

# DESAFÍO ENTREGABLE 1

## SQL Flex



**Profesor:** Camilo Andres Redondo

**Tema**

- ❖ Diseño y modelado de una base de datos para una plataforma de streaming musical

**Autor**

- ❖ Luis Angel Umiña Navia

**Fecha de presentación:**

- ❖ 04/02/2025

# Tabla de Contenido

<b>1. Descripción de la temática de la base de datos.....</b>	<b>3</b>
<b>2. Objetivo del proyecto .....</b>	<b>3</b>
<b>3. Situación Problemática .....</b>	<b>3</b>
<b>4. Diagrama Entidad-Relación .....</b>	<b>1</b>
<b>5. Listado de Tablas .....</b>	<b>1</b>
<b>6. Instrucciones sobre el Script SQL .....</b>	<b>6</b>
<b>7. Conclusión.....</b>	<b>6</b>

**Temática del Proyecto Final:**

# ***Diseño y modelado de una base de datos para una plataforma de streaming musical***

## **1. Descripción de la temática de la base de datos**

SoundWaveDB es una base de datos diseñada para la gestión de una plataforma de streaming musical. Su propósito es organizar la información de usuarios, artistas, álbumes, canciones, listas de reproducción y eventos musicales. La base de datos permite almacenar y relacionar los diferentes elementos de la plataforma de manera eficiente. Este proyecto fue desarrollado como parte del curso de SQL de CoderHouse.

## **2. Objetivo del proyecto**

El objetivo de esta base de datos es estructurar y optimizar la gestión de datos en una plataforma de streaming musical. Permite:

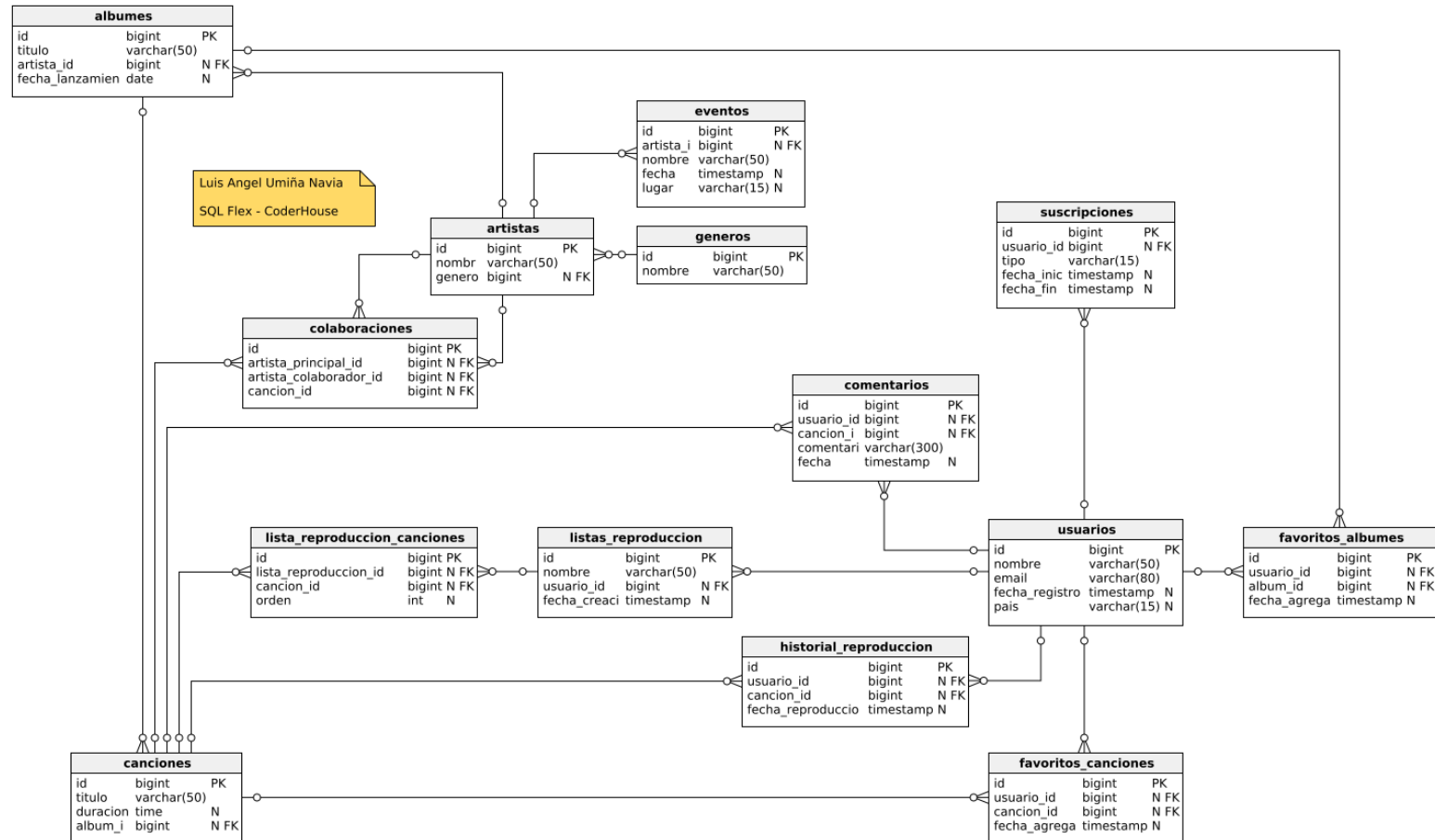
- Registrar y gestionar usuarios y sus preferencias.
- Almacenar información sobre artistas y sus géneros musicales.
- Administrar álbumes y canciones, vinculándolos con artistas.
- Gestionar listas de reproducción y su contenido.
- Registrar eventos musicales y colaboraciones entre artistas.
- Facilitar el registro de suscripciones y comentarios de los usuarios.

## **3. Situación Problemática**

En la actualidad, las plataformas de streaming musical manejan una gran cantidad de información sobre usuarios, artistas y contenido. Contar con una base de datos bien estructurada permite organizar y gestionar esta información de manera eficiente, facilitando la experiencia de los usuarios y optimizando la administración de los datos.

#### 4. Diagrama Entidad-Relación

A continuación, se detalla el diagrama Entidad-Relación:



## 5. Listado de Tablas

A continuación, se describen las tablas de nuestro modelo, indicando las claves primarias y foráneas junto con una breve descripción de cada columna.

### Tabla: usuarios

Contiene la información de los usuarios registrados en la plataforma.

<b>Campo</b>	<b>Tipo de Dato</b>	<b>Clave</b>	<b>Descripción</b>
id	BIGINT	PK	Identificador único de usuario
nombre	VARCHAR(50)		Nombre del usuario
email	VARCHAR(80)	UNIQUE	Correo electrónico
fecha_registro	TIMESTAMP		Fecha de registro en la plataforma
pais	VARCHAR(15)		País de residencia

### Tabla: artistas

Almacena los datos de los artistas que tienen contenido en la plataforma.

<b>Campo</b>	<b>Tipo de Dato</b>	<b>Clave</b>	<b>Descripción</b>
id	BIGINT	PK	Identificador único del artista
nombre	VARCHAR(50)		Nombre del artista
genero_id	BIGINT	FK	Relación con la tabla géneros

### Tabla: generos

Guarda los géneros musicales asociados a los artistas y sus canciones.

<b>Campo</b>	<b>Tipo de Dato</b>	<b>Clave</b>	<b>Descripción</b>
id	BIGINT	PK	Identificador único del género
nombre	VARCHAR(50)		Nombre del género musical

**Tabla: albumes**

Registra los álbumes publicados por los artistas en la plataforma.

<b><i>Campo</i></b>	<b><i>Tipo de Dato</i></b>	<b><i>Clave</i></b>	<b><i>Descripción</i></b>
id	BIGINT	PK	Identificador único del álbum
titulo	VARCHAR(50)		Nombre del álbum
artista_id	BIGINT	FK	Relación con la tabla artistas
fecha_lanzamiento	DATE		Fecha de lanzamiento del álbum

**Tabla: canciones**

Almacena la información de las canciones disponibles en la plataforma.

<b><i>Campo</i></b>	<b><i>Tipo de Dato</i></b>	<b><i>Clave</i></b>	<b><i>Descripción</i></b>
id	BIGINT	PK	Identificador único de la canción
titulo	VARCHAR(50)		Nombre de la canción
duracion	TIME		Duración de la canción
album_id	BIGINT	FK	Relación con la tabla albumes

**Tabla: listas\_reproduccion**

Contiene las listas de reproducción creadas por los usuarios.

<b><i>Campo</i></b>	<b><i>Tipo de Dato</i></b>	<b><i>Clave</i></b>	<b><i>Descripción</i></b>
id	BIGINT	PK	Identificador único de la lista
nombre	VARCHAR(50)		Nombre de la lista de reproducción
usuario_id	BIGINT	FK	Relación con la tabla usuarios
fecha_creacion	TIMESTAMP		Fecha de creación de la lista

**Tabla: lista\_reproduccion\_canciones**

Relaciona las listas de reproducción con las canciones que contienen.

<b><i>Campo</i></b>	<b><i>Tipo de Dato</i></b>	<b><i>Clave</i></b>	<b><i>Descripción</i></b>
id	BIGINT	PK	Identificador único
lista_reproduccion_id	BIGINT	FK	Relación con listas_reproduccion
cancion_id	BIGINT	FK	Relación con canciones
orden	INT		Posición de la canción en la lista

**Tabla: historial\_reproduccion**

Registra las canciones reproducidas por los usuarios.

<b><i>Campo</i></b>	<b><i>Tipo de Dato</i></b>	<b><i>Clave</i></b>	<b><i>Descripción</i></b>
id	BIGINT	PK	Identificador único del historial
usuario_id	BIGINT	FK	Relación con la tabla usuarios
cancion_id	BIGINT	FK	Relación con la tabla canciones
fecha_reproduccion	TIMESTAMP		Fecha y hora de reproducción

**Tabla: suscripciones**

Almacena las suscripciones de los usuarios a distintos planes de la plataforma.

<b><i>Campo</i></b>	<b><i>Tipo de Dato</i></b>	<b><i>Clave</i></b>	<b><i>Descripción</i></b>
id	BIGINT	PK	Identificador único de la suscripción
usuario_id	BIGINT	FK	Relación con la tabla usuarios
tipo	VARCHAR(15)		Tipo de suscripción
fecha_inicio	TIMESTAMP		Fecha de inicio de la suscripción

fecha_fin	TIMESTAMP		Fecha de vencimiento de la suscripción
-----------	-----------	--	--

### **Tabla: comentarios**

Permite a los usuarios comentar sobre canciones en la plataforma.

<b><i>Campo</i></b>	<b><i>Tipo de Dato</i></b>	<b><i>Clave</i></b>	<b><i>Descripción</i></b>
id	BIGINT	PK	Identificador único del comentario
usuario_id	BIGINT	FK	Relación con la tabla usuarios
cancion_id	BIGINT	FK	Relación con la tabla canciones
comentario	VARCHAR(300)		Texto del comentario
fecha	TIMESTAMP		Fecha en que se realizó el comentario

### **Tabla: colaboraciones**

Registra las colaboraciones entre artistas en una misma canción.

<b><i>Campo</i></b>	<b><i>Tipo de Dato</i></b>	<b><i>Clave</i></b>	<b><i>Descripción</i></b>
id	BIGINT	PK	Identificador único
artista_principal_id	BIGINT	FK	Relación con artistas (artista principal)
artista_colaborador_id	BIGINT	FK	Relación con artistas (colaborador)
cancion_id	BIGINT	FK	Relación con canciones



**Tabla: eventos**

Almacena eventos musicales relacionados con los artistas de la plataforma.

<b>Campo</b>	<b>Tipo de Dato</b>	<b>Clave</b>	<b>Descripción</b>
id	BIGINT	PK	Identificador único del evento
artista_id	BIGINT	FK	Relación con la tabla artistas
nombre	VARCHAR(50)		Nombre del evento
fecha	TIMESTAMP		Fecha y hora del evento
lugar	VARCHAR(15)		Ubicación del evento

**Tabla: favoritos\_canciones**

Guarda las canciones favoritas de los usuarios.

<b>Campo</b>	<b>Tipo de Dato</b>	<b>Clave</b>	<b>Descripción</b>
id	BIGINT	PK	Identificador único
usuario_id	BIGINT	FK	Relación con la tabla usuarios
cancion_id	BIGINT	FK	Relación con la tabla canciones
fecha_agregado	TIMESTAMP		Fecha en que se marcó como favorita

**Tabla: favoritos\_albumes**

Registra los álbumes favoritos de los usuarios.

<b>Campo</b>	<b>Tipo de Dato</b>	<b>Clave</b>	<b>Descripción</b>
id	BIGINT	PK	Identificador único
usuario_id	BIGINT	FK	Relación con la tabla usuarios
album_id	BIGINT	FK	Relación con la tabla albumes
fecha_agregado	TIMESTAMP		Fecha en que se marcó como favorito

## 6. Instrucciones sobre el Script SQL

El script de creación de la base de datos se encuentra en el archivo **SoundWaveDB\_CoderHouse\_DB.sql**. Se puede ejecutar en un entorno MySQL utilizando los siguientes pasos:

1. Descargar el archivo del repositorio de GitHub:  
[https://github.com/LuisUmina/SoundWaveDB\\_CH.git](https://github.com/LuisUmina/SoundWaveDB_CH.git)
2. Abrir MySQL Workbench o una herramienta compatible.
3. Ejecutar el script para crear la base de datos y sus tablas.

## 7. Conclusión

SoundWaveDB proporciona una base de datos estructurada y eficiente para una plataforma de streaming musical. Su diseño permite gestionar artistas, canciones y usuarios de manera escalable y segura. Esta primera entrega cubre la base del proyecto, y aún hay espacio para mejoras y optimización en futuras iteraciones.