

# Pre Entrega 2 - Estudio Musical

---

*Facundo Muruchi*

## 1. Introducción

El presente proyecto consiste en el diseño de una base de datos relacional para la gestión integral de un estudio musical. Este sistema tiene como objetivo facilitar la organización y el control de los distintos elementos que intervienen en el proceso de producción musical, como los artistas, álbumes, canciones, sesiones de grabación, productores, estudios, e incluso contratos. A través de esta solución se busca ofrecer una herramienta flexible, escalable y eficiente que permita centralizar y consultar la información de forma rápida y precisa.

### Objetivo

El objetivo principal de este proyecto es diseñar e implementar un modelo de base de datos que permita gestionar todas las operaciones relacionadas con la producción musical en un estudio. Entre los objetivos específicos se incluyen:

- Registrar información detallada de los artistas y productores.
- Organizar las sesiones de grabación, con su correspondiente fecha, lugar y participantes.
- Gestionar álbumes y canciones asociadas a los artistas.
- Llevar un registro de contratos, pagos y royalties.
- Posibilitar análisis posteriores como estadísticas de producción, canciones más grabadas o colaboraciones frecuentes.

Este sistema puede abarcar aspectos **contables** (pagos a artistas/productores), **logísticos** (reserva de estudios y programación de sesiones), y **analíticos** (ranking de colaboraciones, duración promedio de álbumes, etc.).

## 2. Situación Problemática

La industria musical requiere del manejo eficiente de una gran cantidad de datos provenientes de diversas fuentes: artistas, estudios, ingenieros de sonido, discográficas y más. Sin una herramienta centralizada, esta información suele dispersarse en hojas de cálculo, documentos sueltos y correos electrónicos, dificultando la trazabilidad y gestión eficiente de las producciones.

El desarrollo de esta base de datos permite solucionar varias de estas brechas, como:

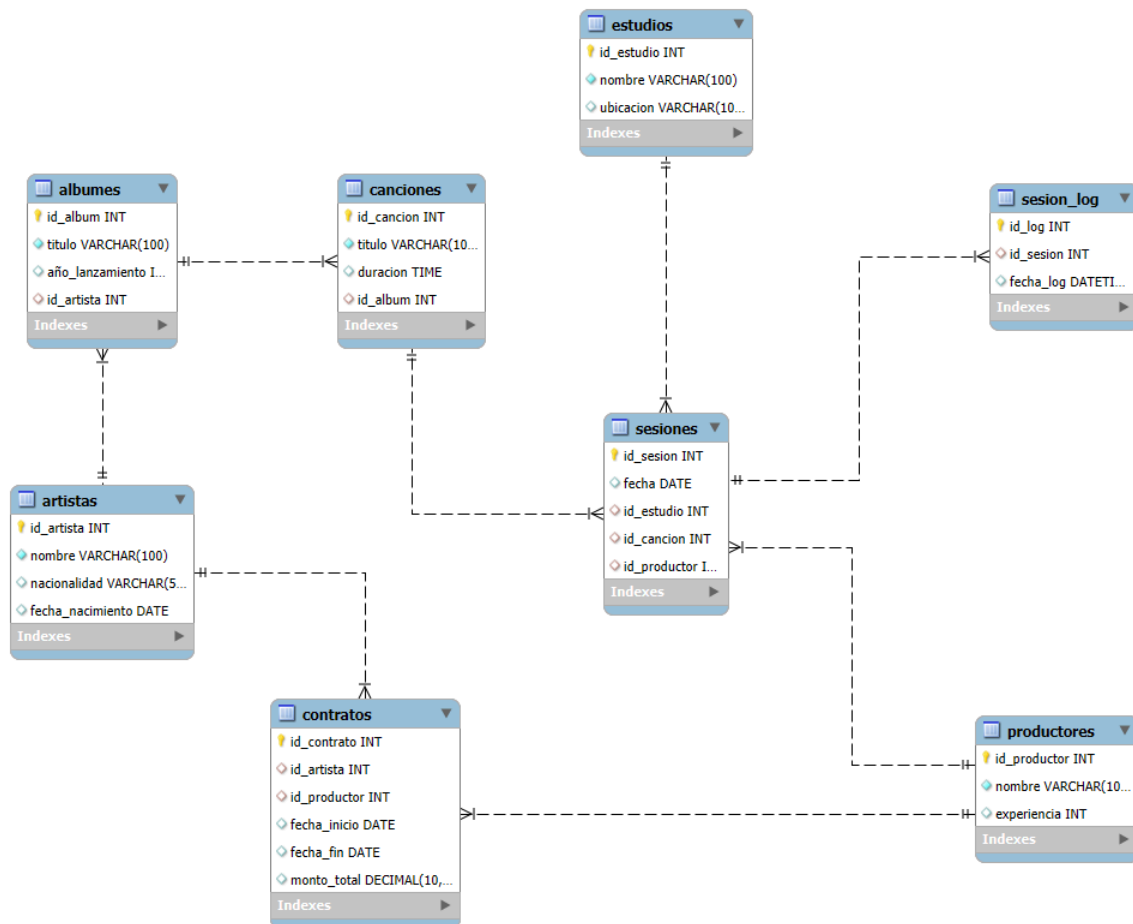
- La falta de una estructura clara para almacenar y recuperar datos de sesiones pasadas.

- Dificultades para conocer la disponibilidad de los estudios o artistas.
- Inexistencia de un registro confiable de contratos y pagos.
- Imposibilidad de obtener reportes sobre métricas clave de la producción musical.

### 3. Modelo de Negocio

La base de datos está pensada para ser utilizada por estudios de grabación musicales independientes o de mediano tamaño que trabajan con múltiples artistas. Estos estudios ofrecen sus instalaciones, equipos y profesionales para llevar a cabo sesiones de grabación, edición y mezcla de canciones. También pueden manejar contratos de producción, distribución o colaboraciones con otros artistas o empresas.

### 4. Diagrama E-R



## 5. Listado de Tablas

**Tabla: Artistas**

Campo	Tipo de Dato	Descripción
id_artista	INT (PK)	Identificador único
nombre	VARCHAR(100)	Nombre artístico
nacionalidad	VARCHAR(50)	País de origen
fecha_nacimiento	DATE	Fecha de nacimiento

**Tabla: Álbumes**

Campo	Tipo de Dato	Descripción
id_album	INT (PK)	Identificador único del álbum
titulo	VARCHAR(100)	Nombre del álbum
año_lanzamiento	INT	Año en que se publicó
id_artista	INT (FK)	Artista principal

**Tabla: Canciones**

Campo	Tipo de Dato	Descripción
id_cancion	INT (PK)	ID único
titulo	VARCHAR(100)	Nombre de la canción
duracion	TIME	Duración
id_album	INT (FK)	Álbum al que pertenece

**Tabla: Productores**

Campo	Tipo de Dato	Descripción
id_productor	INT (PK)	ID único
nombre	VARCHAR(100)	Nombre completo
experiencia	INT	Años de experiencia

**Tabla: Sesiones**

Campo	Tipo de Dato	Descripción
id_sesion	INT (PK)	ID de la sesión
fecha	DATE	Fecha de la sesión
id_estudio	INT (FK)	Estudio donde se grabó
id_cancion	INT (FK)	Canción grabada
id_productor	INT (FK)	Productor encargado

**Tabla: Estudios**

Campo	Tipo de Dato	Descripción
id_estudio	INT (PK)	ID único del estudio
nombre	VARCHAR(100)	Nombre del estudio
ubicacion	VARCHAR(100)	Dirección o ciudad

**Tabla: Contratos**

Campo	Tipo de Dato	Descripción
id_contrato	INT (PK)	ID del contrato
id_artista	INT (FK)	Artista relacionado
id_productor	INT (FK)	Productor involucrado
fecha_inicio	DATE	Inicio del contrato
fecha_fin	DATE	Fin del contrato
monto_total	DECIMAL	Valor monetario acordado

**Tabla: Sesion\_Log**

Campo	Tipo de Dato	Descripción
id_log	INT (PK)	ID del log
id_session	INT (FK)	Referencia a la sesión registrada
fecha_log	DATETIME	Fecha y hora en que se insertó el log automáticamente

## Vistas

### 1. vista\_resumen\_sesiones

**Descripción:** Muestra un resumen de cada sesión con el nombre del estudio, canción y productor.

**Tablas involucradas:** Sesiones, Estudios, Canciones, Productores.

**Uso:** Consultar rápidamente detalles de sesiones grabadas.

### 2. vista\_contratos\_activos

**Descripción:** Lista contratos cuya fecha actual está entre fecha\_inicio y fecha\_fin.

**Tablas involucradas:** Contratos, Artistas, Productores.

**Uso:** Obtener contratos vigentes.

### 3. vista\_artistas\_albumes

**Descripción:** Muestra todos los artistas junto con sus álbumes.

**Tablas involucradas:** Artistas, Álbumes.

**Uso:** Consulta simple de discografía.

---

## Funciones Personalizadas

### 1. `obtener_antiguedad_productor(id INT)`

**Descripción:** Devuelve los años de experiencia del productor.

**Tablas involucradas:** Productores.

**Uso:** Mostrar experiencia sin necesidad de SELECT completo.

### 2. `canciones_por_album(album_id INT)`

**Descripción:** Devuelve el número de canciones asociadas a un álbum.

**Tablas involucradas:** Canciones.

**Uso:** Útil para resumen de contenido por álbum.

---

## Stored Procedures

### 1. `registrar_sesion(fecha DATE, id_estudio INT, id_cancion INT, id_productor INT)`

**Descripción:** Inserta una nueva sesión de grabación.

**Tablas involucradas:** Sesiones.

**Uso:** Automatizar el alta de sesiones.

### 2. `actualizar_monto_contrato(id INT, nuevo_monto DECIMAL)`

**Descripción:** Modifica el monto de un contrato existente.

**Tablas involucradas:** Contratos.

**Uso:** Mantenimiento de información contractual.

---

## Triggers

### 1. `verificar_fechas_contrato`

**Descripción:** Previene insertar un contrato cuya fecha\_inicio sea posterior a la fecha\_fin.

**Tablas involucradas:** Contratos.

**Situación:** Se activa en inserción.

### 2. `log_sesion_insert`

**Descripción:** Inserta log en tabla auxiliar cada vez que se registra una sesión.

**Tablas involucradas:** Sesiones, Sesion\_Log.

**Situación:** Se activa en inserción.

---

## Scripts SQL

- `estudio_musical_objetos_Muruchi.sql`: Contiene creación de vistas, funciones, procedimientos y triggers.
- `estudio_musical_test_Muruchi.sql`: Contiene inserciones adicionales para testear estos objetos.