

# INSTALAR POSTGRESQL EN DOCKER

## 1. Descargar la imagen de PostgreSQL

Ejecuta el siguiente comando para obtener la imagen oficial de PostgreSQL desde **Docker Hub**:

`docker pull postgres`

```
PS C:\Users\2-DAW> docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
Digest: sha256:073e7c8b84e2197f94c8083634640ab37105effe1bc853ca4d5fbece3219b0e8
Status: Image is up to date for postgres:latest
docker.io/library/postgres:latest
PS C:\Users\2-DAW> █
```

## 2. Crear y ejecutar un contenedor con PostgreSQL

Ejecuta el siguiente comando para iniciar un contenedor con PostgreSQL:

`docker run --name mi_postgres -e POSTGRES_USER=usuario -e POSTGRES_PASSWORD=clave -e POSTGRES_DB=midb -p 5432:5432 -d postgres`

```
PS C:\Users\2-DAW> docker run --name mi_postgres -e POSTGRES_USER=usuario -e POSTGRES_PASSWORD=clave -e POSTGRES_DB=midb -p 5432:5432 -d postgres
e519d51a3bd561e2769da29169d0aea137d989482da194232a2c27536c57e1cb
```

### Explicación de los parámetros:

- `--name mi_postgres`: Nombre del contenedor.
- `-e POSTGRES_USER=usuario`: Define el usuario administrador de la base de datos.
- `-e POSTGRES_PASSWORD=clave`: Define la contraseña para el usuario.
- `-e POSTGRES_DB=midb`: Nombre de la base de datos por defecto.
- `-p 5432:5432`: Expone el puerto 5432 del contenedor al puerto 5432 de tu máquina.
- `-d postgres`: Ejecuta el contenedor en segundo plano usando la imagen de PostgreSQL.

## 3. Verificar que el contenedor está en ejecución

Para comprobar que el contenedor está corriendo, usa:

`docker ps`

Si necesitas ver todos los contenedores, incluidos los detenidos:

`docker ps -a`

```
PS C:\Users\2-DAW> docker ps
```

CONTAINER ID	IMAGE	COMMAND	NAMES	CREATED	STATUS	PORTS
e519d51a3bd5	postgres	"docker-entrypoint.s..."	mi_postgres	4 minutes ago	Up 4 minutes	0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp

```
PS C:\Users\2-DAW> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	NAMES	CREATED	STATUS	PORTS
e519d51a3bd5	postgres	"docker-entrypoint.s..."	mi_postgres	4 minutes ago	Up 4 minutes	0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp

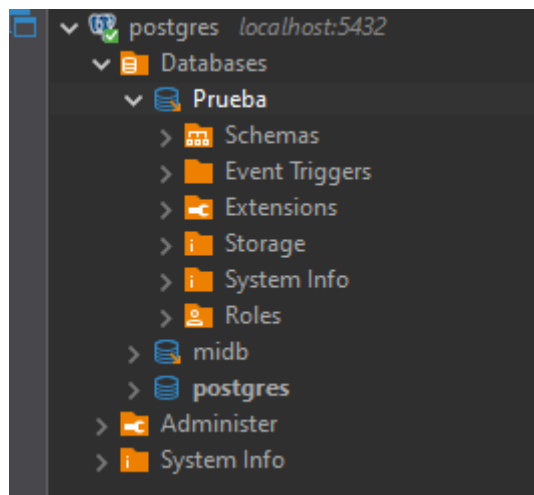
#### 4. Acceder a la base de datos desde el contenedor

Puedes acceder al shell de PostgreSQL dentro del contenedor ejecutando:

`docker exec -it mi_postgres psql -U usuario -d midb`

Esto abrirá la consola de PostgreSQL dentro del contenedor, permitiéndote ejecutar comandos SQL.

```
Terminal
Prueba=# create table pepe;
ERROR:  syntax error at or near "clear"
LINE 1: clear
        ^
Prueba=# exit
PS C:\Users\2-DAW> docker exec -it mi_postgres psql -U usuario -d Prueba
```



#### 5. Detener y eliminar el contenedor

Si quieres detener el contenedor, usa:

`docker stop mi_postgres`

Para eliminarlo completamente:

```
docker rm mi_postgres
```

```
PS C:\Users\2-DAW> docker stop mi_postgres
mi_postgres
PS C:\Users\2-DAW> docker rm mi_postgres
mi_postgres
PS C:\Users\2-DAW> docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
ollama/ollama        latest          a61a8fd395db   4 days ago     5.41GB
postgres             latest         073e7c8b84e2   3 weeks ago    643MB
stremio/server        latest         e732e0dccff2   4 weeks ago    1.72GB
mcp/postgres         latest         b72c08f19e86   4 months ago   239MB
PS C:\Users\2-DAW> docker ps
CONTAINER ID   IMAGE          COMMAND         CREATED        STATUS        PORTS          NAMES
```

## 6. Volumen persistente (Opcional)

Si deseas que los datos de la base de datos no se pierdan al eliminar el contenedor, puedes crear un volumen con:

```
docker run --name mi_postgres -e POSTGRES_USER=usuario -e POSTGRES_PASSWORD=clave
-e POSTGRES_DB=midb -p 5432:5432 -v pgdata:/var/lib/postgresql/data -d postgres
```

- `-v pgdata:/var/lib/postgresql/data`: Crea un volumen persistente llamado **pgdata**.

## 7. Conectar desde un cliente externo

Si tienes un cliente de PostgreSQL (como **DBeaver** o **pgAdmin**) en tu máquina, puedes conectarte usando:

- **Host**: localhost
- **Puerto**: 5432
- **Usuario**: usuario
- **Contraseña**: clave
- **Base de datos**: midb

Esto debería permitirte usar PostgreSQL en un contenedor Docker sin problemas.