

***CODER HOUSE***



**MySQL<sup>®</sup>**

# Proyecto final

**Nombre : Benjamin Andrés Melgarejo**

**Tutor : Camilo Andrés Redondo**

**Curso : SQL**

**Lista : 31275**

# Índice

• Descripción de la temática de la base de datos.....	4
• Diagrama entidad relación.....	5
• Listado de las tablas que comprenden la base de datos.....	7
• Tablas padres.....	8
• Tablas hijas.....	11
• Tablas intermedias.....	16
• Base de datos seleccionada.....	20
• Inserción de datos.....	31
• Creación de vistas.....	38
• Creación de funciones.....	40
• Creación de procedimientos almacenados .....	41
• Creación de triggers .....	44

# Descripción de la temática de la base de datos

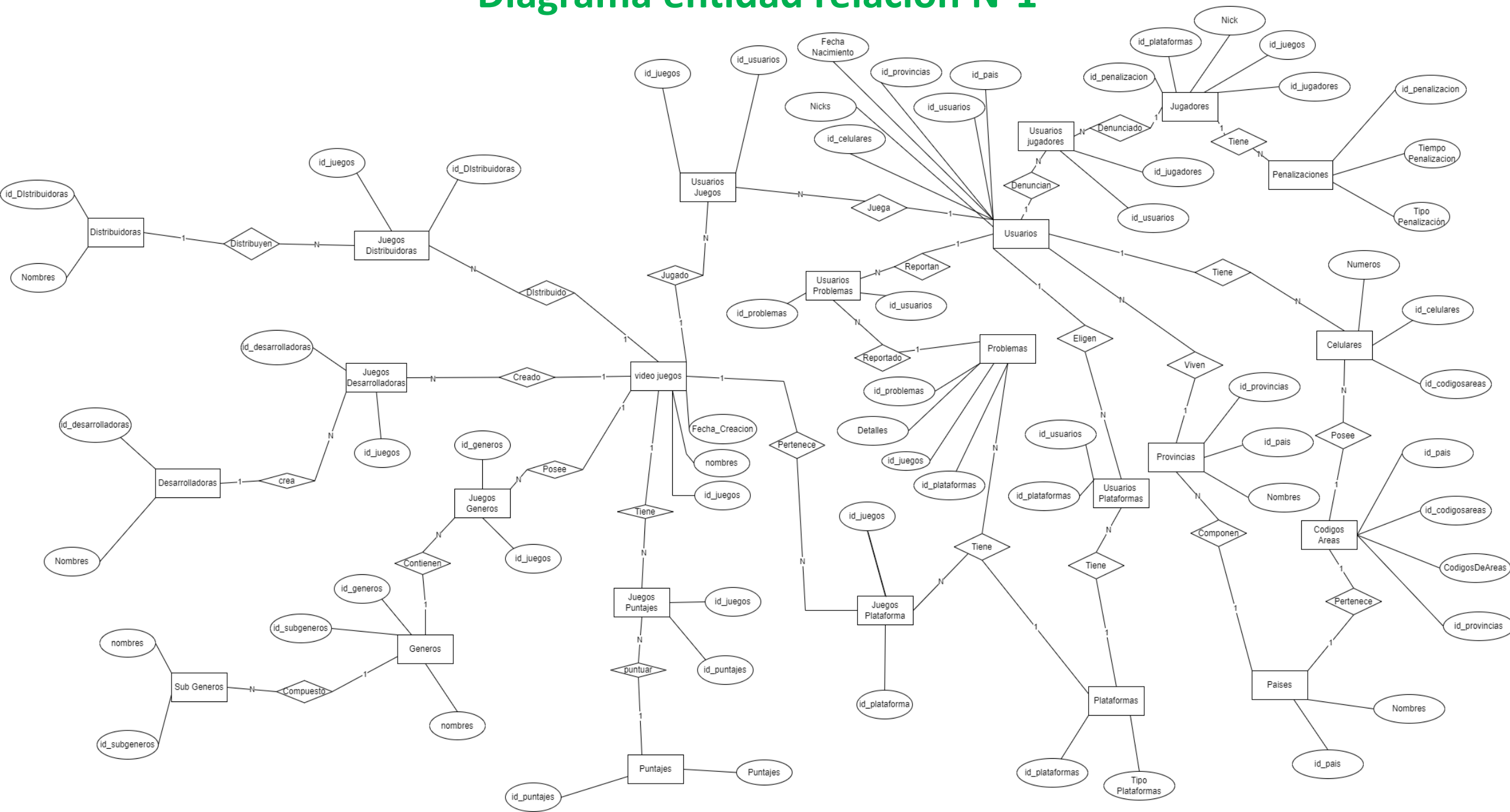
*Este proyecto final del curso de SQL que se da en la academia de CODERHOUSE. Mi proyecto final se trata de una Base de datos de un sitio web GAMING para personas que juegan a cualquier tipo de juego, indiferentemente de la plataforma o países donde estén. Además, el usuario se registra y comparte su experiencia con otros usuarios, juegos en común, plataformas, torneos, errores, en juegos o en plataformas, torneos, errores de juegos o en plataformas, etc. Esto facilita encontrar usuarios activos para determinado juegos, salteando las demoras (Matchmaking) para encontrar jugadores para partidas multijugadores por ejemplo.*

*También facilitara el reportar errores y las soluciones para los juegos, plataformas o aquellos que consideren los usuarios.*

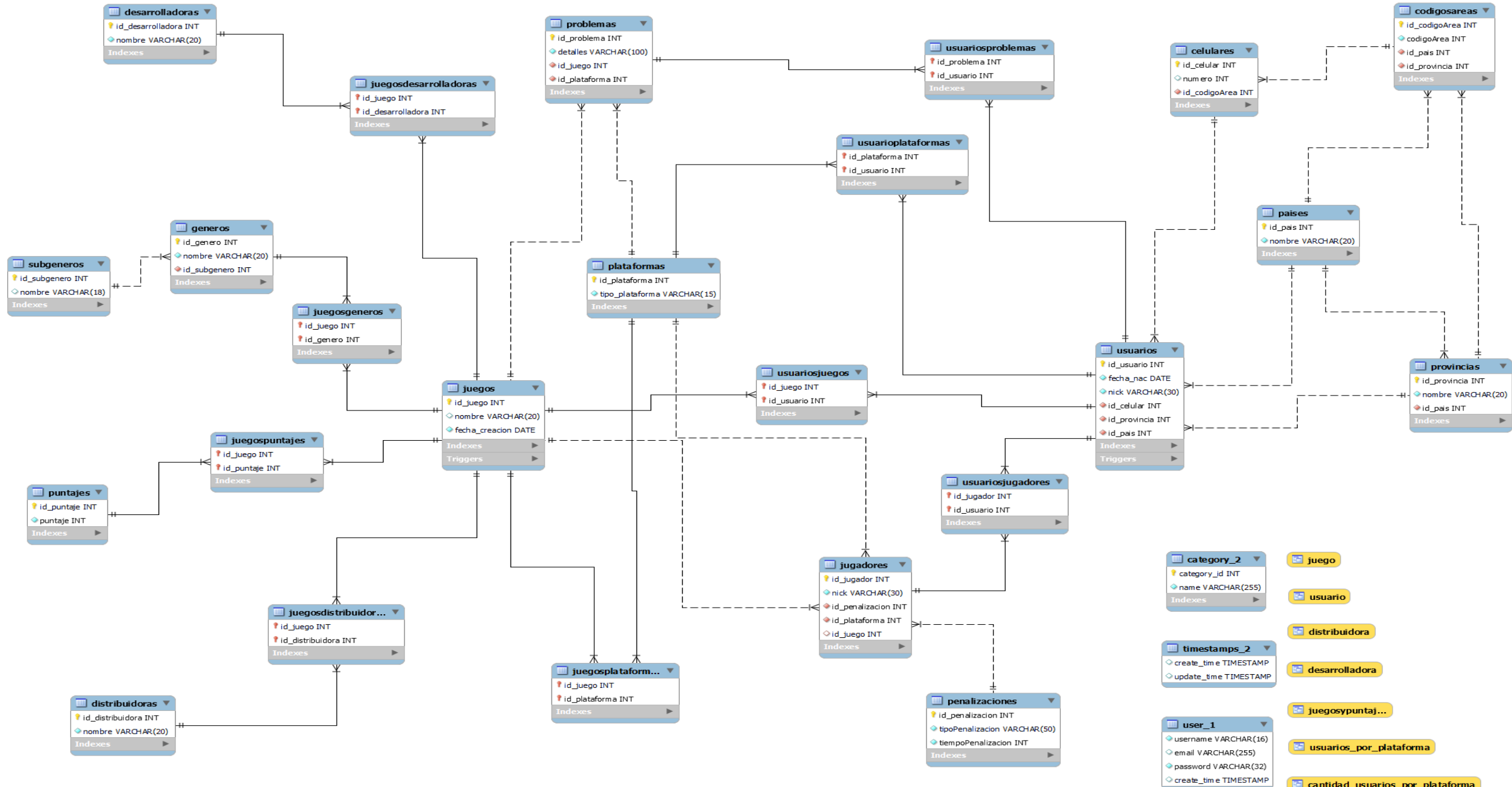
*Estos son algunos de los beneficios de la creación de la plataforma.*

*[https://github.com/Jabenid/DB\\_VideoJuegos](https://github.com/Jabenid/DB_VideoJuegos).*

## Diagrama entidad relación N°1



# Diagrama entidad relación N°2



# Listado de tablas que comprenden la base de datos

*En esta parte del recorrido se van a encontrar las tablas una por una de nuestra base de datos, divididas en tres grupos primero se introducirán las tablas padres, después las tablas hijas y por ultimo las tablas intermedias, deben contener la definición de las tablas de la base de datos junto a la descripción y contener los campos que componen dichas tablas (claves primarias, claves foráneas, campos abreviados, nombre del campo completo y los tipos de datos de cada campo). Dicha tablas son las siguientes:*

# TABLAS PADRES

Nombre de la tabla : PLATAFORMAS			
Descripción: Tabla que almacena la información de las distintas plataformas gaming.			
Campos	Tipos De Datos	Restricción	Descripción
id_plataforma	INT	AUTO_INCREMENT	Identificador único de una plataforma
tipo_plataforma	VARCHAR(15)	NOT NULL	Descripcion de una plataforma
Llave primaria:id_plataforma		Llave foránea: -	Llave única: tipo_plataforma

Nombre de la tabla : DISTRIBUIDORAS			
Descripción: Tabla que almacena la información de las distintas distribuidoras gaming.			
Campos	Tipos De Datos	Restricción	Descripción
Id_distribuidora	INT	AUTO_INCREMENT	Identificador único de una distribuidora
nombre	VARCHAR(20)	NOT NULL	Nombre de las distribuidora
Llave primaria: id_distribuidora		Llave foránea: -	Llave única: nombre



Nombre de la tabla : SUBGENERO			
Descripción: Tabla que almacena el Subgenero de cada video juego de la plataforma gaming.			
Campos	Tipos De Datos	Restricción	Descripción
Id_subgenero	INT	AUTO_INCREMENT	Identificador único de cada subgenero
nombre	VARCHAR(20)	NOT NULL	Nombre de cada subgenero
Llave primaria: Id_subgenero		Llave foránea: -	Llave única: nombre

Nombre de la tabla : PAISES			
Descripción: Tabla que almacena el Pais de cada usuario de la plataforma gaming.			
Campos	Tipos De Datos	Restricción	Descripción
Id_pais	INT	AUTO_INCREMENT	Identificador único de cada país
nombre	VARCHAR(20)	NOT NULL	Nombre de cada país
Llave primaria:		Llave foránea : -	Llave única: nombre

Nombre de la tabla : DESARROLLADORAS			
Descripción: Tabla que almacena el nombre de cada desarrolladora de la plataforma gaming.			
Campos	Tipos De Datos	Restricción	Descripción
Id_desarrolladora	INT	AUTO_INCREMENT	Identificador único de cada desarrolladora
nombre	VARCHAR(20)	NOT NULL	Nombre de cada desarrolladora
Llave primaria: id_desarrolladora		Llave foránea : -	Llave única: nombre

Nombre de la tabla : PUNTAJES			
Descripción: Tabla que almacena el puntaje de cada juego dentro de la plataforma gaming			
Campos	Tipos De Datos	Restricción	Descripción
Id_puntaje	INT	AUTO_INCREMENT	Identificador único de cada puntaje
Puntaje	INT	NOT NULL	Valor de cada puntaje
Llave primaria: id_puntaje		Llave foránea : -	Llave única: -

Nombre de la tabla : PENALIZACIONES			
Descripción: Tabla que almacena el dato de cada jugador penalizado.			
Campos	Tipos De Datos	Restricción	Descripción
Id_penalización	INT	AUTO_INCREMENT	Identificador único de cada penalización
tipoPenalizacion	VARCHAR(50)	NOT NULL	Descripción de cada penalización
tiempoPenalizacion	INT	NOT NULL	Duración de cada penalización
Llave primaria: id_penalizacion		Llave foránea : -	Llave única: -

## TABLAS HIJAS

Nombre de la tabla : PROVINCIAS			
Descripción: Tabla que almacena el dato de la provincia de cada país en el que vive el usuario de la plataforma gaming.			
Campos	Tipos De Datos	Restricción	Descripción
Id_provincia	INT	AUTO_INCREMENT	Identificador único de cada provincia
Nombre	VARCHAR(50)	NOT NULL	Nombre de cada provincia
Id_país	INT	NOT NULL	Identificador único de cada pais
Llave primaria: id_provincia		Llave foránea : id_pais	Llave única: -

Nombre de la tabla : CODIGOSAREAS			
Descripción: Tabla que almacena el dato de cada código de área de cada país.			
Campos	Tipos De Datos	Restricción	Descripción
Id_códigoArea	INT	AUTO_INCREMENT	Identificador única de cada código de área
codigoArea	INT	NOT NULL	Numero de área
Id_país	INT	NOT NULL	Identificador única de cada país
id:_provincia	INT	NOT NULL	Identificación única de cada provincia
Llave primaria: id_codigoArea		Llave foránea : id_país, id_provincia	Llave única: codigoArea

Nombre de la tabla : CELULARES			
Descripción: Tabla que almacena el dato del numero de celular de cada usuario de la plataforma gaming.			
Campos	Tipos De Datos	Restricción	Descripción
Id_celular	INT	AUTO_INCREMENT	Identificador único de cada numero de celular
Numero	INT	NOT NULL	Numero de celular
Id_codigodeArea	INT	NOT NULL	Identificador único de cada código de area
Llave primaria: id_celular		Llave foránea : id_codigoArea	Llave única: numero

Nombre de la tabla : JUEGOS			
Descripción: Tabla que almacena el nombre de cada juego			
Campos	Tipos De Datos	Restricción	Descripción
Id_juego	INT	AUTO_INCREMENT	Identificador único de cada juego
Nombre	VARCHAR(50)	NOT NULL	Nombre de cada juego
Fecha_de_creacion	DATE	NOT NULL	Fecha en la que se creo cada juego
Llave primaria: id_juego		Llave foránea : -	Llave única: -

Nombre de la tabla : PROBLEMAS			
Descripción: Tabla que almacena el dato de cada problema que en un juego o plataforma.			
Campos	Tipos De Datos	Restricción	Descripción
Id_problema	INT	AUTO_INCREMENT	Identificador única de cada problema
detalles	VARCHAR(100)	NOT NULL	Detalle del problema
Id_juego	INT	NOT NULL	Identificador única de cada juego
id:_plataforma	INT	NOT NULL	Identificador única de cada plataforma
Llave primaria: id_problema		Llave foránea : id_juego, id_plataforma	Llave única: -

Nombre de la tabla : USUARIOS

Descripción: Tabla que almacena el dato de cada Usuario.

Campos	Tipos De Datos	Restricción	Descripción
Id_usuario	INT	AUTO_INCREMENT	Identificador única de cada usuario
Fecha_nac	DATE	NOT NULL	Fecha de nacimiento del usuario
nick	VARCHAR(30)	NOT NULL	Nick de cada usuario
Id_celular	INT	NOT NULL	Numero de celular
id:_provincia	INT	NOT NULL	Identificador única de cada provincia
Id_pais	INT	NOT NULL	Identificador única de cada país
Llave primaria: id_usuario		Llave foránea : id_país, id_provincia, id_celular	Llave única: codigoArea

Nombre de la tabla : GENEROS			
Descripción: Tabla que almacena el nombre de cada genero.			
Campos	Tipos De Datos	Restricción	Descripción
Id_genero	INT	AUTO_INCREMENT	Identificador único de cada genero
Nombre	VARCHAR(20)	NOT NULL	Nombre de cada genero
Id_subgenero	INT	NOT NULL	Fecha en la que se creo cada subgénero
Llave primaria: id_genero		Llave foránea : id_subgenero	Llave única: -

Nombre de la tabla : JUGADORES			
Descripción: Tabla que almacena el dato de cada jugador.			
Campos	Tipos De Datos	Restricción	Descripción
Id_jugador	INT	AUTO_INCREMENT	Identificador única de cada jugador
Id_penalizacion	INT	NOT NULL	Identificador de cada penalización
nick	VARCHAR(30)	NOT NULL	Nick de cada usuario
Id_plataforma	INT	NOT NULL	Identificador de cada plataforma
id:_juego	INT	NOT NULL	Identificador única de cada juego
Llave primaria: id_jugador		Llave foránea : id_juego, id_plataforma, id_penalizacion	Llave única: -

# TABLAS INTERMEDIAS

## Nombre de la tabla : USUARIOSJUGADORES

Descripción: Tabla intermediaria entre jugadores y usuarios contiene el identificador único de ambas.

Campos	Tipos De Datos	Restricción	Descripción
Id_jugador	INT	NOT NULL	Identificador único de cada jugador
Id_usuario	INT	NOT NULL	Identificador único de cada usuario
Llave primaria: (id_jugador, id_usuario)		Llave foránea : (id_jugador, id_usuario)	Llave única: -

## Nombre de la tabla : USUARIOSPLATAFORMA

Descripción: Tabla intermediaria entre plataforma y usuarios contiene el identificador único de ambas.

Campos	Tipos De Datos	Restricción	Descripción
Id_plataforma	INT	NOT NULL	Identificador único de cada plataforma
Id_usuario	INT	NOT NULL	Identificador único de cada usuario
Llave primaria: (id_plataforma, id_usuario)		Llave foránea : (id_plataforma, id_usuario)	Llave única: -



Nombre de la tabla : JUEGOSDESARROLLADORA			
Descripción: Tabla intermediaria entre juegos y desarrolladora contiene el identificador único de ambas.			
Campos	Tipos De Datos	Restricción	Descripción
Id_juego	INT	NOT NULL	Identificador único de cada juego
Id_desarrolladora	INT	NOT NULL	Identificador único de cada desarrolladora
Llave primaria: (id_juego, id_desarrolladora)		Llave foránea : (id_juego, id_desarrolladora)	Llave única: -

Nombre de la tabla : JUEGODISTRIBUIDORA			
Descripción: Tabla intermediaria entre juegos y ditribuidora contiene el identificador único de ambas.			
Campos	Tipos De Datos	Restricción	Descripción
Id_juego	INT	NOT NULL	Identificador único de cada juego
Id_distribuidora	INT	NOT NULL	Identificador único de cada distribuidora
Llave primaria: (id_juego, id_distribuidora)		Llave foránea : (id_juego, id_distribuidora)	Llave única: -

Nombre de la tabla : USUARIOSPROBLEMAS			
Descripción: Tabla intermediaria entre problema y usuarios contiene el identificador único de ambas.			
Campos	Tipos De Datos	Restricción	Descripción
Id_problema	INT	NOT NULL	Identificador único de cada problema
Id_usuario	INT	NOT NULL	Identificador único de cada usuario
Llave primaria: (id_problema, id_usuario)		Llave foránea : (id_problema, id_usuario)	Llave única: -

Nombre de la tabla : USUARIOSJUEGOS			
Descripción: Tabla intermediaria entre juegos y usuarios contiene el identificador único de ambas.			
Campos	Tipos De Datos	Restricción	Descripción
Id_juego	INT	NOT NULL	Identificador único de cada juegos
Id_usuario	INT	NOT NULL	Identificador único de cada usuario
Llave primaria: (id_juego, id_usuario)		Llave foránea : (id_juego, id_usuario)	Llave única: -

Nombre de la tabla : JUEGOSPLATAFORMA			
Descripción: Tabla intermediaria entre juegos y plataformas contiene el identificador único de ambas.			
Campos	Tipos De Datos	Restricción	Descripción
Id_juego	INT	NOT NULL	Identificador único de cada juegos
Id_plataforma	INT	NOT NULL	Identificador único de cada plataforma
Llave primaria: (id_juego, id_plataforma)		Llave foránea : (id_juego, id_plataforma)	Llave única: -

Nombre de la tabla : JUEGOSGENEROS			
Descripción: Tabla intermediaria entre juegos y generos contiene el identificador único de ambas.			
Campos	Tipos De Datos	Restricción	Descripción
Id_juego	INT	NOT NULL	Identificador único de cada juegos
Id_genero	INT	NOT NULL	Identificador único de cada genero
Llave primaria: (id_juego, id_genero)		Llave foránea : (id_juego, id_genero)	Llave única: -

Nombre de la tabla : JUEGOSPUNTAJES			
Descripción: Tabla intermediaria entre juegos y puntajes contiene el identificador único de ambas.			
Campos	Tipos De Datos	Restricción	Descripción
Id_juego	INT	NOT NULL	Identificador único de cada juegos
Id_puntaje	INT	NOT NULL	Identificador único de cada puntaje
Llave primaria: (id_juego, id_puntaje)		Llave foránea : (id_juego, id_puntaje)	Llave única: -

## Script de creación de base de datos

*Luego del análisis previo realizado en la descripción de tablas, se mostrará el código sql con la finalidad de mostrar la creación de una base de datos y sus respectivas tablas.*

[https://github.com/Jabenid/DB\\_VideoJuegos/blob/main/Script.sql.proyectofinal](https://github.com/Jabenid/DB_VideoJuegos/blob/main/Script.sql.proyectofinal)

[https://github.com/Jabenid/DB\\_VideoJuegos/blob/main/Creacion\\_de\\_tablas.Script.sql](https://github.com/Jabenid/DB_VideoJuegos/blob/main/Creacion_de_tablas.Script.sql).

# Inserción de datos

*El script deberá tener la inserción de datos de las tablas generadas hasta ahora en tu proyecto. Si tu archivo es en formato .sql, deberá poder ejecutarse de una sola vez en la ventana de script de MySQL Workbench.*

*INSERCIÓN DE DATOS: la inserción de registros a cada una de las tablas se realizó de manera manual, aunque se puede utilizar el asistente para la importación, esta herramienta se utiliza para una carga masiva de datos. Al igual que en la creación de las tablas, la inserción también se tuvo en cuenta el orden ya que hay subconsultas para inserción de los valores, primero se insertan los registros de las tablas padres, siguen los registro de tablas hijas y por último los registros de la tablas intermedias.*

[https://github.com/Jabenid/DB\\_VideoJuegos/blob/main/Insercion\\_de\\_datos.Script.sql](https://github.com/Jabenid/DB_VideoJuegos/blob/main/Insercion_de_datos.Script.sql).

# Creación de vistas

*Se deberá presentar en formato .sql el script de creación de cinco vistas con base en los datos de la base de datos del proyecto final. En base a las tablas elegidas para el proyecto, deberás idear cinco más vistas de una o más tablas.*

*VISTAS: en esta sección se refiere a las vistas de las tablas presenten información más relevante.*

*Las vistas creadas son: Usuarios: muestra los datos de los usuarios. Juego: muestra los datos de los juegos. Juegos y puntajes: muestra los juegos y sus puntajes. Desarrolladora: muestra los datos de las desarrolladoras. Distribuidora: muestra los datos de las distribuidoras. Cantidad de usuarios por plataforma: muestra la cantidad de usuarios por plataforma.*

[https://github.com/Jabenid/DB\\_VideoJuegos/blob/main/Creacion\\_de\\_vistas.Script.sql](https://github.com/Jabenid/DB_VideoJuegos/blob/main/Creacion_de_vistas.Script.sql).

# Creación de Funciones

*Se deberá presentar el script de creación de dos funciones almacenadas con base en los datos del proyecto final. Se podrá incluir una función que haga uso interno de funciones propias de SQL, y una segunda función que permita obtener valores de otra tabla, reemplazando, un parámetro de entrada.*

***FUNCIONES:** son códigos de SQL que reciben datos de entrada, realizan operaciones con ellos y luego devuelven un resultado. Se realizan 2 funciones, su finalidad se encuentran comentadas al inicio de ejecutar cada una de ellas.*

***1ra FUNCION:** la funcion se llama usuarios\_por\_ plataforma, se le da como atributo un varchar (ej:pc) y después esta devuelve la cantidad de usuarios que pertenecen a la plataforma.*

***2da FUNCIÓN:** la función se llama CANTIDAD\_DE\_ERRORES, se le asigna 2 atributos varchar (1ro nombre de un juego y 2do una plataforma) y después esta función devuelve la cantidad de errores que tiene el juego y la plataforma.*

[https://github.com/Jabenid/DB\\_VideoJuegos/blob/main/Creacion\\_de\\_Funciones.script.sql](https://github.com/Jabenid/DB_VideoJuegos/blob/main/Creacion_de_Funciones.script.sql)

# Creación de procedimientos almacenados

***STORED PROCEDURES:*** Su objetivo es realizar una tarea determinada, desde operaciones sencillas hasta tareas muy complejas. Se realizan 4 stored procedures, su finalidad se encuentran comentadas al inicio de ejecutar cada una de ellos.

***1re Store Procedure:*** este stored procedure ordena la tabla juegos según un campo de la tabla y si coloca 1 lo ordena de manera descendente.

***2do Store Procedure:*** este stored procedure hace lo mismo que el anterior con la diferencia que se aplica a la tabla usuarios.

***3er Store Procedure:*** este stored procedure se crea de manera de ejemplo , se debería hacer para las demás tablas, lo cual varía la cantidad de campos y datos a insertar según la tabla.

***4to Store Procedure:*** este stored procedure se crea de manera genérica para elegir la tabla e insertar un registro nuevo

***en el campo nombre, en la tabla que se indique.***

***En mi caso se puede aplicar a 4 tablas: paises, sugeneros, desarrolladoras y distribuidoras. Esta tabla tiene en común el campo nombre.***



# Creación de Triggers

**Triggers:** Los disparadores o triggers son objetos cuyo objetivo es ejecutar el código en respuesta a un evento que ocurre en una tabla. Los eventos pueden ser de tres tipos:

**INSERT, UPDATE o DELETE.** Se crearon 6 triggers, esto se aplicaron a las tablas que son más importantes pero también se puede extender a las demás tablas. La funcionalidad de cada trigger está comentado antes de ejecutar el código.

**1er Trigger:** este trigger se dispara antes de eliminar un registro de la tabla usuarios e inserta datos en la tabla log\_del\_usuario sobre:

Que id de usuario y Nick se borró, que usuario lo hizo y la fecha y hora en que se eliminó.

**2do Trigger:** este trigger se dispara después de que se inserte un registro en la tabla usuarios y crea un registro en la tabla log\_ins\_usuario de que, quien y cuando se insertó el registro.

**3er Trigger:** este trigger se dispara después de que se actualice un registro en la tabla usuarios y crea un registro en la tabla log\_upd\_usuario de que, quien y cuando se insertó el registro.

**4to Trigger:** este trigger es parecido al 1ro, se dispara antes de borrar un registro de la tabla juegos.

**5to Trigger:** este trigger se dispara después de que se inserte un registro en la tabla juegos y crea un registro en la tabla log\_ins\_juego de que, quien y cuando se insertó el registro.

**6to Trigger:** este trigger es parecido al 3ro, se dispara después de actualizar un registro de la tabla juegos.

[https://github.com/Jabenid/DB\\_VideoJuegos/blob/main/Creacion\\_de\\_Triggers.script.sql](https://github.com/Jabenid/DB_VideoJuegos/blob/main/Creacion_de_Triggers.script.sql)

# Sentencias DCL

***Sentencias DCL: estas sentencias se utiliza para otorgar, mostrar y revocar(revoke, no se utilizó) permisos a un usuario determinado.***

[https://github.com/Jabenid/DB\\_VideoJuegos./blob/main/Creacion\\_de\\_Sentencias\\_DCL.Script.sql](https://github.com/Jabenid/DB_VideoJuegos./blob/main/Creacion_de_Sentencias_DCL.Script.sql).

# Sentencias TCL

***Sentencias TCL (commit y rollback): se utilizan para ejecutar código y aprobar dicha transacción(commit), volver a un punto de guardado(savepoint) o volver la bd a el mismo que estado que estaba antes de ejecutar cualquier transacción(rollback).***

[https://github.com/Jabenid/DB\\_VideoJuegos./blob/437e4eb45fb1141501399f654bfad8d6a9417562/Creacion\\_de\\_Sentencias\\_TCL.Script.sql](https://github.com/Jabenid/DB_VideoJuegos./blob/437e4eb45fb1141501399f654bfad8d6a9417562/Creacion_de_Sentencias_TCL.Script.sql).