

## **Documentación de la Base de Datos**

### **1. Descripción General**

Este proyecto replica la estructura de una plataforma de streaming como Netflix. La base de datos fue diseñada en MySQL e implementa un sistema de roles, usuarios, planes de suscripción, contenidos (películas y series), favoritos, historial de visualización y tendencias.

### **2. Modelo Entidad-Relación (DER)**

#### **Entidades Principales:**

- **usuarios:** almacena la información de los usuarios.
- **roles:** define los distintos tipos de rol para los usuarios (admin, cliente, etc).
- **planes:** define los planes de suscripción.
- **generos:** categoría de películas y series.
- **peliculas / series:** contenido multimedia.
- **favoritos:** relación de favoritos de usuarios con contenido.
- **historial:** registro de contenido visto por usuario.
- **tendencias:** registro de visualizaciones por contenido.

### 3. Modelo Relacional

#### Tabla: **roles**

```
CREATE TABLE roles (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(50) NOT NULL,  
  descripcion TEXT,  
  fecha_creacion DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```

#### Tabla: **planes**

```
CREATE TABLE planes (  
  id_plan INT AUTO_INCREMENT PRIMARY KEY,  
  nombre_plan VARCHAR(50),  
  precio DECIMAL(10,2),  
  resolucion_maxima VARCHAR(20),  
  cantidad_perfiles INT  
);
```

#### Tabla: **generos**

```
CREATE TABLE generos (  
  id_genero INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(50)  
);
```

#### Tabla: **usuarios**

```
CREATE TABLE usuarios (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  username VARCHAR(20),  
  mail VARCHAR(100),  
  pass VARCHAR(30),  
  id_plan INT,  
  rol_id INT,  
  nombre VARCHAR(100),  
  apellido VARCHAR(100),  
  numero_tarjeta VARCHAR(30),  
  FOREIGN KEY (id_plan) REFERENCES planes(id_plan),  
  FOREIGN KEY (rol_id) REFERENCES roles(id)  
);
```

**Tabla: películas**

```
CREATE TABLE peliculas (  
  id_pelicula INT AUTO_INCREMENT PRIMARY KEY,  
  id_genero INT,  
  titulo VARCHAR(200),  
  descripcion VARCHAR(200),  
  clasificacion_edad VARCHAR(200),  
  duracion INT,  
  fecha_lanzamiento DATETIME,  
  foto VARCHAR(2000),  
  trailer VARCHAR(2000),  
  FOREIGN KEY (id_genero) REFERENCES generos(id_genero)  
);
```

**Tabla: series**

```
CREATE TABLE series (  
  id_serie INT AUTO_INCREMENT PRIMARY KEY,  
  id_genero INT,  
  titulo VARCHAR(200),  
  descripcion VARCHAR(200),  
  clasificacion_edad VARCHAR(200),  
  duracion INT,  
  fecha_lanzamiento DATETIME,  
  foto VARCHAR(2000),  
  trailer VARCHAR(2000),  
  FOREIGN KEY (id_genero) REFERENCES generos(id_genero)  
);
```

**Tabla: favoritos**

```
CREATE TABLE favoritos (  
  id_favorito INT AUTO_INCREMENT PRIMARY KEY,  
  id_usuario INT,  
  fecha_agregado DATETIME,  
  id_pelicula INT,  
  id_serie INT,  
  FOREIGN KEY (id_usuario) REFERENCES usuarios(id),  
  FOREIGN KEY (id_pelicula) REFERENCES peliculas(id_pelicula),  
  FOREIGN KEY (id_serie) REFERENCES series(id_serie)  
);
```

**Tabla: historial**

```
CREATE TABLE historial (  
  id_historial INT AUTO_INCREMENT PRIMARY KEY,  
  id_usuario INT,  
  fecha_visto DATETIME,  
  progreso_minutos INT,  
  id_pelicula INT,  
  id_serie INT,  
  FOREIGN KEY (id_usuario) REFERENCES usuarios(id),  
  FOREIGN KEY (id_pelicula) REFERENCES peliculas(id_pelicula)  
    ON DELETE CASCADE,  
  FOREIGN KEY (id_serie) REFERENCES series(id_serie)  
    ON DELETE CASCADE  
);
```

**Tabla: tendencias**

```
CREATE TABLE tendencias (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  id_pelicula INT,  
  id_serie INT,  
  visualizaciones INT DEFAULT 0,  
  FOREIGN KEY (id_pelicula) REFERENCES peliculas(id_pelicula),  
  FOREIGN KEY (id_serie) REFERENCES series(id_serie)  
);
```

---

## 4. Inserts Iniciales

```
INSERT INTO roles (nombre, descripcion) VALUES  
( 'admin', 'Administrador del sistema'),  
( 'cliente', 'Usuario final de la plataforma');
```

```
INSERT INTO planes (nombre_plan, precio, resolucion_maxima, cantidad_perfiles) VALUES  
( 'Básico', 7.99, '480p', 1),  
( 'Estándar', 11.99, '1080p', 2),  
( 'Premium', 15.99, '4K', 4);
```

```
INSERT INTO generos (nombre) VALUES ('Acción'), ('Comedia'), ('Drama'), ('Ciencia  
Ficción');
```

```
INSERT INTO usuarios (username, mail, pass, id_plan, rol_id, nombre, apellido,
numero_tarjeta) VALUES
('agus_updated', 'nuevo@mail.com', 'nuevaClave123', 1, 1, 'Agus', 'Actualizado',
'1111222233334444');
```

## 5. Diagrama UML (describir)

El diagrama UML representa las clases siguientes:

- Usuario
- Pelicula
- Serie
- Favorito
- Historial
- Tendencia

Relaciones:

- Usuario tiene muchos Favoritos
- Usuario tiene muchos Historiales
- Pelicula o Serie puede estar en muchas Tendencias y muchos Historiales

## 6. Consideraciones Finales

- La base está normalizada hasta 3FN.
- Existen restricciones de clave foránea para mantener la integridad referencial.
- Se implementaron triggers y controladores para actualizar tendencias automáticamente.
- La estructura está preparada para expandirse con funcionalidades como perfiles, comentarios o reproducciones.