Instalación y entorno PostgreSQL

Checklist	☑
# Día	1
Estado	Completado
Fecha	@19 de diciembre de 2025
Notas Técnicas	**EJERCICIO PRÁCTICO:** 1. ¿Qué es PostgreSQL? Es un Gestor de Bases de Datos Relacional de Código Abierto, con un enfoque Orientado a Obejtos como en el paradigma de programación, este usa un modelo Cliente/Servidor para la gestión y ejecución de las consultas. 2. ¿Qué es un **cluster** en PostgreSQL? Un **Cluster** es una colección de Bases de Datos administrada por una sola Instancia de Servidor, estas bases de datos comparten la misma configuración y recursos. 3. ¿Cuántas bases de datos puede tener un cluster? puede manejar **cientos o incluso miles de bases de datos**, ya que el límite real no es un número fijo sino el espacio en disco, la memoria disponible y la capacidad de gestionar conexiones y recursos del sistema 4. ¿Qué es un **schema** y para qué sirve? Un **Schema** es un Contenedor Lógico dentro de una Base de Datos que organiza objetos como Tablas, Vistas, Funciones,

Restricciones, Índices, Secuencias, entre otros; permitiendo que estos se organicen y agrupen por funcionalidades o aplicación, permitiendo una mejor gestión y evitando conflictos de nombres.

5. Diferencia entre:

• **Cluster:** Es una colección de bases de datos que **c**omparten la misma instancia de PostgreSQL, es decir, el mismo motor, configuración, usuarios y directorio de datos.

Un Cluster tiene:
- Varias databases
- Usuarios/Roles comunes
- Configuración común (`postgressql.conf`)

Importante recordar que un Servidor puede tener varios Clusters, pero un Cluster tiene un solo directorio de datos.

• **Database:** Es una Base de Datos individual dentro de un Cluster, la cual contiene Schemas y Objetos, y a la cual es posible conectarse para trabajar.

Puntos claves de Databases:
- No se puede consultar objetos de otra databases de forma directa
- Cada conexión en PostgreSQL es una sola Database
- Dentro de la Database están los Schemas.

• **Schema:** Es un contenedor lógico dentro de una base de datos que organiza objetos como tablas, vistas, funciones, índices y restricciones, y permite evitar conflictos de nombres y gestionar permisos.

Detalles importantes de los Schemas:
- Una Database puede tener muchos schemas
- `public` es el schema por defecto
- Los schemas ayudan en la: Organizar, Seguridad y Separación

de Dominios

En PostgreSQL, un cluster es la instancia que agrupa varias bases de datos y comparte configuración y usuarios.
Cada database es una unidad independiente dentro del cluster, y dentro de cada database los schemas organizan los objetos como tablas, vistas y funciones.

6. Cómo se relacionan procesos / memoria / almacenamiento

Procesos: Son los procesos del motor PostgreSQL en ejecución que reciben las conexiones de los clientes, interpretan las consultas SQL y coordinan el acceso a la memoria y al almacenamiento.

Memoria: Es el espacio utilizado por PostgreSQL para operaciones temporales como caché de datos, buffers, ordenamientos, joins y ejecución de consultas. No almacena los datos de forma permanente.

Almacenamiento: Es el medio físico (disco) donde PostgreSQL guarda de forma persistente los datos de las bases de datos, índices y archivos de control del cluster.

* 🧠 **Proceso** → “el que piensa y ejecuta”
* ⚡ **Memoria** → “el espacio de trabajo rápido”
* 💾 **Almacenamiento** → “donde viven los datos de verdad”

En PostgreSQL, los procesos ejecutan las consultas, la memoria se usa como área de trabajo y caché, y el almacenamiento es donde los datos se guardan de forma persistente.

7. Diferencia principal entre **MySQL** y **PostgreSQL** que notes hasta ahora

PostgreSQL:
• Sistema de base de datos relacional **orientado a estándares**

• Soporte avanzado para **consultas complejas**
• Excelente manejo de **JOINs, CTEs y Window Functions**
• Soporte nativo para **datos geoespaciales** (PostGIS)
• Sistema extensible mediante **extensiones**
• Mejor manejo de **concurrencia e integridad**
• Ideal para **sistemas grandes y críticos**
• Licencia BSD (más permisiva)
• En PostgreSQL: **Database ≠ Schema** (los schemas viven dentro de la database)

MySQL:
• Sistema de base de datos relacional **enfocado en simplicidad y velocidad**
• Muy usado en aplicaciones web tradicionales
• Excelente rendimiento en **lecturas simples**
• Más fácil de administrar para escenarios básicos
• Licencia GNU GPL
• En MySQL: **Database ≈ Schema**
• Ideal para **aplicaciones CRUD sencillas**

MySQL está más orientado a simplicidad y velocidad en escenarios simples, mientras que PostgreSQL prioriza robustez, estándares SQL y consultas complejas, lo que lo hace ideal para sistemas grandes y críticos.

Licencias: MySQL vs PostgreSQL

🟠 **GNU GPL (MySQL)**
• Permite usar y modificar el software libremente.
• Si el software se **distribuye**, obliga a **liberar el código fuente** bajo la misma licencia.
• Puede ser limitante para proyectos **comerciales o propietarios**.
• Muchas empresas optan por licencias comerciales o servicios gestionados.

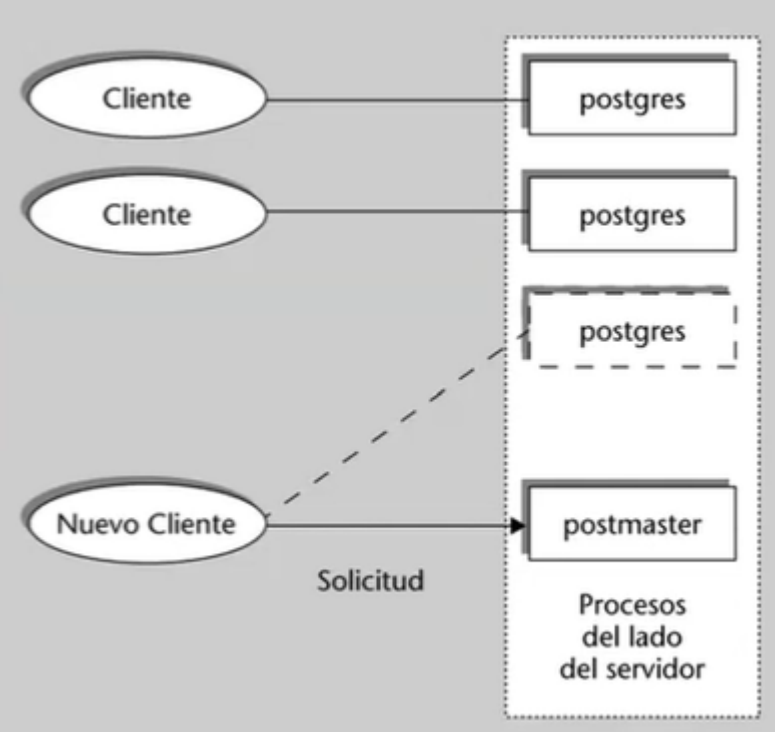
🟢 **BSD (PostgreSQL)**
• Licencia **muy permisiva**.

	• Permite usar, modificar y distribuir PostgreSQL **sin obligación de liberar el código**. • Ideal para **empresas, productos comerciales y entornos enterprise**. • Facilita adopción en grandes plataformas (AWS, GCP, Azure). 🧠 **Idea clave** • **MySQL (GPL)** → libertad con obligaciones legales. • **PostgreSQL (BSD)** → libertad casi total.
= Tiempo Invertido	5 horas

Primeros pasos en PostgreSQL

Arquitectura de PostgreSQL:

- **Cliente** es quién hace la solicitud
- **Postgres** es el Servidor que se encarga de procesar la solicitud hecha por el cliente
- **Postmaster** es el encargado de administrar el flujo de solicitudes, este se encarga de crea un **backend process** por conexión, ósea para cada **Nuevo Cliente** que va ingresando con su solicitud



Servidor / PC

```
│
├── Instancia PostgreSQL (proceso en ejecución)
│     │
│     └── Cluster
│           ├── Database 1
│           ├── Database 2
│           └── Database N
│                 │
│                 └── Schemas
│                       ├── Tablas
│                       ├── Vistas
│                       └── Funciones
```

Componentes principales:

- **Postmaster Process:** Gestiona las conexiones de los clientes
- **Backend Process:** Gestiona las consultas y ejecuta las consultas

- **Shared Memory:** La Memoria Compartida se usa para el almacenamiento en Caché y comunicación entre procesos
- PostgreSQL usa un único motor de almacenamiento, a diferencia de MySQL que soporta varios engines.
- **WAL (Write - Ahead Log):** Registro de escritura anticipado que garantiza la durabilidad de los datos

Ciclo de Vida del Proceso:

1. Conexión de la Solicitud
2. Bifurcación de Proceso
3. Autenticación
4. Ejecución de la Consulta
5. Cierre de la Conexión

🔁 Flujo real de una consulta SQL

Cuando haces un `SELECT`:

1️⃣ El **proceso** recibe la consulta

2️⃣ Busca los datos en **memoria** (cache)

3️⃣ Si no están, los lee del **almacenamiento**

4️⃣ Procesa el resultado en memoria

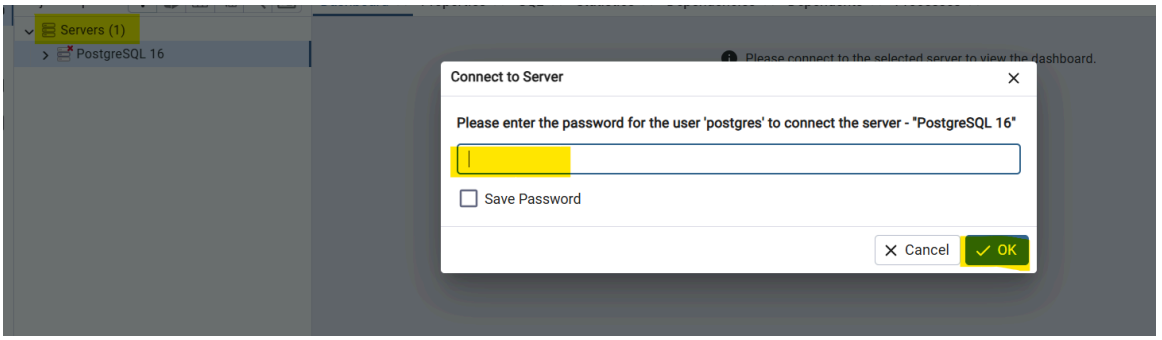
5️⃣ Devuelve la respuesta al cliente

Entorno PostgreSQL

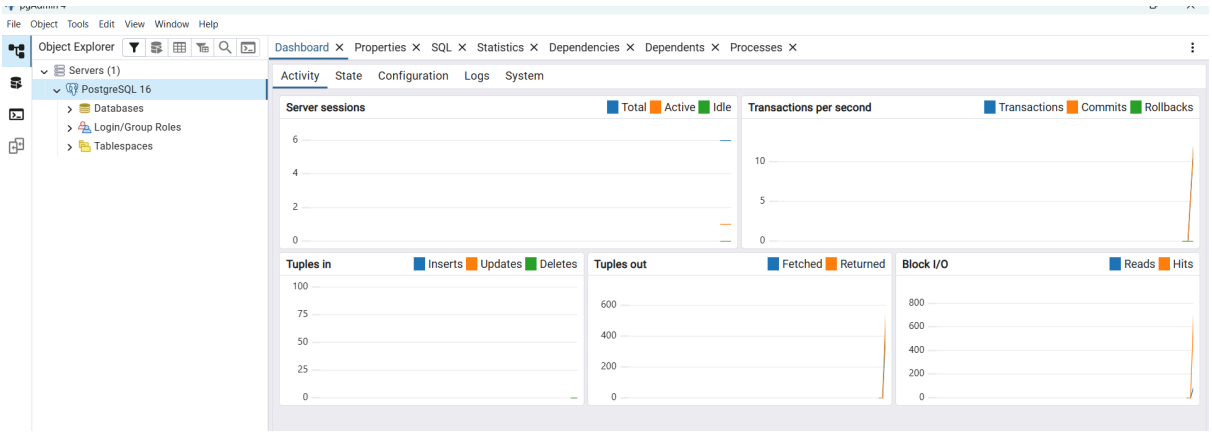
Conexión a PostgreSQL:

1. Abrir la aplicación **pgAdmin4** que es el entorno virtual que me permite trabajar con PostgreSQL

2. Dar click en la parte superior izquierda que dice **Servers** e ingresar la clave que se determinó en el momento de la instalación de PostgreSQL

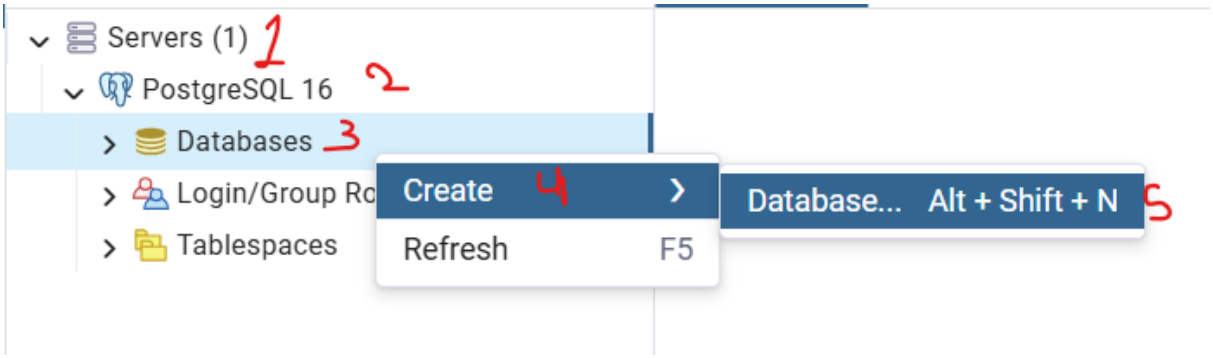


3. Conexión realizada:



Creación de una Base de Datos:

Opción 1: Desde la interfaz gráfica, dando click derecho en el icono de Databases → Create

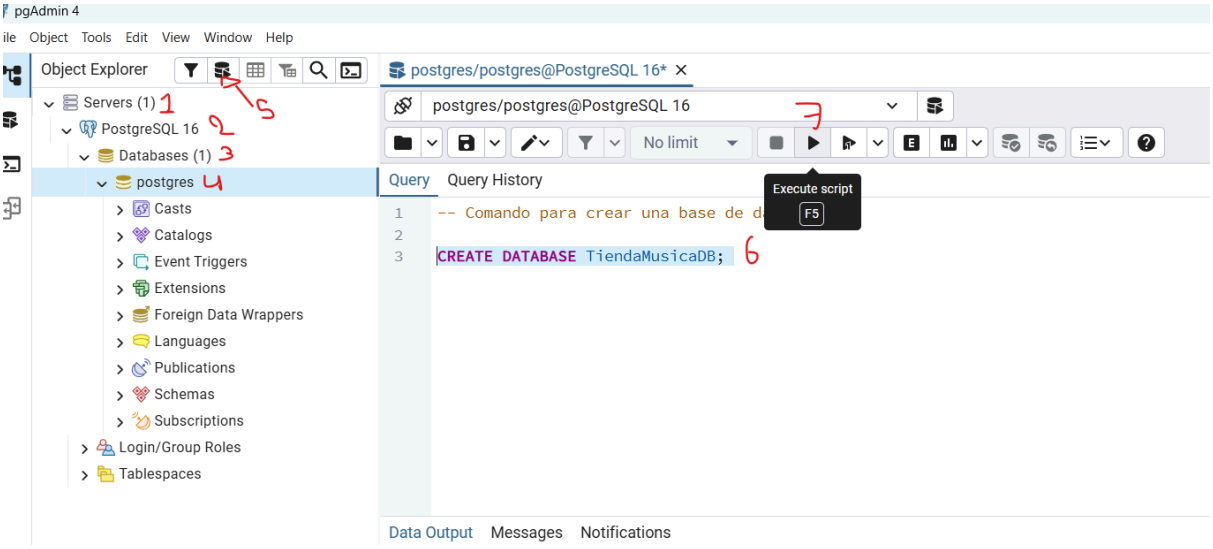


Opción 2: A través de comandos SQL

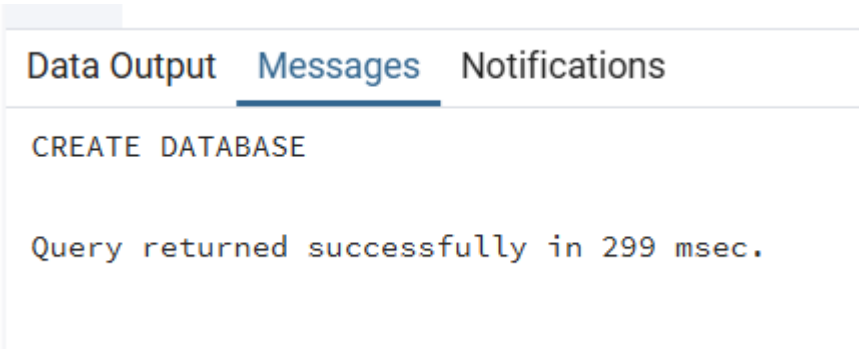
1. Dar click en **Servers**
2. Dar click en **PostgreSQL16**
3. Dar click en **Databases**
4. Seleccionar una base de datos ya existente, por ejemplo la que viene que es **Databases**
5. Dar click en el botón superior de **Query Tool**
6. Ingresar el código SQL a la terminal que se abre

- ```
  -- Crear la base de datos
  CREATE DATABASE TiendaMusicaDB;
  ```

7. Correr el código, para esto se debe seleccionar el fragmento del código y dar click en el símbolo de play ▶️
  ```

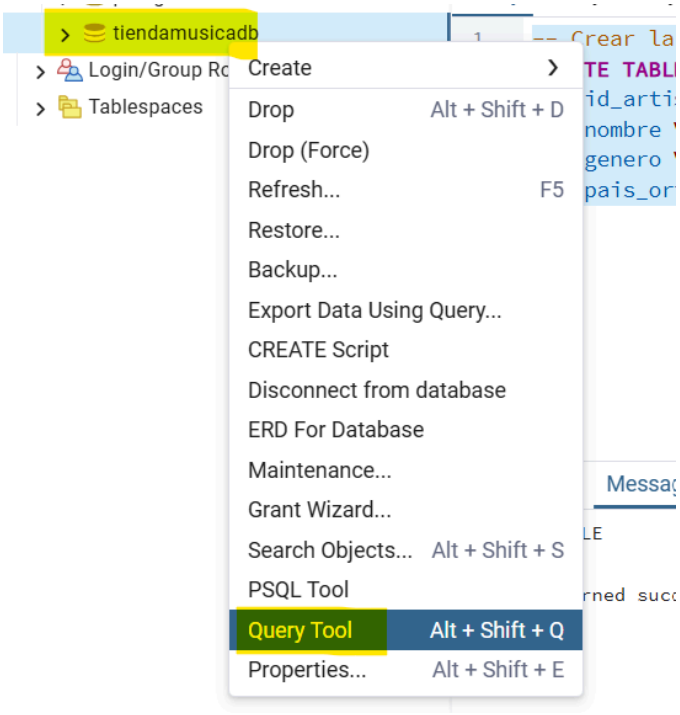


Nota: Debe aparecer un mensaje donde se informa que la base de datos fue creada con suceso



Creación de una Tabla:

Nota: Es importante antes que todo verificar que se esté trabajando sobre la Base de Datos correcta, para que la table se cree en la Base de Datos que se desea, para confirmar esto, se debe dar click derecho sobre la Base de Datos y seleccionar la opción de **Query Tool**



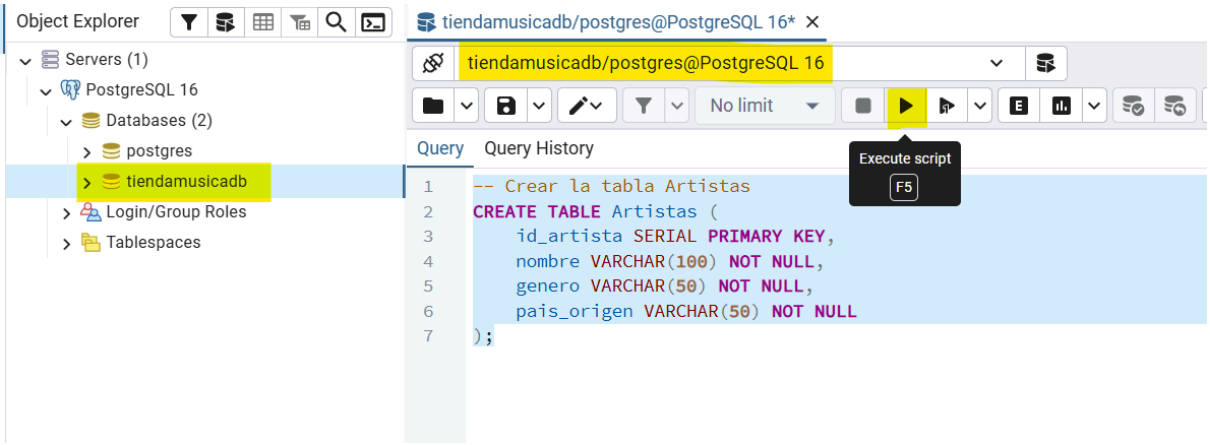
1. Ingresar la sentencia de creación de tablas en la terminal,

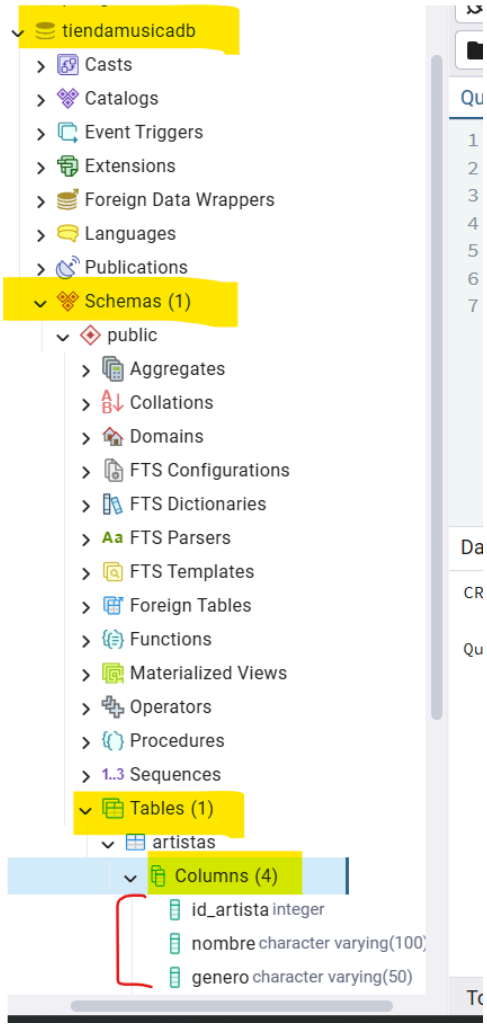
- `-- Crear la tabla Artistas`
`CREATE TABLE nombreTabla(`
`nombreColumna tipoDato propiedadesDato);`

Ejemplo de tabla:

- `CREATE TABLE Artistas (`
`id_artista SERIAL PRIMARY KEY,`
`nombre VARCHAR(100) NOT NULL,`
`genero VARCHAR(50) NOT NULL,`
`pais_origen VARCHAR(50) NOT NULL`
`);`

2. Seleccionar la sentencia de creación de la tabla y dar click en el botón de **Play** ▶️
3. Verificar que la tabla se creó correctamente





4. Ingresar valores en la Tabla

- `-- Insertar datos en la tabla Artistas`
  `INSERT INTO nombreTabla (nombreColumna1, nombreColumna2, nombreColumna3) VALUES ('valorColumna1', 'valorColumna2', 'valorColumna3');`

Ejemplo de inserción de información:

- ```
  -- Insertar datos en la tabla Artistas
  INSERT INTO Artistas (nombre, genero, pais_origen) VALUES
  ('The Beatles', 'Rock', 'Reino Unido'),
  ('Beyoncé', 'Pop', 'Estados Unidos'),
  ('Shakira', 'Pop', 'Colombia'),
  ('Adele', 'Soul', 'Reino Unido'),
  ('Ed Sheeran', 'Pop', 'Reino Unido'),
  ('Drake', 'Hip-Hop', 'Canadá'),
  ('Rihanna', 'Pop', 'Barbados'),
  ('Coldplay', 'Rock', 'Reino Unido'),
  ('Bruno Mars', 'Pop', 'Estados Unidos'),
  ('Taylor Swift', 'Pop', 'Estados Unidos'),
  ('Bad Bunny', 'Reguetón', 'Puerto Rico'),
  ('Billie Eilish', 'Pop', 'Estados Unidos'),
  ('J Balvin', 'Reguetón', 'Colombia'),
  ('Luis Miguel', 'Pop', 'México'),
  ('Madonna', 'Pop', 'Estados Unidos'),
  ('Elton John', 'Rock', 'Reino Unido'),
  ('Ariana Grande', 'Pop', 'Estados Unidos'),
  ('Maroon 5', 'Pop', 'Estados Unidos'),
  ('Selena Gomez', 'Pop', 'Estados Unidos'),
  ('Queen', 'Rock', 'Reino Unido');
  ```

Checklist:

- ✅ ~~Estudio del tema~~
- ✅ ~~Ejecución de scripts~~
- ✅ ~~Evidencia guardada~~
- ✅ ~~Notas técnicas escritas~~

Tareas del día

- ✅ ~~Instalar Postgresql~~
- ✅ ~~Verificar instalación~~
- ✅ ~~Crear mi base de laboratorio~~
- ✅ ~~Explorar el entorno~~

Recursos de estudio:
  ```

- https://www.postgresql.org/docs/current/tutorial-start.html
- https://youtu.be/MTXs5Igyn0I
- https://youtu.be/M3MG_3cRPJo
- https://youtu.be/w4sYxQ76PVU?si=T0Jwoihc9Xda152K
- https://youtu.be/n6JB4yxZnM8?si=uGu26bAU7xA04pNI