

Bases de datos en un contexto FOSS

Una introducción práctica a PostgreSQL

Miguel Mejía J.

2026



Índice

Introducción

Pasos previos

Configurando y levantando PostgreSQL

Conectándose a PostgreSQL

Consideraciones finales

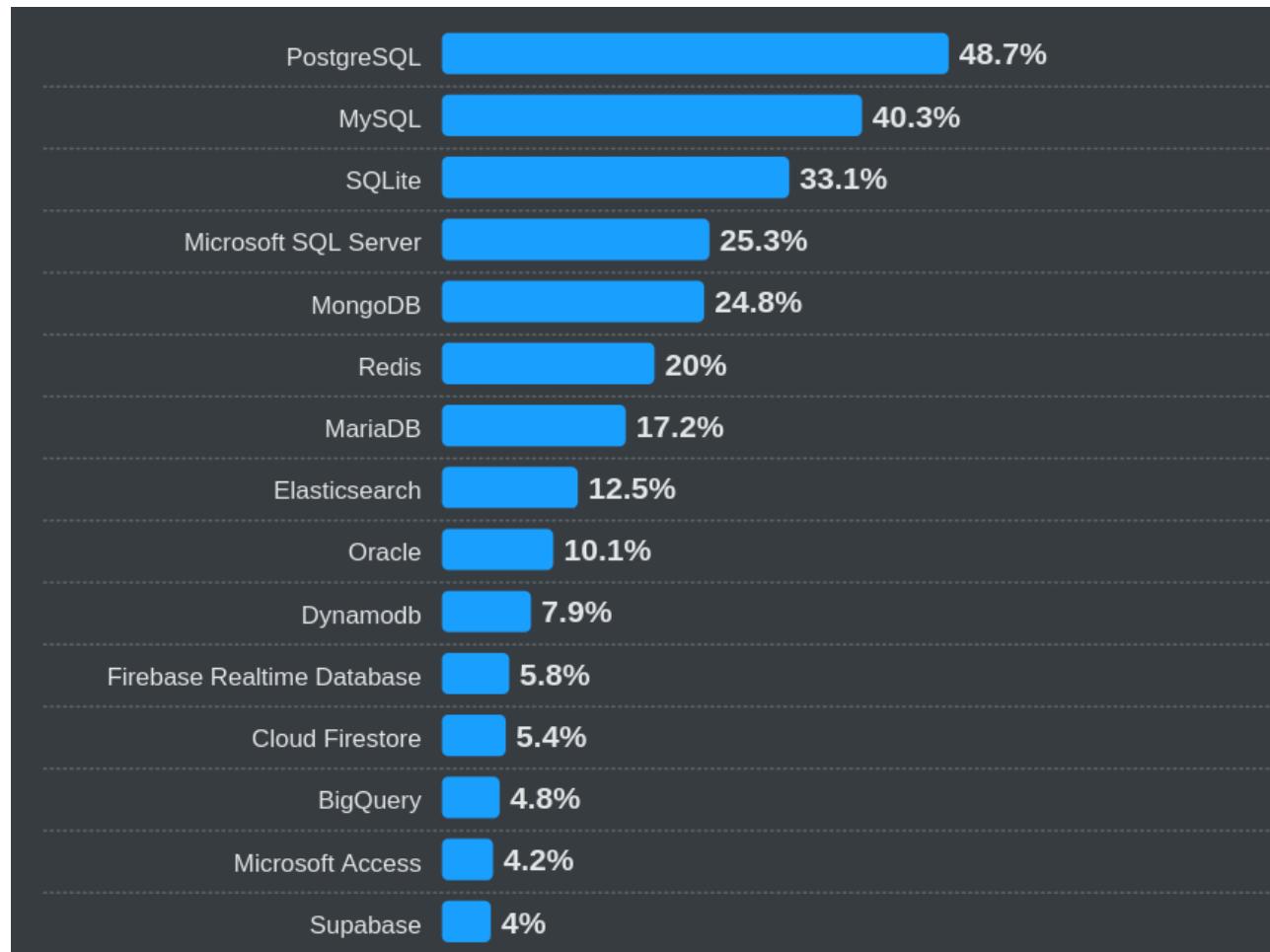
INTRODUCCIÓN

PostgreSQL debuted in the developer survey in 2018 when 33% of developers reported using it, compared with the most popular option that year: MySQL, in use by 59% of developers. Six years later, **PostgreSQL is used by 49% of developers** and is the most popular database for the second year in a row.

— [2024 Stack Overflow Developer Survey](#)



Introducción



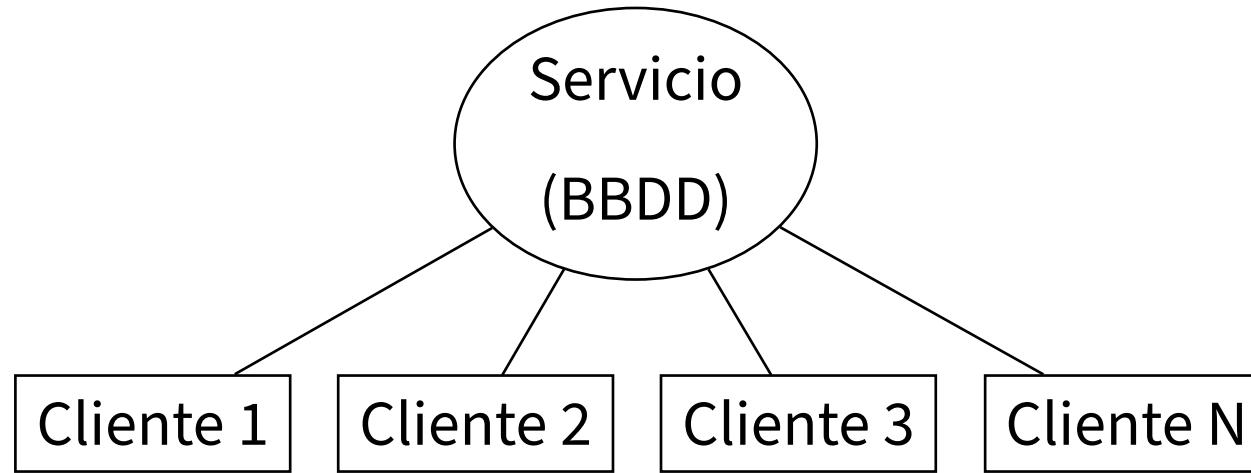
Will PostgreSQL ever be released under a different license?

The PostgreSQL Global Development Group remains committed to making PostgreSQL available as free and open source software in perpetuity. There are no plans to change the PostgreSQL License or release PostgreSQL under a different license.

Captura de pantalla de <https://www.postgresql.org/about/licence>

PASOS PREVIOS

Fundamentos



Los procesos que quieren acceder a los datos, se tienen que conectar (sockets, IP) a un proceso (un *servicio*) que atiende sus consultas.

Alcance de este taller

1. Levantar un **servidor** PostgreSQL.
2. Conectarnos a él mediante distintos **clientes**:
 - psql (CLI)
 - DBeaver (GUI).
 - Python (Aplicación propia).
3. ¡Trastear!

Descarga e instalación del software

PostgreSQL

Linux (apt)

```
$ sudo apt install postgresql
```

Linux (pacman)

```
$ sudo pacman -S postgresql
```

Notas sobre Windows

- Durante la propia instalación se configura una base de datos.
- De los componentes que ofrece el instalador, nos interesa solo:
 - ▶ PostgreSQL Server
 - ▶ Command Line Tools

Tutoriales

- <https://www.postgresql.org/download/linux> (Linux - Debian, Redhat, SUSE, Ubuntu)
- <https://www.geeksforgeeks.org/installation-guide/how-to-install-postgresql-on-arch-based-linux-distributions-manjaro> (Linux - Arch, Manjaro)
- https://www.w3schools.com/postgresql/postgresql_install.php (Windows)



DBeaver

DBeaver Community

- <https://dbeaver.io>
- Soporte para múltiples sistemas gestores de BBDD (PostgreSQL, MySQL, Oracle, MariaDB, SQLite, entre otros)
- Licencia de código abierto (Apache 2.0)

DBeaver Pro (versión cerrada)

- <https://dbeaver.com>

Python & psycopg2

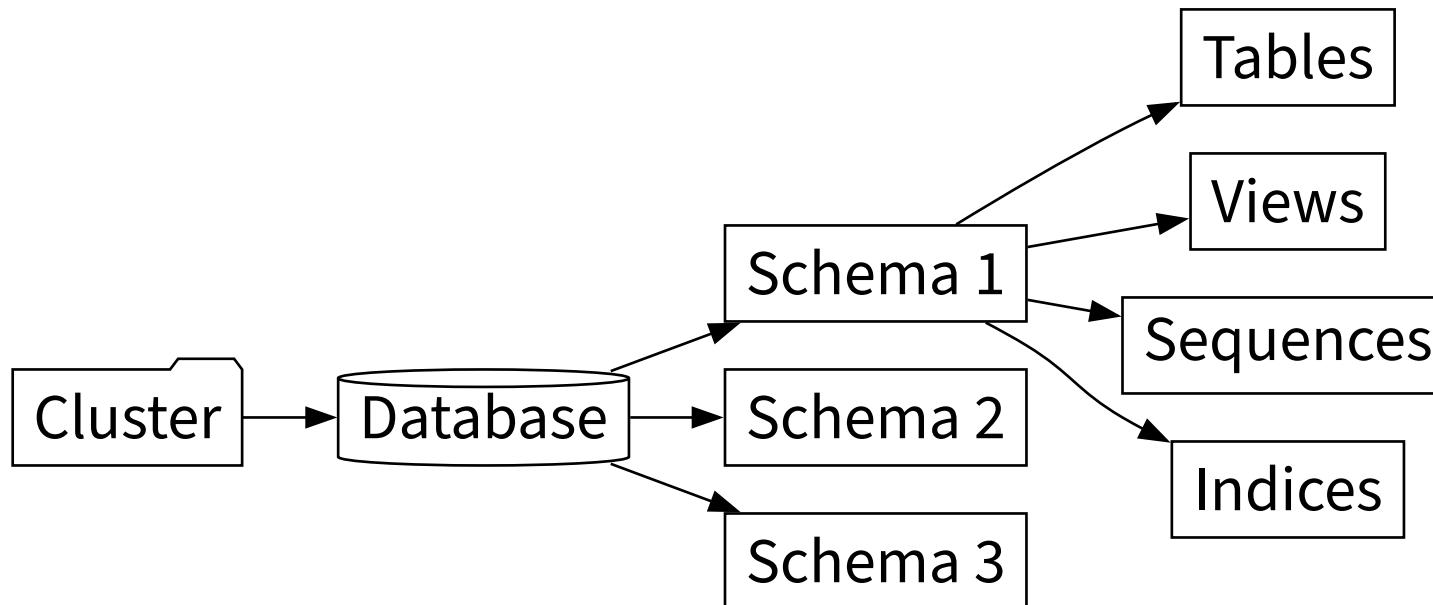
- <https://www.python.org/downloads>
- <https://pypi.org/project/psycopg2>

Otros drivers

- <https://www.postgresql.org/download/products/2-drivers-and-interfaces/>

CONFIGURANDO Y LEVANTANDO POSTGRESQL

Conceptos iniciales



- **Cluster** - A efectos de este taller, es un directorio en el sistema de ficheros (directorio de datos).
- **Base de datos** - Almacenada en un *cluster*, formada por *esquemas*.
- **Esquema** - Espacio de nombres. Permite una gestión modular.
- **Tablas** - Conjuntos de datos estructurados.

- **Usuario de BBDD** - Para autenticación y autorización.

Usuario del SO \neq Usuario de BBDD

- ▶ Se recomienda que PostgreSQL se gestione con un usuario propio en el sistema operativo, para un mejor control de acceso. En Linux, la propia instalación de PostgreSQL suele encargarse de crear uno llamado `postgres`.
- ▶ Además, el superusuario de la BBDD también se llama `postgres`.

Siguientes pasos

Iniciar el cluster

Arrancar el servicio

Cambiar contraseña por defecto de postgres

Configurar autenticación

Crear una base de datos

Iniciar el cluster

```
$ sudo -i -u postgres
$ initdb --locale $LANG -E UTF8 -D '/var/lib/postgres/data/'
```

Arrancar el servicio

```
$ sudo systemctl start postgresql
$ sudo systemctl status postgresql
```

Cambiar contraseña por defecto de postgres

Para modificarlo, nos conectamos mediante la herramienta `psql` (en Windows, SQL Shell).

```
$ sudo -u postgres psql
postgres=# ALTER USER postgres WITH PASSWORD 'my_pass';
```

Configurar autenticación

Debemos modificar el fichero de configuración correspondiente.

```
$ sudo -u postgres nano '/var/lib/postgres/data/pg_hba.conf'
```

El fichero pg_hba.conf se encuentra en la raíz del directorio de datos que indicamos al inicializar el cluster. También en el caso de Windows.

pg_hba.conf

#	TYPE	DATABASE	USER	ADDRESS	METHOD
	local	all	all		peer
	host	all	all	127.0.0.1/32	scram-sha-256
	host	all	all	::1/128	scram-sha-256

- local se refiere a las conexiones por socket de Unix (ejemplo: psql).
- host se refiere a las conexiones por IP.
- peer sólo comprueba que el usuario de SO coincide con el de BBDD (delega la autenticación al sistema operativo).
- scram-sha-256 (md5 anteriormente) exige el uso de una contraseña.

Después de guardar la nueva configuración, hay que indicarle al servicio que la recargue:

```
$ sudo systemctl reload postgresql
```

En Windows lo debemos hacer desde `psql` directamente.

```
postgres=# pg_reload_conf();
```

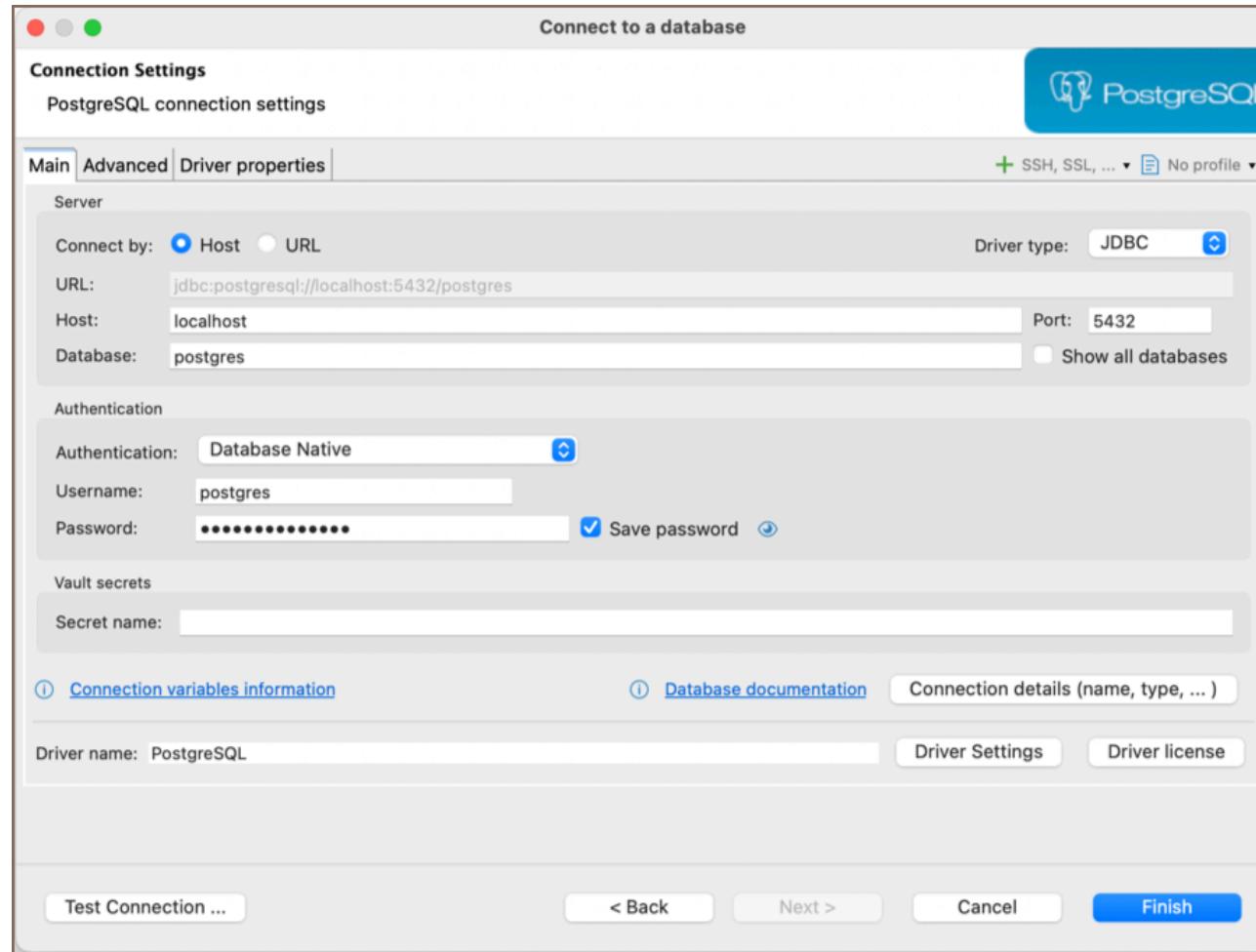
Crear una base de datos

```
$ sudo -u postgres psql
postgres=# CREATE DATABASE taller_bokeron;
```

CONECTÁNDOSE A POSTGRESQL

DBeaver

- <https://dbeaver.com/docs/dbeaver/Create-Connection>



<https://dbeaver.com/docs/dbeaver/Create-Connection/#configuring-connection-settings>

Python + psycopg2

- <https://www.geeksforgeeks.org/python/python-psycopg2-connection-class/>

Ejemplo en: code/query_version.py

CONSIDERACIONES FINALES

- Cada sistema tiene pequeñas variaciones en su **dialecto de SQL**, principalmente en las funciones nativas y los tipos. Tenedlas en cuenta.
- Para el uso en aplicaciones reales, conviene entender a fondo los conceptos de **seguridad** (usuarios, permisos, ...) y el resto de la arquitectura de la aplicación: qué otros controles de acceso existen, cómo se relacionan, etc.
- No hemos entrado en la cuestión de las **conexiones remotas**. Hacen falta configuraciones especiales en pg_hba.conf.
- Aprended **Docker** (y docker-compose). Con ello podréis levantar y tirar bases de datos en cuestión de segundos.

Bases de datos en un contexto FOSS

Una introducción práctica a PostgreSQL

Miguel Mejía J. (contacto: miguelmj.dev/social)

2026



Diapositivas hechas con Typst

