el título del videojuego.

### 3. Vista: vista\_biblioteca\_usuarios

**Descripción:** Esta vista muestra el contenido de la biblioteca de cada usuario, incluyendo los videojuegos que ha agregado a su colección y la fecha en que los añadió.

**Objetivo:** Permitir a los usuarios visualizar rápidamente su biblioteca de juegos, así como su historial de adiciones. También es útil para administración al verificar datos de usuarios específicos.

**Tablas involucradas:**

**Usuarios:** Para obtener el nombre del usuario (userID, nombreUsuario).

**Biblioteca:** Para acceder a los videojuegos que cada usuario ha agregado (juegoID, fecha\_de\_adicion).

**Videojuegos:** Para mostrar los títulos y otros datos relevantes del videojuego.

## Funciones incluidas

### 1. Función: calcular\_ingreso\_total\_usuario

**Descripción:** Esta función calcula el ingreso total generado por un usuario específico en todas las compras que ha realizado.

**Objetivo:** Proveer información financiera sobre los ingresos generados por cada usuario en la plataforma, útil para análisis de comportamiento de compra y estadísticas de clientes.

**Tablas involucradas:**

**Ventas:** Para calcular el total de los ingresos provenientes de las compras realizadas por el usuario.

### 2. Función: promedio\_calificacion\_juego

**Descripción:** Esta función devuelve el promedio de calificación para un videojuego en específico, basado en las reseñas de los usuarios.

**Objetivo:** Ayudar a identificar qué videojuegos tienen mejor recepción por parte de los usuarios y realizar análisis de satisfacción de los clientes.

**Tablas involucradas:**

**Reseña:** Para calcular el promedio de las calificaciones asignadas a un videojuego.

### **3. Función: conteo\_juegos\_usuario**

**Descripción:** Esta función devuelve el número total de videojuegos que un usuario ha agregado a su biblioteca.

**Objetivo:** Permitir conocer el tamaño de la biblioteca de cada usuario, útil para analizar el nivel de compromiso y uso de la plataforma.

**Tablas involucradas:**

**Biblioteca:** Para contar los registros asociados a un usuario en particular.

## **Stored Procedures incluidas**

### **1. Stored Procedure: registrar\_nueva\_venta**

**Descripción:** Registra una nueva venta en la base de datos, incluyendo los detalles del usuario, videojuego, precio y tipo de pago.

**Objetivo:** Automatizar el registro de ventas en la plataforma, asegurando que todos los datos relevantes se inserten correctamente en la tabla Ventas.

**Tablas manipuladas:**

**Ventas:** Para insertar el registro de la venta.

### **2. Stored Procedure: actualizarPrecioVideojuego**

**Descripción:** Actualiza el precio de un videojuego específico en la base de datos.

**Objetivo:** Permitir la modificación de precios de videojuegos de manera centralizada, asegurando



que los valores se actualicen correctamente.

**Tablas manipuladas:**

**Videojuegos:** Para modificar el precio del videojuego.

### **3. Stored Procedure: agregar\_reseña**

**Descripción:** Inserta una nueva reseña para un videojuego por parte de un usuario.

**Objetivo:** Simplificar el proceso de agregar reseñas, asegurando que los datos se registren correctamente en la tabla Reseña.

**Tablas manipuladas:**

**Reseña:** Para insertar la calificación y comentario del usuario sobre un videojuego.

## **Tiggers creados**

### **1. Trigger: actualizar\_preco\_venta**

**Descripción:** Este trigger se activa antes de insertar un registro en la tabla Ventas. Calcula y actualiza el precio de compra de un videojuego, aplicando un descuento del 10% si el tipo de pago es criptomoneda.

**Objetivo:** Automatizar el cálculo de descuentos en las ventas realizadas con criptomonedas, asegurando que el precio refleje correctamente la promoción sin intervención manual.

**Tablas involucradas:**

**Ventas:** Inserta los datos con el precio ajustado.

### **2. Trigger: actualizar\_fecha\_biblioteca**

**Descripción:** Este trigger se activa después de insertar un nuevo registro en la tabla Biblioteca. Actualiza automáticamente la columna fecha\_de\_adicion con la fecha y hora actuales.

**Objetivo:** Garantizar que cada videojuego agregado a la biblioteca de un usuario tenga registrada la fecha y hora exactas de su incorporación.

**Tablas involucradas:**

**Biblioteca:** Actualiza la columna fecha\_de\_adicion después de la inserción.

## **Script en SQL**

Link del repositorio GitHub con el ScriptSQL:

<https://github.com/Gastuzar/ScriptsSQL.git>