

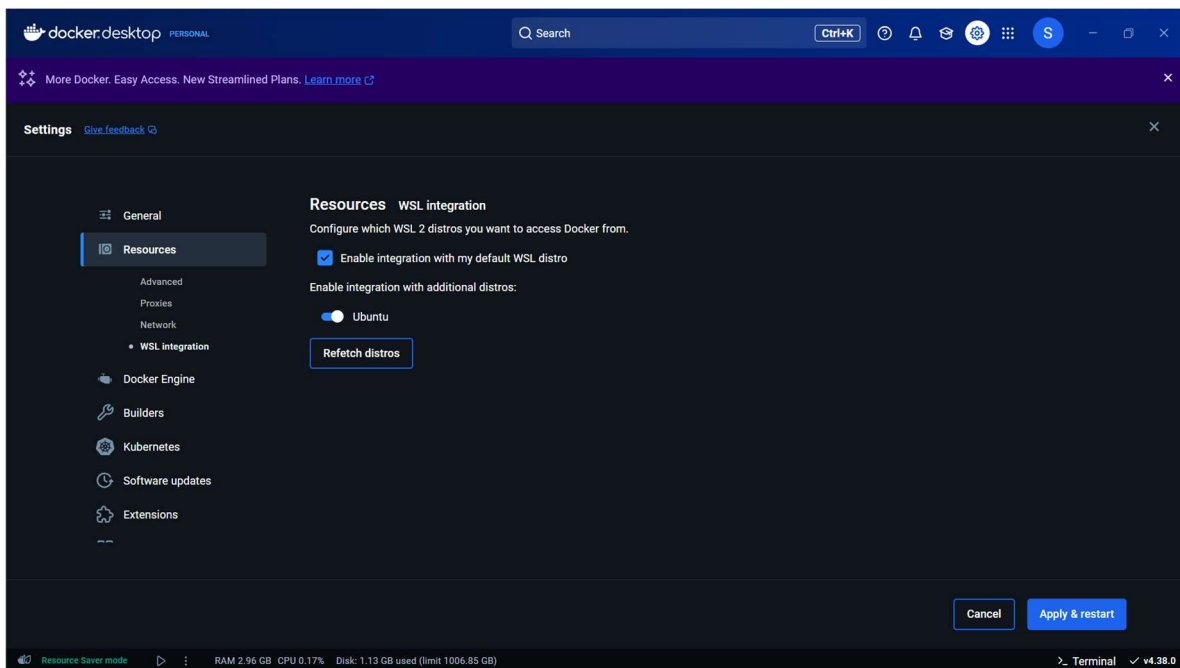
El primer método de abastecimiento que estamos usando (Samuel Uribe) es Docker, ya que, vemos grandes ventajas en él que nos facilitan el trabajo, como por ejemplo que nos permiten separar la aplicación para repararla o actualizarla sin necesidad deshabilitarla toda. A continuación, dejaremos un paso a paso que tuvimos que seguir para poder usar esta técnica de abastecimiento.

```
PS C:\Users\USUARIO> wsl --install
Descargando: Ubuntu
```

Luego después de revisar que la instalación de wsl se haya hecho correctamente, pasamos a descargar el Docker desktop desde la página oficial de Docker.

<https://www.docker.com/products/docker-desktop/>

una vez instalado el Docker desktop e iniciar sesión con mi cuenta de github (samueluribe27) revisamos su configuración y habilitamos la integración de wsl desde el Docker desktop.



La imagen que escogimos es la de postgres, la principal razón por la que decidimos escoger esta herramienta es porque ya hemos tenido manejo con la misma y nos parece práctica y sencilla.

Copiamos el Docker pull de postgres y lo pegamos en la cmd de Windows para la instalación de la imagen:

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22631.4890]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\USUARIO>docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
1c47da8f7970: Download complete
3d0b32ee6c43: Download complete
6565e7451acf: Download complete
5d821824937c: Download complete
3e0e9cc584ce: Download complete
34047ddafe46: Download complete
f62596e0b037: Download complete
5ea2d101ccee: Download complete
63b9b48ec2a9: Download complete
f69cbdd712da: Download complete
91f23f0b9c1b: Download complete
94f0cbfc26b6: Download complete
edea13d45b54: Download complete
c29f5b76f736: Download complete
Digest: sha256:6ebbefa123e24f6b02019440d2b4b922f4c039ca64f510423cddb3abf3005d7d
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
```

Luego creamos el contenedor con nombre “examen\_01\_tadb” y contraseña “samuel\_y\_joaco”.

```
C:\Users\USUARIO>docker run --name examen_01_tadb -e POSTGRES_PASSWORD=samuel_y_joaco -d -p 5432:5432 postgres:latest
28918914b67beec4b5169e727b3821f0d1c6d196d71a1761b9bf4e3bc057d39d
```

Una vez creado el contenedor, creamos un usuario con los permisos mínimos para poder desarrollar el proyecto, sin tener los permisos de un dba. Entonces para crear el usuario, primero entramos a la consola de Windows y ejecutamos el siguiente comando para entrar a la consola de postgres.

```
C:\Users\USUARIO>docker exec -it examen_01_tadb psql -U postgres
```

Una vez estamos adentro de la consola de postgres, creamos el usuario con una contraseña de nuestra elección. En mi caso le puse al usuario “samuel” y con una contraseña “Martina257\*”.

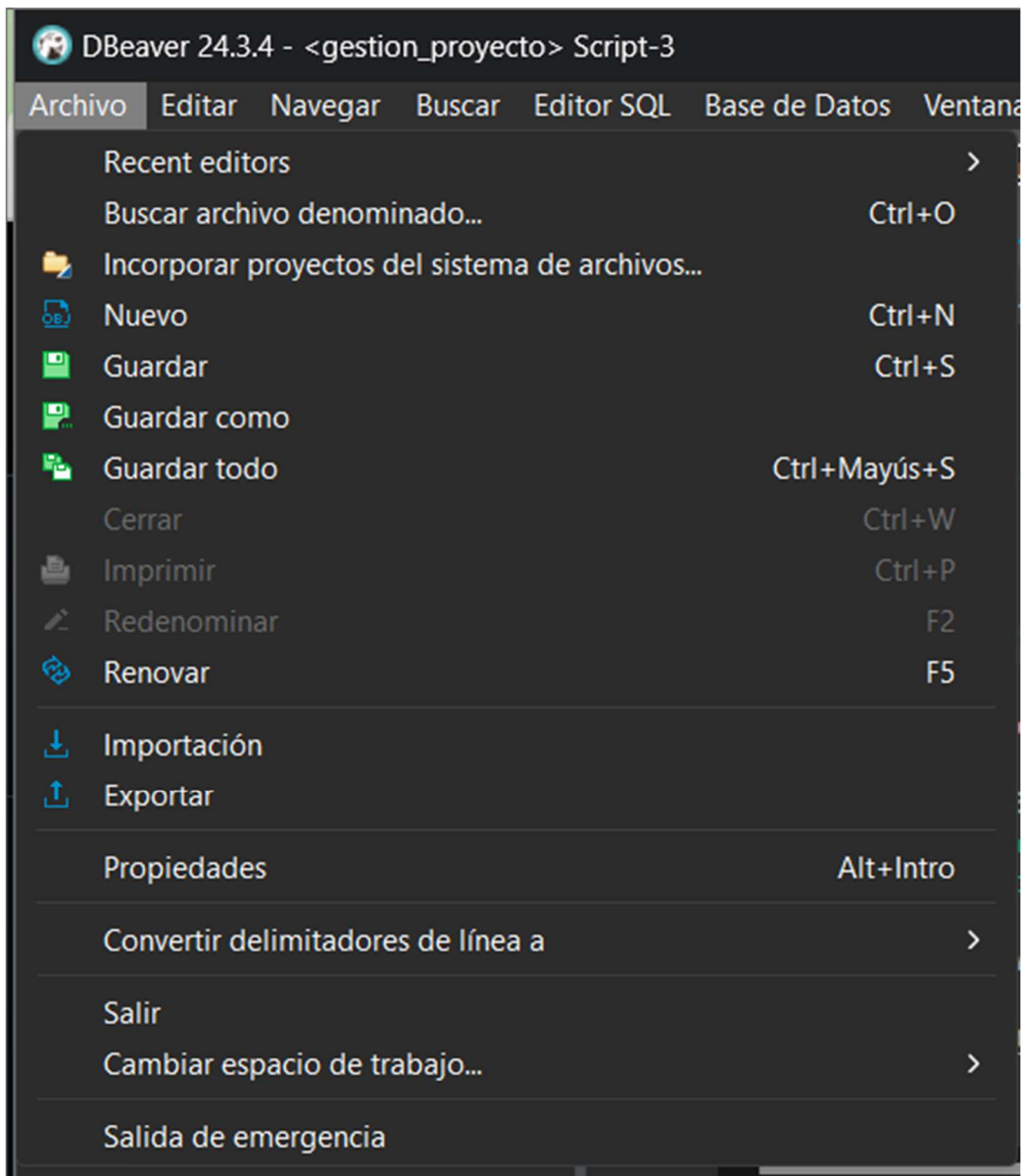
```
CREATE USER 'samuel' WITH PASSWORD 'hola';
```

Primeramente, el usuario se nos va a crear sin permisos, pero le asignaremos los permisos mínimos para que el usuario samuel pueda desarrollar sin problema pero que tampoco tenga permisos que tiene un administrador de bases de datos. Entonces en la misma bash, ponemos los siguientes comandos:

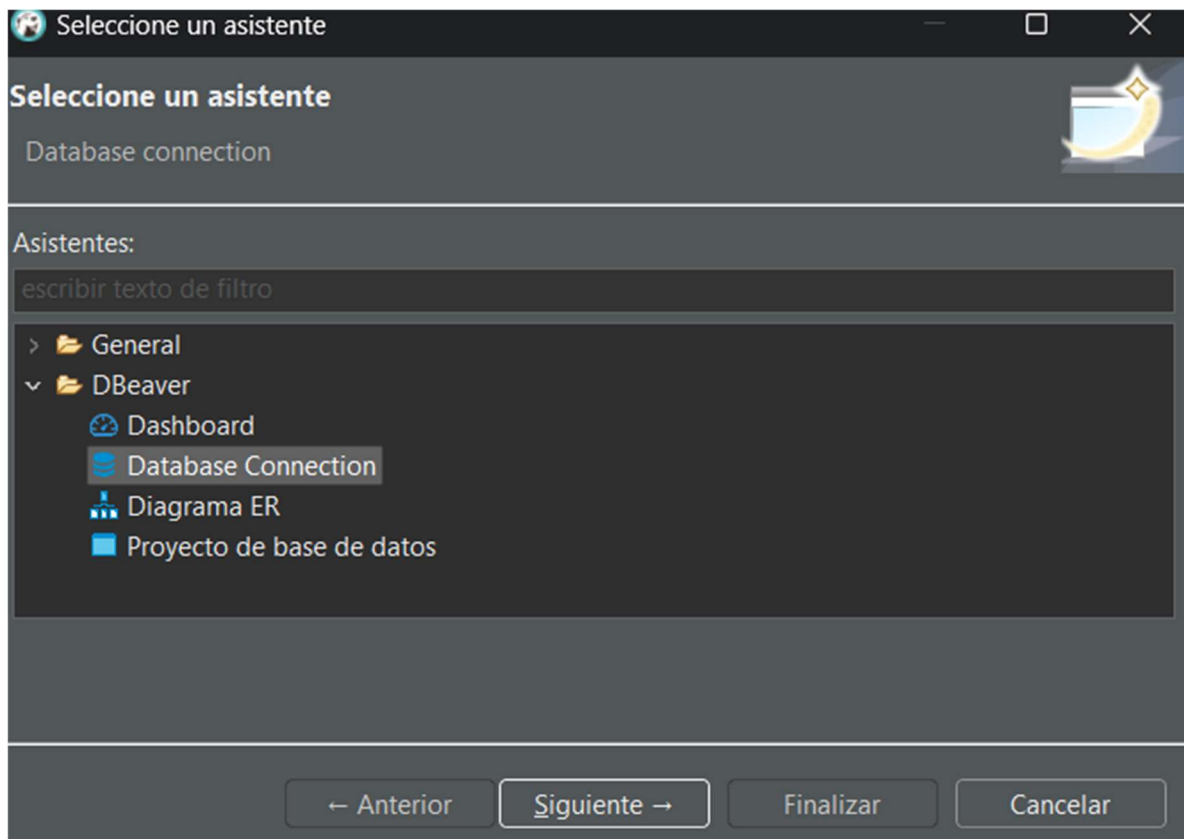
```
postgres=# GRANT CONNECT ON DATABASE postgres TO samuel;
GRANT
postgres=# CREATE DATABASE gestion_proyecto;
CREATE DATABASE
postgres=# \l
postgres=# GRANT CONNECT ON DATABASE gestion_proyecto TO samuel;
GRANT
postgres=# GRANT USAGE ON SCHEMA public TO samuel;
GRANT
postgres=# GRANT SELECT, INSERT, DELETE ON ALL TABLES IN SCHEMA public TO samuel;
GRANT
postgres=# GRANT CREATE, USAGE ON SCHEMA public TO samuel;
GRANT
postgres=# ALTER DEFAULT PRIVILEGES IN SCHEMA public
postgres=# GRANT SELECT, INSERT, UPDATE, CREATE ON TABLES TO samuel;
```

Aquí ya creamos la base de datos llamada “gestión\_proyecto”, ahora creamos tablas y todo el esquema.

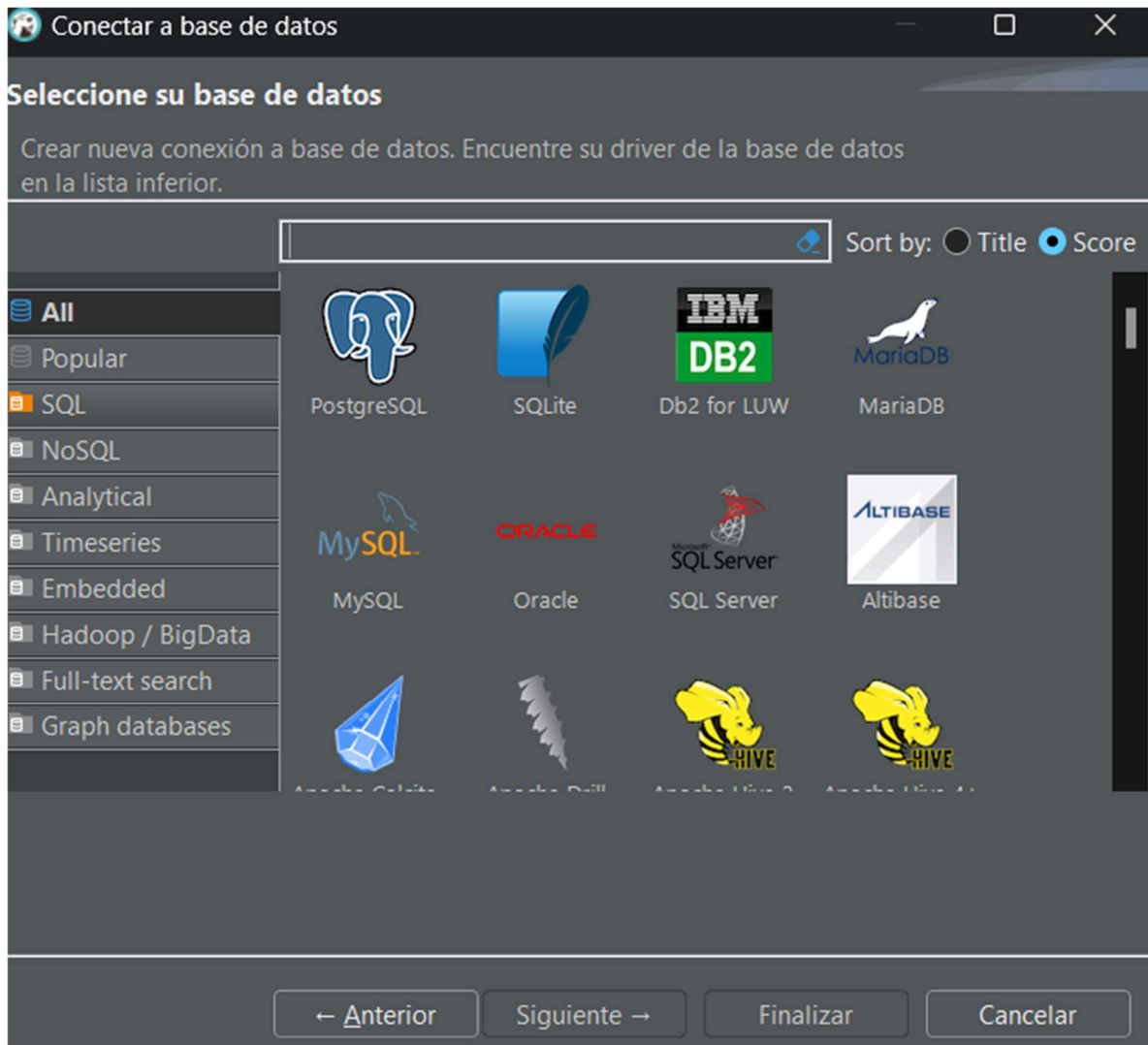
Ahora hacemos la conexión a la base de datos desde Dbeaver con el usuario, la contraseña y el nombre de la base de datos.



Aquí le damos a “nuevo”.



Aquí le damos a "Database connection".



Seleccionamos "PostgreSQL".

Conectar a base de datos

## Propiedades de Conexión

PostgreSQL ajustes de conexión

General PostgreSQL Driver properties SSH SSL + Network configurations...

URL: jdbc:postgresql://localhost:5432/gestion\_proyecto

Host: localhost Port: 5432

Database: gestion\_proyecto ☐ Show all data

Authentication

Authentication: Database Native

Nombre de usuario: samuel

Contraseña:  ☒ Save password


Advanced

Session role: Local Client: PostgreSQL 17

[Connection variables information](#) [Database documentation](#) [Connection details \(name, type, ...\)](#)

← Anterior    Siguiente →    Finalizar    Cancelar

Acá habilitamos el puerto tcp 5432 para la conexión, ponemos el nombre de la base de datos, y ponemos nuestro usuario y contraseña, por ultimo le damos a finalizar para que se cree nuestra conexión.

```
>  gestion_proyecto localhost:5432
```

Una vez tenga el chulito verde, es porque ya se estableció la conexión exitosamente.