

Actividad 4 - Instalación PostgreSQL

Cristóbal Suárez Abad

Administración de Sistemas Gestores de Bases de Datos.

2º ASIR.

Con el fin de experimentar el despliegue de otros sistemas gestores de base de datos instala un nuevo SGBD, PostgreSQL:

Desarrolla un documento con el procedimiento de instalación y puesta en marcha

1. Indica la versión instalada, la capacidad mínima de almacenamiento, la memoria RAM mínima
2. Los pasos generales de instalación
3. Términos de uso y condiciones sobre el producto.
4. Una enumeración de las posibles instalaciones posibles.
5. ¿Tiene alguna interfaz gráfica para usar o extensión? Explica su utilidad

*para contraseñas pon siempre -> dh202

1. **Indica la versión instalada, la capacidad mínima de almacenamiento, la memoria RAM mínima**

Versión instalada: 14.19

Se puede comprobar una vez instalado usando "SELECT version ();" dentro de postgresql.

```
version
-----
PostgreSQL 14.19 (Ubuntu 14.19-0ubuntu0.22.04.1) on x86_64-pc-linux-gnu, compiled by gcc (Ubuntu 11.4.0-1ubuntu1~22.04.2) 11.4.0, 64-bit
(1 row)
```

O desde fuera con:

psql --version

```
[usuario@csapostgre ~]$psql --version
psql (PostgreSQL) 14.19 (Ubuntu 14.19-0ubuntu0.22.04.1)
Cristobal - domingo, 5 octubre 2025 07:46
```

Requisitos de hardware¹:

Mínimo 2 núcleos de CPU, 2GB de RAM y 10 GB de almacenamiento (preferiblemente SSD).

¹ <https://www.getgalaxy.io/learn/glossary/how-to-meet-minimum-system-requirements-in-postgresql>

2. Los pasos generales de instalación:

En Ubuntu Server²:

- Actualizamos el repositorio:

sudo apt update

- Instalamos postgresql:

sudo apt install postgresql postgresql-contrib

- Iniciamos el servicio:

sudo systemctl start postgresql

Cuando se instala, crea de manera automática un usuario llamado “postgres”. Tanto en el programa como en Linux (podemos verlo en “/etc/passw”).

```
usuario:x:1000:1000:usuario:/home/usuario:/bin/bash
lxd:x:999:100::/var/snap/lxd/common/lxd:/bin/false
postgres:x:114:120:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
Cristobal sábado 4 octubre 2025 19:45
```

Para **entrar en postgres**, debes cambiarte al usuario creado y luego ejecutar “psql”. O todo a la vez.

sudo -i -u postgres psql

```
Cristobal sábado 4 octubre 2025 19:47
[usuario@server2asir ~]$ sudo -i -u postgres
postgres@server2asir:~$ createuser --interactive
Enter name of role to add: cristobal
Shall the new role be a superuser? (y/n) y
postgres@server2asir:~$ createdb 2ASIR
```

Con el usuario de Linux “postgres” podemos crear usuarios³:

“**createuser --interactive**”

O

“**CREATE USER _____;**”

```
postgres=# CREATE USER prueba01;
CREATE ROLE
```

² <https://www.digitalocean.com/community/tutorials/how-to-install-postgresql-on-ubuntu-20-04-quickstart>

³ <https://www.postgresql.org/docs/8.0/sql-createuser.html>

Crear bases de datos:

"createdb _____"

O

"CREATE DATABASE _____;"

Desde dentro, podemos ver las bases de datos creadas.

Podemos otorgar contraseñas a los usuarios:

"ALTER USER _____ WITH PASSWORD ' _____';"

```
postgres=# ALTER USER prueba01 WITH PASSWORD '12345';
ALTER ROLE
postgres=#
```

Y también privilegios⁴⁵:

A bases de datos concretas:

GRANT CONNECT ON DATABASE nombre_basedatos TO nombre_usuario;

GRANT CREATE ON DATABASE "2ASIR" TO prueba01;

GRANT ALL PRIVILEGES ON DATABASE "2ASIR" TO prueba01;

```
postgres=# GRANT CREATE ON DATABASE "2ASIR" TO prueba01;
GRANT
postgres=# \du
postgres=# GRANT ALL PRIVILEGES ON DATABASE "2ASIR" TO prueba01;
GRANT
```

O en los esquemas: (el esquema "public" se crea durante la instalación).

GRANT ALL PRIVILEGES ON SCHEMA public TO prueba01;

GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO prueba01;

GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO prueba01;

```
2ASIR=# GRANT ALL PRIVILEGES ON SCHEMA public TO prueba01;
GRANT
2ASIR=# GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO prueba01;
GRANT
2ASIR=# GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO prueba01;
GRANT
2ASIR=#
```

- Podemos acceder directamente con un usuario a una tabla especificada:

"psql -U usuario -d "base_datos" -h localhost"

⁴ <https://stackoverflow.com/questions/22483555/postgresql-give-all-permissions-to-a-user-on-a-postgresql-database>

⁵ <https://stackoverflow.com/questions/24918367/grant-privileges-for-a-particular-database-in-postgresql/24923877#24923877>

```
[usuario@server2asir ~]$psql -U prueba01 -d 2ASIR -h localhost
Password for user prueba01:
psql (14.19 (Ubuntu 14.19-0ubuntu0.22.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, comp
ression: off)
Type "help" for help.

2ASIR=>
```

Dentro podemos comprobar con que usuario estamos usando:

"SELECT current_user;"

```
2ASIR=> SELECT current_user;
current_user
-----
prueba01
(1 row)

2ASIR=> _
```

- Conexión desde un equipo cliente:

Para ello debemos modificar dos archivos que se encuentran en
"/etc/postgresql/14/main/"

El primero es **"postgresql.conf"**. En él debemos buscar la línea **"listen_addresses"**.
 Debemos **"descomentarla"** y cambiar el valor que tenga por **"*"**.

```
listen_addresses = '*'
```

El segundo archivo es **"pg_hba.conf"** en él debemos poner al final del archivo una
 línea con:

host "base_datos" "nombre_usuario" "ips_desde_donde_conectar" md5

Ejemplo que permite a todos los usuarios conectarse a cualquier base de datos desde
 cualquier IP:

host all all 0.0.0.0/0 md5

Puede que tengamos algún problema con la encriptación **"SSL"**:

**FATAL: no pg_hba.conf entry for host "10.8.3.7", user "prueba01", database "2ASIR",
 SSL encryption**

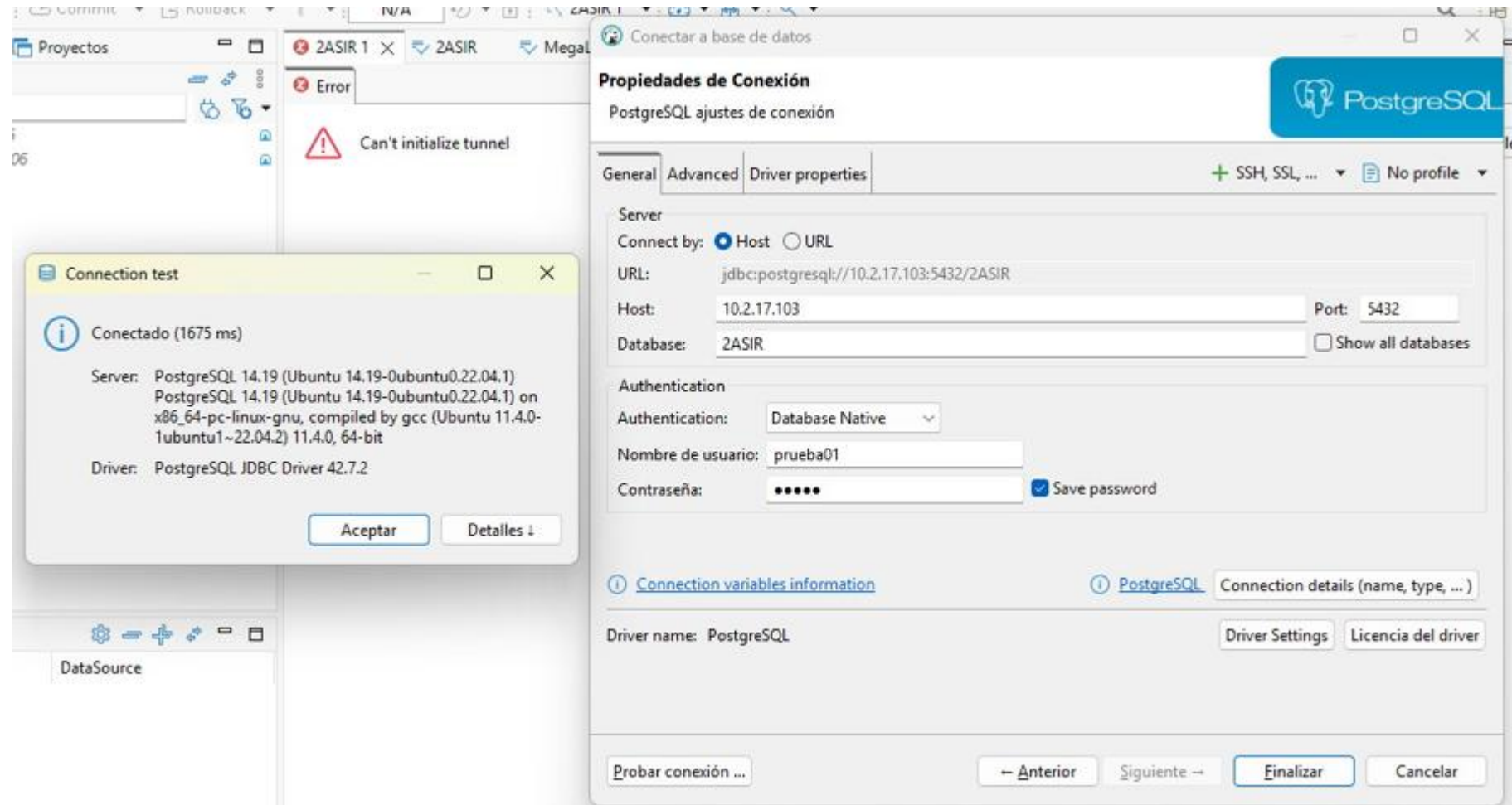
La solución es poner esta línea al final del documento:

hostssl "2ASIR" prueba01 10.8.3.7/32 md5

o

hostssl "2ASIR" prueba01 10.8.3.0/24 md5

Después en DBeaver tenemos que crear una nueva conexión. Elegimos con PostgreSQL y luego solo tenemos que rellenar los datos de la pestaña general.



3. **Términos de uso y condiciones sobre el producto.**

PostgreSQL está publicado bajo la licencia su propia licencia⁶, la cual permite usar, copiar, modificar y distribuir el software y su documentación, sin ningún cobro y sin necesidad de acuerdo entre las partes; siempre y cuando se mantenga el tipo de licencia⁷.

4. **Una enumeración de las posibles instalaciones posibles.**

Además de la instalación en Linux que hemos visto anteriormente, PostgreSQL se puede instalar en Windows, macOS, BSD y Solaris⁸.

5. **¿Tiene alguna interfaz gráfica para usar o extensión? Explica su utilidad.**

Tiene varias, ninguna oficial.

- pgAdmin⁹.
- DBeaver¹⁰.
- DataGrip¹¹.
- TablePlus¹².
- DbVisualizer¹³.

Funcionan como programas clientes para conectarse a PostgreSQL, tanto desde el propio host, como en remoto. Todas permiten administrar las bases de datos sin necesidad de usar la consola (crear, modificar y eliminar bases de datos, tablas, etc). También, si el usuario quiere, pueden hacerse desde la aplicación con comandos. Se pueden hacer consultas SQL y ver los resultados.

⁶ <https://www.postgresql.org/about/licence/>

⁷ <https://opensource.org/license/postgresql>

⁸ <https://www.postgresql.org/download/>

⁹ <https://www.pgadmin.org/>

¹⁰ <https://dbeaver.com/docs/dbeaver/Database-driver-PostgreSQL/>

¹¹ <https://www.datansen.com/blog/postgresql/top-5-gui-tools-for-postgresql/>

¹² Idem.

¹³ Idem.

- Bibliografía extra:

Conectar PostgreSQL y DBeaver:

<https://dbeaver.com/docs/dbeaver/Database-driver-PostgreSQL/>

<https://hevodata.com/learn/dbeaver-postgresql/>

<https://medium.com/@alberto.menendez89/connects-free-database-dbeaver-with-postgresql-step-by-step-guide-2260cd6b4f40>

<https://stackoverflow.com/questions/3278379/how-to-configure-postgresql-to-accept-all-incoming-connections>