

# Manual de ejecución

El objetivo principal de este manual es generar los pasos necesarios para que el usuario pueda ejecutar el proyecto sin problema alguno.

Requisitos:

Para poder ejecutar este proyecto necesitamos las siguientes herramientas instaladas en nuestro equipo de cómputo.

- Python
- Visual Studio code
- DBeaver
- Docker

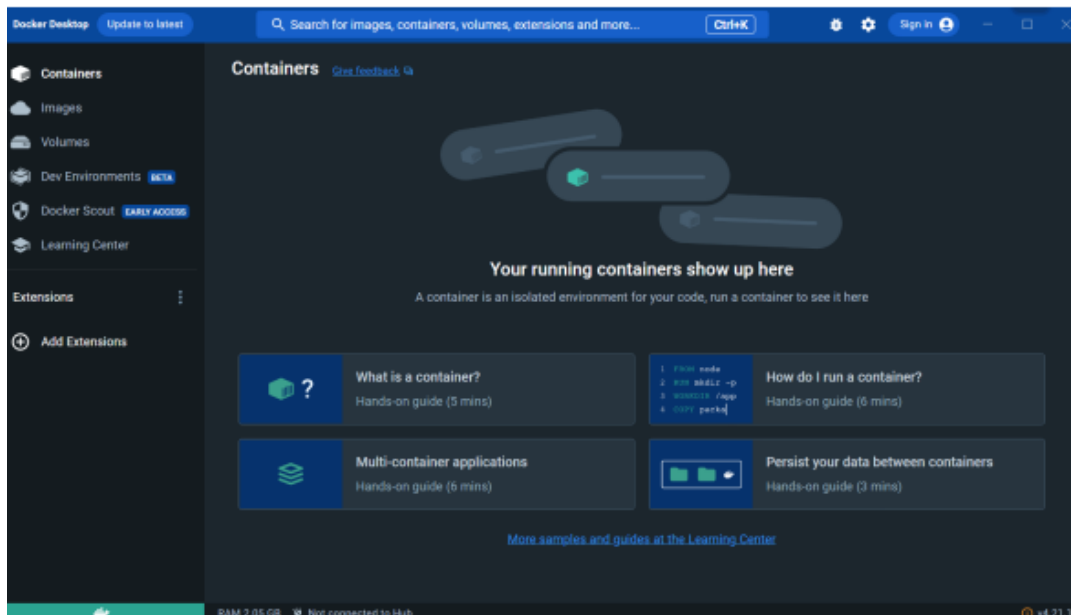
Pasos de ejecución

El primer paso para poder ejecutar este proyecto es descargar las imágenes, para poder usar PostgreSQL desde Docker, por lo que abrimos el CMD y ejecutamos las siguientes instrucciones (Nota esto debemos tener instalado Docker).

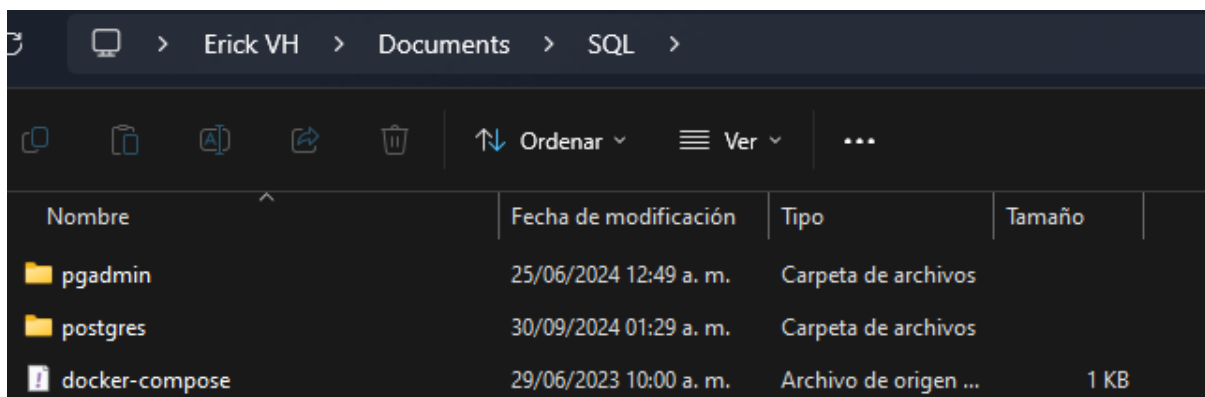
- `docker pull postgres:15.3`
- `docker pull dpage/pgadmin4`

Una vez que terminen de descargar las imágenes, el siguiente paso es crear una carpeta en donde guardaremos todo el proyecto. Una vez dentro de esa carpeta, crearemos una nueva carpeta y depositaremos el archivo `docker-compose.yml` el cual contiene toda la configuración para poder usar PostgreSQL

El siguiente paso es abrir Docker, una vez abierto se ve de la siguiente manera



A continuación, crearemos una carpeta y colocaremos nuestro Docker-compose.yml



Una vez hecho esto abrimos el cmd y nos situamos en la ruta de nuestra carpeta previamente creada y ejecutamos el siguiente comando `docker compose up -d`, esto nos levantara el servicio de postgresQL. Es importante aclarar que mientras estemos realizando la ingesta de la informacion Docker siempre tiene que permanecer abierto para que funcione.

Nota: cuando terminemos de usar docker, para darlo de baja tendremos que usar el comando `docker compose down`

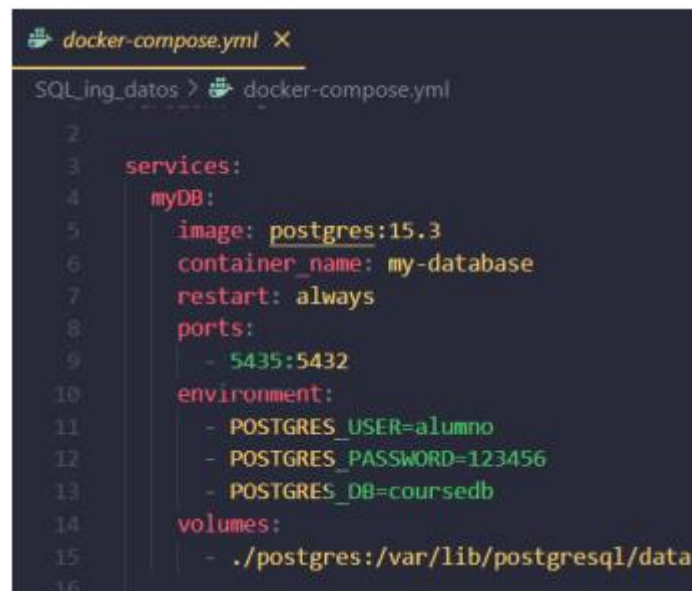
Continuando la ejecución del proyecto, abriremos visual studio code y crearemos entorno virtual dentro de nuestra carpeta donde estarán los scripts

Nota: En nuestro caso usamos virtualenv para crear este entorno virtual para instalarlo necesitamos como primer paso tener Python instalado, enseguida ejecutar el comando en la termina que nos brinda visual studio, `pip install virtualenv` una vez instalada esta librería ejecutaremos nuevamente en la terminal el comando `virtualenv env` con esto se genera nuestro entorno virtual

Para activar el entorno virtual, se ejecuta el comando `.\env\Scripts\activate` esto nos activara nuestro entorno virtual a continuación ejecutaremos el siguiente comando para instalar las librerías que se usaran en el proyecto `pip install -r requirements.txt` para poder ejecutar esto con éxito el archivo `requirements` debe de estar dentro de la carpeta principal.

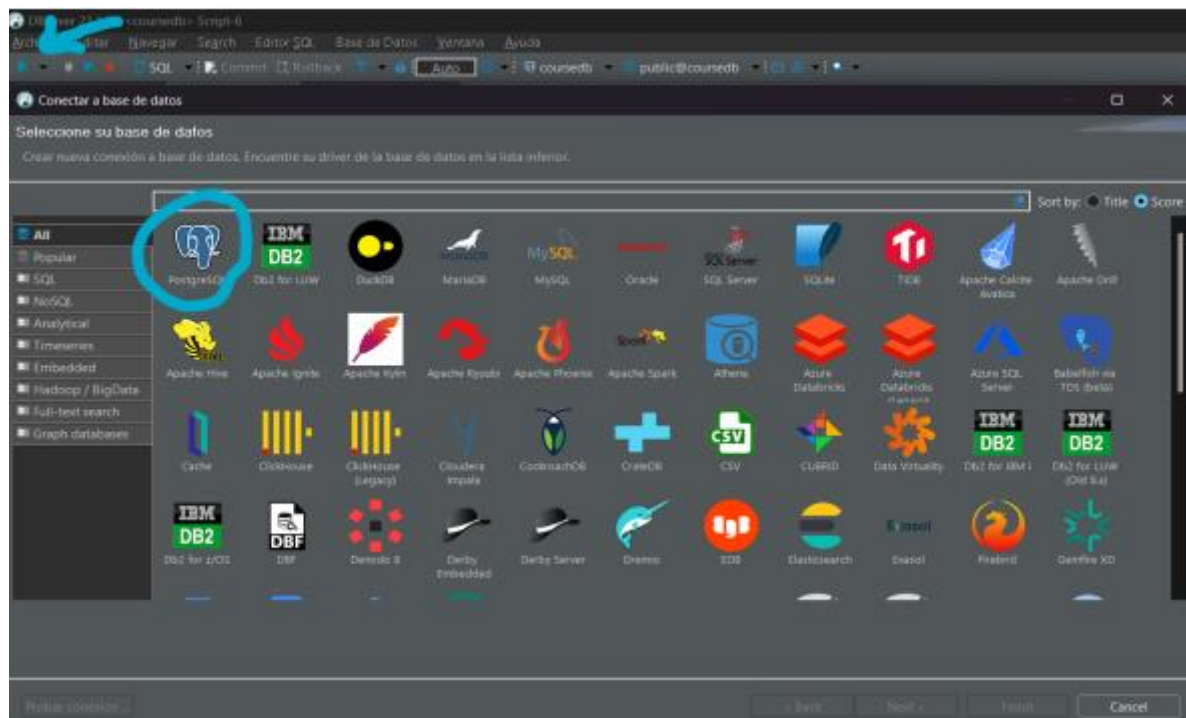
Después de esto, creamos una carpeta llamada donde colocaremos los scripts (de igual forma dentro de nuestra carpeta principal)

Antes de continuar tenemos que crear las tablas donde se ingestaran los datos, por lo que abriremos DBeaver y realizaremos la conexión para poder crear las tablas de interés (así mismo encontraras el archivo para crear las tablas en el repositorio). Para establecer la conexión, debemos de abrir el archivo `docker-compose.yml` y ubicaremos la siguiente información

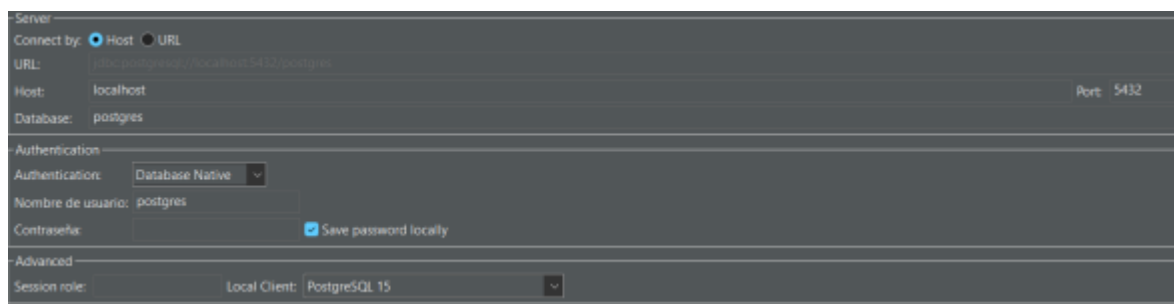


```
SQL_ing_datos > docker-compose.yml
2
3 services:
4   myDB:
5     image: postgres:15.3
6     container_name: my-database
7     restart: always
8     ports:
9       - 5435:5432
10    environment:
11      - POSTGRES_USER=alumno
12      - POSTGRES_PASSWORD=123456
13      - POSTGRES_DB=coursedb
14    volumes:
15      - ./postgres:/var/lib/postgresql/data
16
```

**ADVERTENCIA:** En cualquier otra situación estos parámetros son estrictamente confidenciales ya que pueden comprometer los datos. Este paso solo se realiza por dos razones por esta ocasión son parámetros que no comprometen de mucho la información y para la facilitar la ejecución del proyecto. Para establecer la conexión, hacemos click donde está la flecha y escogemos en este caso a PostgreSQL.



A continuación, aparece el cuadro para ingresar los datos, previamente presentados. Nota: Host se mantiene como localhost, los otros parámetros database, port, nombre usuario y contraseña se cambian por los valores presentados previamente.



Después creamos un nuevo scrip y colocamos las queries para crear las tablas (recordando estos queries se encuentran dentro de los archivos del repositorio).

Una vez creadas estas tablas regresamos a visual studio code y ejecutamos el script main.py para comenzar la ingesta de la información, este proceso dura alrededor de 35 min en cargar ambos archivos.

Una vez cargada la información podemos ir a nuestro DBeaver y ejecutar una consulta a cualquier tabla y veremos información dentro de las tablas.