

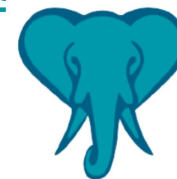
Instalación de PostgreSQL

info@todopostgresql.com



**Todo
PostgreSQL**
by Abatic

1. [Plataformas](#)
2. [Privilegios](#)
3. [Downloads](#)
4. [Instancia](#)
5. [Instalación desde paquete Binario](#)
6. [Instalación Código Fuente](#)
 - [Requisitos](#)
 - [Paquetes opcionales](#)
 - [./configure](#)



7.Post Instalación

8.Desinstalación y liberar espacio

9.Creación del cluster

10.Configuración base

11.Arranque y Parada

12.Conexión inicial

Plataformas

- Distribuciones BSD (FreeBSD, OpenBSD).
- Distribuciones GNU/Linux.
 - Red Hat, CentOS, Fedora, Oracle Linux y análogas.
 - Debian GNU/Linux, Ubuntu y análogas.
 - Suse y OpenSuse.
- MacOS X.
- Windows 32 y 64 bits.
- Solaris.

Privilegios

PostgreSQL corre como un demonio en Unix/Linux o como un servicio en Windows.

Todos los procesos y archivos de datos de PostgreSQL deben pertenecer a un usuario del Sistema Operativo (no requiere privilegios).

Privilegios

Por razones de seguridad, el usuario del Sistema Operativo no debe ser root o una cuenta administrativo.

El usuario de Sistema Operativo de postgres no tiene relación con las cuentas de usuario de la base de datos.

Downloads

Paquete Binario:

- Instalador de un clic.
- Red Hat/CentOS (RPM + yum).
- Debian/Ubuntu (DEB + apt).
- FreeBSD port.
- Solaris package.

Downloads

Código Fuente:

- Compilador ANSI C (gcc o LLVM+clang).

Instancia

Cada instancia de PostgreSQL se denomina “**cluster**”.

- Comprende un directorio de datos que contiene todos los datos y archivos de configuración.
- Uno y **sólo un postmaster** para cada cluster.
- Un único puerto.

Instancia

Se puede hacer referencia al cluster de dos maneras:

- Por la **ubicación** del directorio de datos.
- Por el número de **puerto**.

Instancia

Un único servidor (equipo) puede tener muchas instalaciones.

- Se pueden crear múltiples clusters utilizando el comando `initdb`.
- Pueden compartirse ejecutables (instalación de binarios).
- Cada postmaster: un `$PGDATA` y un `$PGPORT`.

Instalación desde paquete Binario

Existe un instalador creado por [EnterpriseDB](#) para Windows y Mac OS X.

Debe ser ejecutado con un usuario con permisos de super-administrador (root).

PostgreSQL Download

Windows installers

Interactive installer by EDB

Download the **installer** certified by EDB for all supported PostgreSQL versions.

Platform support

The installers are tested by EDB on the following platforms.

PostgreSQL Version	64 Bit Windows Platforms	32 Bit Windows Platforms
14	2019, 2016	
13	2019, 2016	
12	2019, 2016, 2012 R2	

Instalación desde paquete Binario

- Red Hat / CentOS / Fedora / Scientific / Oracle

```
yum/dnf install postgresql-server  
postgresql-setup --initdb  
systemctl enable postgresql.service  
systemctl start postgresql.service
```

Instalación desde paquete Binario

● Debian / Ubuntu

apt-get install postgresql

apt-get install postgresql-contrib

● Mac OS X

Homebrew

Postgres.app (portable)

Fink

MacPorts

Instalación Código Fuente

Requisitos

- ♦ **GNU make** versión 3.80 o superior. En algunas distribuciones también es llamado gmake.

`make --version`

- ♦ Compilador ISO/ANSI C. Lo recomendable es una versión actualizada de **GCC**.
- ♦ Para poder descomprimir el paquete descargado necesitaremos utilizar **tar**, **gzip** o **bzip2**.

Instalación Código Fuente

Requisitos

- ♦ Por defecto utilizaremos **zlib** para el soporte a archivos comprimidos a la hora de exportar / restaurar con `pg_dump` / `pg_restore`.
- ♦ Se puede deshabilitar con la opción `--without-zlib`
`sudo apt-get install zlib1g-dev`

Instalación Código Fuente

Requisitos

- ♦ La herramienta psql utiliza la librería **readline** para usar las teclas de flecha para recuperar y editar comandos anteriores.
- ♦ Se puede deshabilitar con la opción `--without-readline`
`sudo apt-get install libreadline-dev`

Instalación Código Fuente

Paquetes opcionales

- ♦ **Libperl:** para programar con PL/Perl. Mínimo 5.8.3
- ♦ **Python:** para programar con PL/Python. Mínimo 2.6 / 3.1
- ♦ **Tcl:** para programar con PL/Tcl. Mínimo 8.4
- ♦ **Kerberos, OpenSSL, OpenLDAP, PAM:** si queremos utilizar estos servicios para autenticación de usuarios o encriptación.

Instalación Código Fuente `./configure`

El **script de configuración** chequea que todas las rutas están correctas, con los permisos adecuados y que están instaladas y accesibles las herramientas y librerías requeridas. Una vez solventado el error (si lo hubiera), hay que volver a ejecutar el script `./configure`

Instalación Código Fuente `./configure`

Se le puede indicar unas opciones, como las rutas de instalación, paquetes opcionales o deshabilitar librerías.

Ejemplo: para indicar donde se instalarán los ejecutables, se utiliza la opción `--bindir=DIRECTORY`
`./configure --bindir=/curso/bin`

[Más opciones](#)

Instalación Código Fuente

1) [Descargar el Código Fuente.](#)

2) Descomprimir

1) `gunzip postgresql-x.x.tar.gz`

2) `tar xf postgresql-x.x.tar`

3) Ejecutar el script de configuración. **`./configure`**

Nota: Para el fichero .bz2, utilizar bunzip2 en vez de gunzip.

Instalación Código Fuente

4) Compilar o Construir.

1. Compilar **sólo PostgreSQL**. Comando **make**.

All of PostgreSQL is successfully made. Ready to install.

2. Construcción de Postgres con la documentación (HTML y man) y con los módulos adicionales (contrib). Comando **make world**.

PostgreSQL, contrib and HTML documentation successfully made. Ready to install.

Instalación Código Fuente

5) Instalación. Normalmente se ejecuta con permisos root (sudo).

1. Instalación sólo de PostgreSQL.

Comando `sudo make install`.

2. Instalar sólo la documentación.

Comando `sudo make install-docs`.

3. Instalarlo todo, Postgres, documentación y contrib.

Comando `sudo make install-world`.

Post Instalación

Variables de entorno: se utilizan para almacenar directorios o parámetros de conexión.

- Shared Libraries: En algunos S.O. es necesario indicarle donde se encuentran las librerías compartidas.
- En los Sistemas **FreeBSD, HP-UX, Linux, NetBSD, OpenBSD y Solaris**, **no es necesario** establecer la variable de entorno `LD_LIBRARY_PATH`.

Post Instalación

Establecer la variable de entorno **LD_LIBRARY_PATH**

Tipo de shell: Bourne (sh, ksh, bash, zsh):

```
export LD_LIBRARY_PATH=  
/usr/local/pgsql/lib:$LD_LIBRARY_PATH
```

Tipo de shell: csh o tcsh

```
setenv LD_LIBRARY_PATH /usr/local/pgsql/lib
```

Post Instalación

Variables de entorno: se utilizan para almacenar directorios o parámetros de conexión.

➤ Variable PATH: se incluye el directorio de los ejecutables.

Tipo de shell: Bourne (sh, ksh, bash, zsh):

```
export PATH =/usr/local/pgsql/bin/:$PATH
```

Tipo de shell: csh o tcsh

```
set path = (/usr/local/pgsql/bin $path)
```

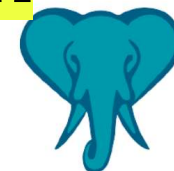
Post Instalación

Puedes incluir estas variables de entorno en el fichero de arranque para todos los usuarios (/etc/profile) o en tu usuario en particular en el fichero (~/.bash_profile).

```
export LD_LIBRARY_PATH=  
/usr/local/pgsql/lib:$LD_LIBRARY_PATH
```

```
PATH=/usr/local/pgsql/bin/:$PATH
```

```
export MANPATH=  
/usr/local/pgsql/share/man:$MANPATH
```



Desinstalación y Liberar espacio

- Desinstalación: no eliminará ningún directorio creado. Comando `make uninstall`.
- Liberar espacio: elimina pequeños archivos utilizados para la instalación. Comando `make clean`.

Creación del Cluster

Usando el comando `initdb` podemos crear un cluster de bases de datos.

Tiene que ejecutarse como el usuario del S.O. bajo el que correrá la instancia (o cambiar permisos).

Recomendado encoding UTF-8.

```
initdb -D <data directory> [-E <encoding>]
```

¡Es necesario configurar el servidor antes de usarlo en producción!

Configuración base

Dos ficheros principales:

- postgresql.conf → configuración de PostgreSQL.
- pg_hba.conf → configuración de autenticación.

Fundamental: asignar otro puerto si tenemos ya un cluster. El puerto por defecto es el 5432.

La configuración por defecto es muy conservadora.
Rendimiento bajo, pero segura y estable.

Arranque y Parada

Utilizamos `pg_ctl` desde la línea de comandos para arrancar y detener el cluster (instancia de servidor).

Necesario: identificación del cluster

`-D $PGDATA (<data_directory>)`

`-p $PGPORT (si no es 5432)`

Arranque y Parada

`pg_ctl start [<options>]` → Arranca el servidor.

`pg_ctl stop [<options>]` → Detiene el servidor.

`pg_ctl status [<options>]` → Estado del servidor.

`pg_ctl restart [<options>]` → Reinicia el servidor.

`pg_ctl reload [<options>]` → Recarga los archivos de configuración (parámetros que no requieran restart).

Arranque y Parada

Parámetros útiles de pg_ctl:

-l <log file> → Escribe salidas de registro al archivo especificado (los logs que salen por consola cuando no se usa logging_collector).

-o OPTIONS → Opciones de configuración.

Arranque y Parada

Parámetros útiles de pg_ctl:

- m smart → Espera por la desconexión de los clientes.
- m fast → La transacción activa (tx) se deshace (rollback) y los clientes son desconectados. (Default)
- m immediate → Aborta procesos sin un apagado limpio.

Conexión inicial

Programa cliente de PostgreSQL: **psql**

psql [OPTIONS]... [DBNAME [USERNAME]]

Otra forma de conectar con el servidor PostgreSQL.

-d BD → Conectarse a la bd.

-U user → El usuario para conectarse.

Conexión inicial

Si no se especifica usuario: el usuario de la BD será el nombre del usuario del S.O. que haya ejecutado el comando `psql`.

Si tampoco se especifica la bd: también tomará el nombre del usuario que ejecute el comando `psql`.

Nota: debe de estar creada la BD y el usuario.

Conexión inicial

Estos detalles son especialmente útiles:

- Para poder conectar ágilmente con PostgreSQL.
- Conectarse introduciendo sólo psql, sin argumentos.
- Resulta muy conveniente.

Conexión inicial

Para cambiar la conexión

```
\c [DBNAME [USERNAME]]
```

```
\connect [DBNAME [USERNAME]]
```

Para salir de la sesión:

```
\q o bien CTRL+D
```

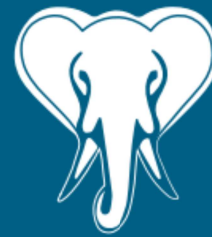
A partir de la versión 11, también exit

Conexión inicial

Además de psql, también disponemos de herramientas gráficas (GUI) para conectarnos con PostgreSQL, como por ejemplo:

- PgAdmin 4 (open source).
- Postgres Enterprise Manager.
- DBeaver

Cursos de PostgreSQL para DBA y Developers



**Todo
PostgreSQL**
by Abatic