

# PostgreSQL (I)

- PostgreSQL es un Sistema Gestor de Bases de Datos (SGBD) relacionales.
- Es código libre bajo licencia BSD.
- Su primera versión operativa estuvo disponible en 1988.
- La primera versión con soporte del lenguaje SQL estuvo disponible en 1995.
  - Anteriormente utilizaba un interprete de consultas basado en lenguaje QUEL - Ingres.

# PostgreSQL (II)

- Su novedad inicial fue la posibilidad de definir objetos:
  - Nuevos tipos de datos basados en tipos de datos simples.
- Funciona en multitud de sistemas operativos:
  - Linux.
  - Windows.
  - FreeBSD/OpenBSD.
  - Mac OS X.
  - Solaris.

# PostgreSQL (III)

- Las conexiones a la base de datos se realizan mediante TCP/IP.
  - En Linux permite la conexión mediante sockets locales.
- El tamaño máximo del espacio de tablas es teóricamente ilimitado.
  - La limitación la impone el sistema operativo:
    - Linux 2.4 o superior: 16 Tbytes en ext4.
    - Windows: 16 Tbytes en NTFS.

# PostgreSQL (IV)

- Los límites de PostgreSQL versiones 8.x y 9.x son:

<u>Límite</u>	<u>Valor</u>
Tamaño máximo de la base de datos	Ilimitado
Tamaño máximo de una tabla	32 TB
Tamaño máximo de una fila	1,6 TB
Tamaño máximo de un campo	1 GB
Máximo número de filas por tabla	Ilimitado
Máximo número de columnas por tabla	250 a 1600, dependiendo del tipo de columna.
Máximo número de índices por tabla	Ilimitado

# Instalación de PostgreSQL (I)

- La instalación de PostgreSQL esta formada por varios paquetes:
  - postgresql: Programas cliente y librerías compartidas.
  - postgresql-libs: Librerías compartidas.
  - postgresql-server: Servidor de postgresQL.
- Existen además otros paquetes adicionales:
  - postgresql-test: Programas para comprobar el funcionamiento y rendimiento.
  - postgresql-jdbc: JDBC driver para MySQL.
  - postgresql-odbc: ODBC driver para postgresQL.
  - postgresql-devel: Ficheros para desarrollar aplicaciones que usen postgresQL.

# Instalación de PostgreSQL (II)

- La instalación de PostgreSQL se realiza ejecutando el comando:

```
yum -y install postgresql-server
```

- El comando instala el paquete postgresql-server con todas las dependencias necesarias.

# Instalación de PostgreSQL (III)

```
root@localhost:~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
(1/3): postgresql-8.4.20-1.el6_5.i686.rpm | 2.6 MB 00:00
(2/3): postgresql-libs-8.4.20-1.el6_5.i686.rpm | 205 kB 00:00
(3/3): postgresql-server-8.4.20-1.el6_5.i686.rpm | 3.4 MB 00:02
-----
Total | 1.3 MB/s | 6.2 MB 00:04
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : postgresql-libs-8.4.20-1.el6_5.i686 1/3
  Installing : postgresql-8.4.20-1.el6_5.i686 2/3
  Installing : postgresql-server-8.4.20-1.el6_5.i686 3/3
  Verifying : postgresql-8.4.20-1.el6_5.i686 1/3
  Verifying : postgresql-server-8.4.20-1.el6_5.i686 2/3
  Verifying : postgresql-libs-8.4.20-1.el6_5.i686 3/3

Installed:
  postgresql-server.i686 0:8.4.20-1.el6_5

Dependency Installed:
  postgresql.i686 0:8.4.20-1.el6_5      postgresql-libs.i686 0:8.4.20-1.el6_5

Complete!
[root@localhost ~]#
```



# Instalación de PostgreSQL (IV)

- Una vez instalado PostgreSQL es necesario crear los directorios y ficheros necesarios para su funcionamiento.
  - El directorio `/var/lib/pgsql/data`:
    - Contiene los ficheros de la base de datos.
  - El directorio `/var/lib/pgsql/data/pg_log`:
    - Contiene los ficheros de log de ejecución de PostgreSQL.
  - El fichero `/var/lib/pgsql/pgstartup.log`:
    - Contiene los mensajes de arranque del servidor de PostgreSQL.



# Instalación de PostgreSQL (V)

- Además, es necesario crear los ficheros iniciales de la base de datos ejecutando el comando:

```
su -l postgres -c "/usr/bin/initdb  
--pgdata='/var/lib/pgsql/data' --auth='ident'" >  
"/var/lib/pgsql/pgstartup.log" 2>&1 < /dev/null
```

- Todo lo comentado anteriormente puede realizarse ejecutando el comando:

```
service postgresql initdb
```

# Instalación de PostgreSQL (VI)

- El servidor de PostgreSQL puede arrancarse, pararse o rearrancarse de forma manual con el comando:

```
service postgresql {start|stop|restart}
```

- Puede configurarse también para que arranque o se detenga con el sistema ejecutando el comando:

```
chkconfig postgresql on
```

# Instalación de PostgreSQL (VII)



A terminal window titled 'root@localhost:~' with a menu bar containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal shows the following commands and output:

```
[root@localhost ~]# service postgresql initdb
Iniciando la base de datos: [ OK ]
[root@localhost ~]# service postgresql start
Iniciando servicios postgresql: [ OK ]
[root@localhost ~]#
```

Two red arrows point from the right side of the terminal to the 'initdb' and 'start' commands.

# Instalación de PostgreSQL (VIII)

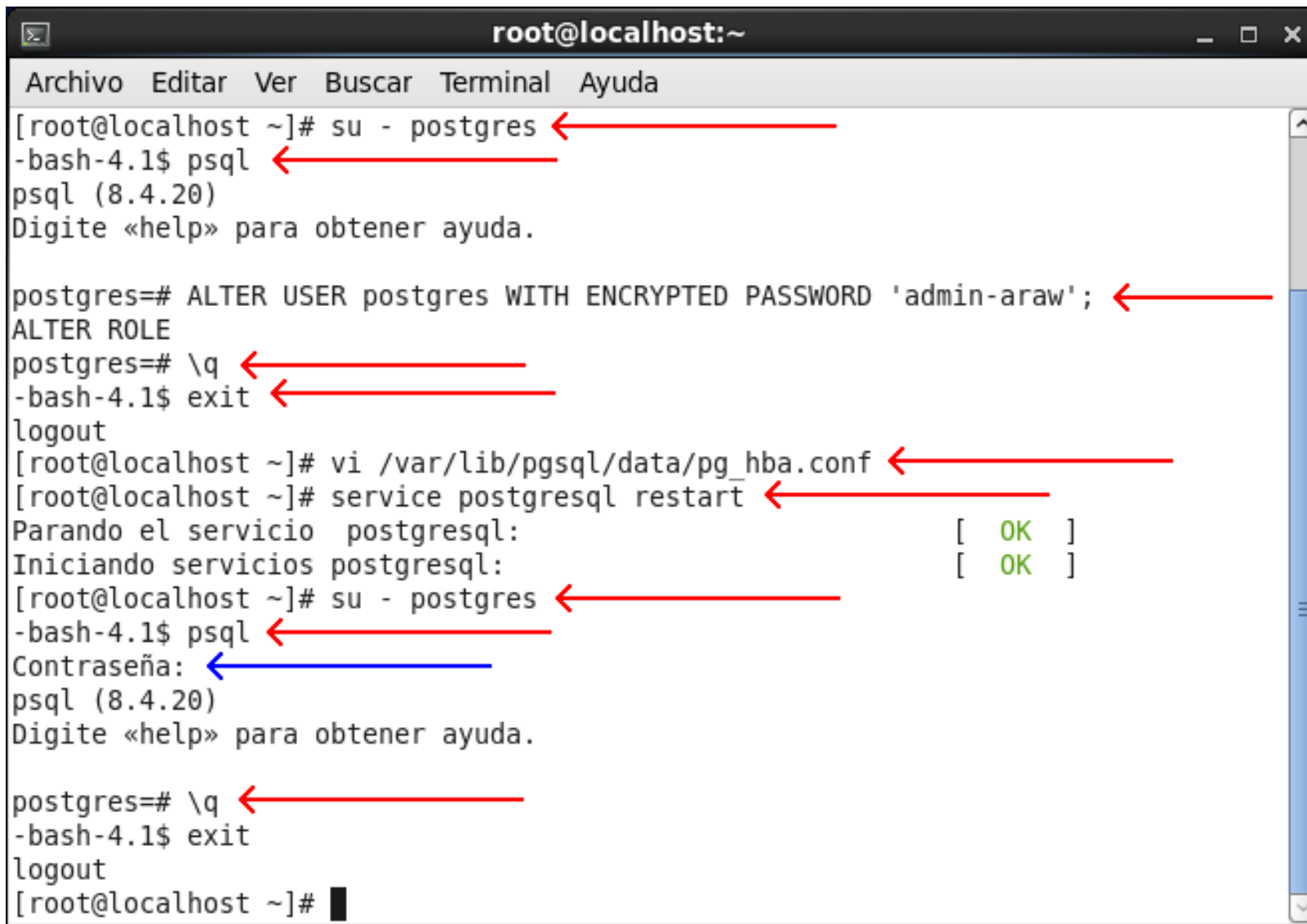
- Inicialmente el usuario postgres (administrador) de la base de datos no tiene contraseña:

```
su - postgres  
psql
```

```
ALTER USER postgres WITH ENCRYPTED PASSWORD 'clave';
```

- Además, esta configurado para que el usuario postgres pueda acceder sin proporcionar contraseña.
  - Modificar el fichero /var/lib/pgsql/data/pg\_hba.conf y cambiar en todas las líneas ident por md5.

# Instalación de PostgreSQL (IX)



```
root@localhost:~  
Archivo  Editar  Ver  Buscar  Terminal  Ayuda  
[root@localhost ~]# su - postgres  
-bash-4.1$ psql  
psql (8.4.20)  
Digite «help» para obtener ayuda.  
  
postgres=# ALTER USER postgres WITH ENCRYPTED PASSWORD 'admin-araw';  
ALTER ROLE  
postgres=# \q  
-bash-4.1$ exit  
logout  
[root@localhost ~]# vi /var/lib/pgsql/data/pg_hba.conf  
[root@localhost ~]# service postgresql restart  
Parando el servicio postgresql: [ OK ]  
Iniciando servicios postgresql: [ OK ]  
[root@localhost ~]# su - postgres  
-bash-4.1$ psql  
Contraseña:  
psql (8.4.20)  
Digite «help» para obtener ayuda.  
  
postgres=# \q  
-bash-4.1$ exit  
logout  
[root@localhost ~]#
```

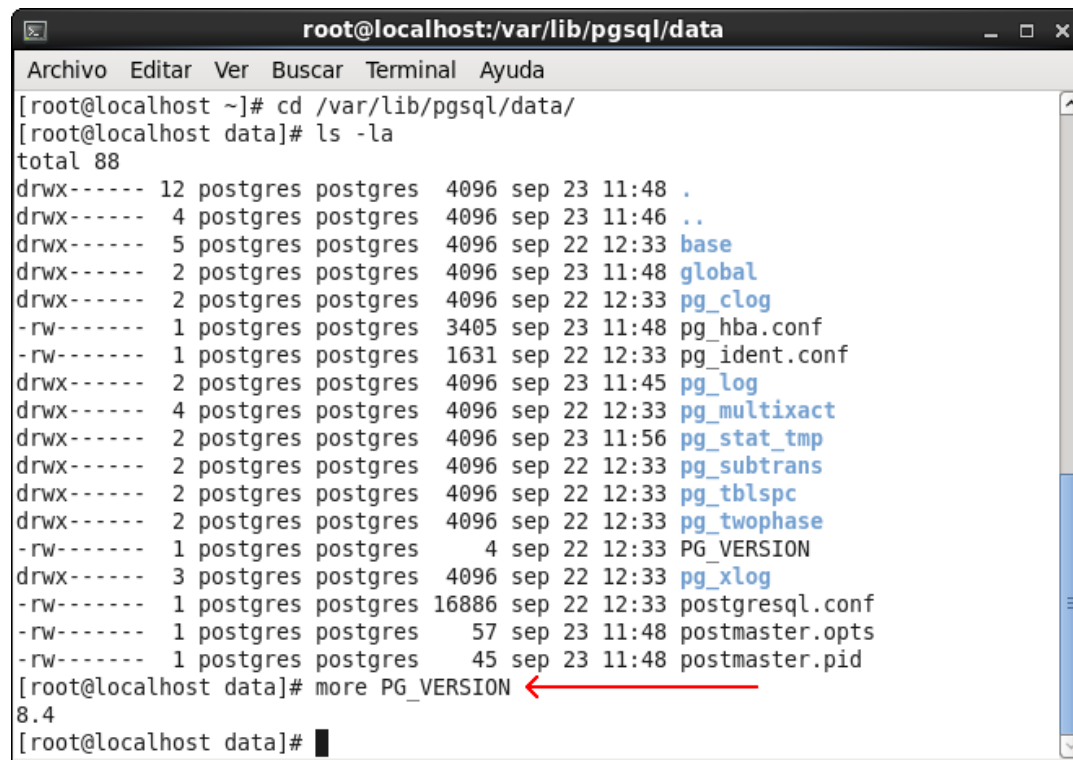
# Configuración de PostgreSQL (I)

- Los ficheros de configuración de PostgreSQL son:
  - /var/lib/pgsql/data/PG\_VERSION.
    - Versión de PostgreSQL del servidor.
  - /var/lib/pgsql/data/pg\_hba.conf.
    - Autenticación de usuarios y accesos a las bases de datos.
  - /var/lib/pgsql/data/pg\_ident.conf.
    - Fichero auxiliar a pg\_hba.conf. Indica la relación entre usuarios del S.O. y de PostgreSQL con autenticación ident.
  - /var/lib/pgsql/data/postgresql.conf.
    - Fichero de configuración principal.
    - Contiene los parámetros que configuran el funcionamiento del servidor.



# Configuración de PostgreSQL (II)

- El fichero PG\_VERSION:
  - Es un fichero de texto.
  - Indica la versión de postgresQL instalada.



```
root@localhost:/var/lib/pgsql/data
Archivo Editar Ver Buscar Terminal Ayuda
[root@localhost ~]# cd /var/lib/pgsql/data/
[root@localhost data]# ls -la
total 88
drwx----- 12 postgres postgres 4096 sep 23 11:48 .
drwx----- 4 postgres postgres 4096 sep 23 11:46 ..
drwx----- 5 postgres postgres 4096 sep 22 12:33 base
drwx----- 2 postgres postgres 4096 sep 23 11:48 global
drwx----- 2 postgres postgres 4096 sep 22 12:33 pg_clog
-rw----- 1 postgres postgres 3405 sep 23 11:48 pg_hba.conf
-rw----- 1 postgres postgres 1631 sep 22 12:33 pg_ident.conf
drwx----- 2 postgres postgres 4096 sep 23 11:45 pg_log
drwx----- 4 postgres postgres 4096 sep 22 12:33 pg_multixact
drwx----- 2 postgres postgres 4096 sep 23 11:56 pg_stat_tmp
drwx----- 2 postgres postgres 4096 sep 22 12:33 pg_subtrans
drwx----- 2 postgres postgres 4096 sep 22 12:33 pg_tblspc
drwx----- 2 postgres postgres 4096 sep 22 12:33 pg_twophase
-rw----- 1 postgres postgres 4 sep 22 12:33 PG_VERSION
drwx----- 3 postgres postgres 4096 sep 22 12:33 pg_xlog
-rw----- 1 postgres postgres 16886 sep 22 12:33 postgresql.conf
-rw----- 1 postgres postgres 57 sep 23 11:48 postmaster.opts
-rw----- 1 postgres postgres 45 sep 23 11:48 postmaster.pid
[root@localhost data]# more PG_VERSION
8.4
[root@localhost data]#
```



# Configuración de PostgreSQL (III)

- El fichero `pg_hba.conf`:
  - Contiene los tipos de conexión, modos de autenticación permitidos, etc.
  - La sintaxis de sus líneas es:

```
<tipo> <basedatos> <usuario> [dirección IP]  
<método> [opciones]
```

# Configuración de PostgreSQL (IV)

- El campo <tipo> indica desde donde se permiten las conexiones.
- Puede tomar los valores:
  - local: Conexiones locales mediante socket.
  - host: Conexiones TCP/IP en modo plano o cifradas utilizando SSL.
  - hostssl: Conexiones TCP/IP cifradas utilizando SSL.
  - hostnossll: Conexiones TCP/IP en modo plano.

# Configuración de PostgreSQL (V)

- El campo `<basedatos>` indica a que base de datos se refiere esta línea de configuración.
- Puede tomar los valores:
  - `all`: Todas las bases de datos.
  - `sameuser`: El nombre de la base de datos que coincide con el nombre del usuario.
  - `samerole`: El nombre de la base de datos que coincide con el rol del usuario (grupo del usuario).
  - `nombre`: El nombre de una de las bases de datos existentes.
- Si se desea, puede ser una lista, separada por comas, de los valores anteriores.

# Configuración de PostgreSQL (VI)

- El campo `<usuario>` indica los usuarios que pueden conectarse a la base de datos con ese tipo de conexión.
- Puede tomar los valores:
  - `all`: Todas los usuarios.
  - `nombre usuario`: Un nombre de usuario.
  - `nombre grupo`: Un nombre de grupo.
    - Para diferenciarlo de un nombre de usuario debe ir precedido del signo `+`.
- Si se desea, puede ser una lista, separada por comas, de los valores anteriores.

# Configuración de PostgreSQL (VII)

- El campo opcional <dirección IP> especifica las direcciones IP que están autorizadas a conectarse a la base de datos mediante el usuario y tipo indicado.
- Permite especificar una dirección IPv4 o IPv6 en formato dirección IP/mascara.
  - Si no se especifica se supone cualquier dirección IP.
- Por ejemplo:
  - 147.156.0.0/16 para cualquier dirección IPv4 de la UV.
  - 2001:720:1014::/80 para cualquier dirección IPv6 de la UV.
  - 147.156.222.65/32 para indicar un ordenador en particular.

# Configuración de PostgreSQL (VIII)

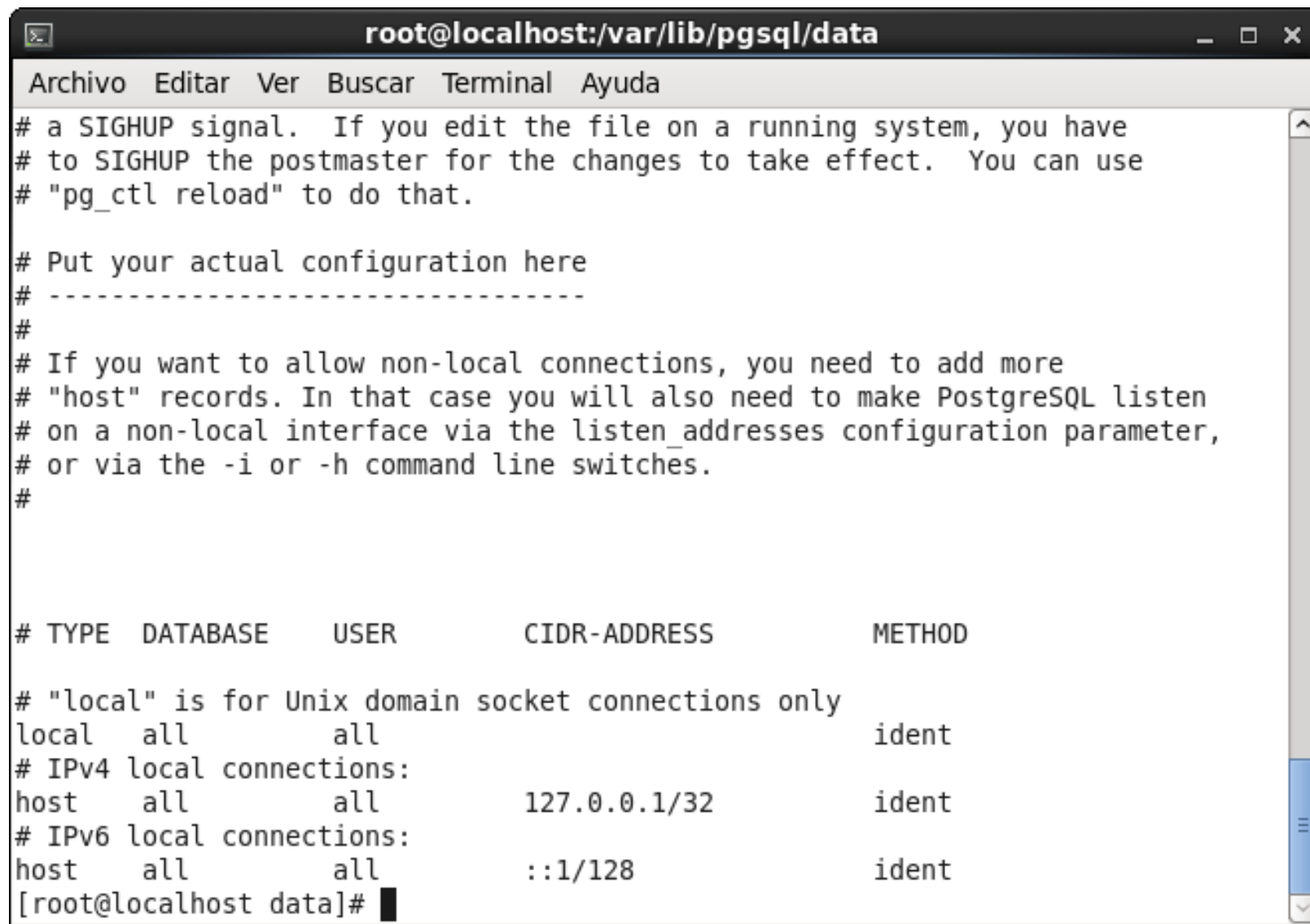
- El campo <método> indica el tipo de autenticación requerido.
- Algunos tipos de autenticación son:
  - **ident**: Utiliza el nombre del usuario proporcionado por el sistema operativo.
    - Puede modificarse ese nombre con una opción.
    - No solicita contraseña.
  - **trust**: Permitir el acceso a cualquier usuario que pueda conectarse sin solicitar ningún tipo de autenticación.
    - Solo debería usarse en conexiones locales y de forma muy puntual (migración de versión de PostgreSQL).
  - **password**: La contraseña es enviada en texto plano.
  - **md5**: La contraseña es enviada como un compendio.

# Configuración de PostgreSQL (IX)

- El campo `<opciones>` es un campo opcional que se utiliza en determinados casos.
  - Autenticación `ident` y cambio (mapeo) de nombre en los usuarios.
    - `map=nombre_mapa`.



# Configuración de PostgreSQL (X)



```
root@localhost:/var/lib/pgsql/data
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
# a SIGHUP signal.  If you edit the file on a running system, you have
# to SIGHUP the postmaster for the changes to take effect.  You can use
# "pg_ctl reload" to do that.

# Put your actual configuration here
# -----
#
# If you want to allow non-local connections, you need to add more
# "host" records.  In that case you will also need to make PostgreSQL listen
# on a non-local interface via the listen_addresses configuration parameter,
# or via the -i or -h command line switches.
#

# TYPE  DATABASE  USER  CIDR-ADDRESS  METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     ident
# IPv4 local connections:
host    all             all             127.0.0.1/32   ident
# IPv6 local connections:
host    all             all             ::1/128        ident
[root@localhost data]#
```

# Configuración de PostgreSQL (XI)

- El fichero `pg_ident.conf`:
  - Contiene la modificación de nombre de usuarios que se realiza en la autenticación `ident` si se especifica un mapeo de los mismos.
  - Esta formado por líneas con sintaxis:

`<mapa> <usuario_so> <usuario_pg>`

- Donde:
  - `mapa`: Es el nombre del cambio.
  - `usuario_so`: Es el nombre del usuario del S.O.
  - `usuario_pg`: Es el nombre del usuario de PostgreSQL.
- Inicialmente no contiene ninguna línea sin comentar.

# Configuración de PostgreSQL (XII)

- El fichero `postgresql.conf`:
  - Contiene la configuración de funcionamiento de PostgreSQL.
  - Esta formado por líneas con sintaxis:  
`<nombre> = <valor>`
    - Donde:
      - `nombre`: Nombre del atributo que se desea asignar.
      - `valor`: Valor que se desea asignar.
- Permite ajustar el funcionamiento de PostgreSQL a los requisitos que existan.

# Configuración de PostgreSQL (XIII)

- Algunas de las opciones más comunes del fichero `postgresql.conf` son:
  - `listen_addresses`: Direcciones IP donde permanece a la escucha el servidor.
    - Por defecto únicamente escucha en 'localhost'.
    - Puede ser una lista de direcciones separada por comas.
    - Todas las direcciones se especifica como '\*'.
  - `port`: Puerto donde escucha el servidor.
    - Por defecto es el 5432.
  - `max_connections`: Número máximo de conexiones.
    - Por defecto esta configurado a 100 conexiones simultáneas.

# Configuración de PostgreSQL (XIV)

- Algunas opciones del fichero `postgresql.conf` que pueden afectar a su rendimiento son:
  - `shared_buffers`: Tamaño de la memoria cache que utiliza PostgreSQL.
    - Se recomienda que sea un 25% de la memoria.
  - `temp_buffers`: Tamaño del buffer temporal de cada conexión.
    - Su valor optimo depende de la carga de cada sesión.
  - `work_mem`: Tamaño de memoria usado para ordenaciones y consultas agrupadas.
    - Su valor optimo depende del uso de estas consultas.
  - `bgwriter_delay`: Retraso de escritura de datos a disco.
    - Su incremento mejora el rendimiento pero aumenta el riesgo de perdida de datos por fallo del sistema.

# Bases de datos en PostgreSQL (I)

- PostgreSQL permite crear bases de datos con el comando **CREATE DATABASE**.

```
CREATE DATABASE nombre [TEMPLATE=patrón]  
[OWNER=propietario] [ENCODING=codificación]  
[LC_COLLATE=idioma] [LC_CTYPE=idioma];
```

- Se pueden listar las bases de datos existentes con el comando de PostgreSQL:

```
\1
```

- Y borrar las bases de datos con **DROP DATABASE**.

```
DROP DATABASE nombre;
```



# Bases de datos en PostgreSQL (II)

```

root@localhost:~
Archivo Editar Ver Buscar Terminal Ayuda
-bash-4.1$ psql ←
Contraseña:
psql (8.4.20)
Digite «help» para obtener ayuda.

postgres=# \l ←
                                Listado de base de datos
  Nombre  | Dueño   | Codificación | Collation | Ctype   | Privilegios
-----+-----+-----+-----+-----+-----
 postgres | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 |
 template0 | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres
           |          |              |             |             | : postgres=CTc/postgres
 template1 | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres
           |          |              |             |             | : postgres=CTc/postgres
(3 filas)

postgres=# CREATE DATABASE db1 TEMPLATE=template0 LC_COLLATE='es_ES.UTF-8' LC_CTYPE='es_ES.UTF-8'; ←
CREATE DATABASE
postgres=# \l ←
                                Listado de base de datos
  Nombre  | Dueño   | Codificación | Collation | Ctype   | Privilegios
-----+-----+-----+-----+-----+-----
    db1   | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 |
 postgres | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 |
 template0 | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres
           |          |              |             |             | : postgres=CTc/postgres
 template1 | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres
           |          |              |             |             | : postgres=CTc/postgres
(4 filas)

postgres=# █

```



# Bases de datos en PostgreSQL (III)

```

root@localhost:~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
postgres=# CREATE DATABASE db1 TEMPLATE=template0 LC_COLLATE='es_ES.UTF-8' LC_CTYPE='es_ES.UTF-8';
CREATE DATABASE
postgres=# \l

                          Listado de base de datos
  Nombre  | Dueño   | Codificación | Collation | Ctype  | Privilegios
-----+-----+-----+-----+-----+-----
db1       | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 |
postgres  | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 |
template0 | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres
          |          |              |              |              | : postgres=CTc/postgres
template1 | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres
          |          |              |              |              | : postgres=CTc/postgres
(4 filas)

postgres=# DROP DATABASE db1;
DROP DATABASE
postgres=# \l

                          Listado de base de datos
  Nombre  | Dueño   | Codificación | Collation | Ctype  | Privilegios
-----+-----+-----+-----+-----+-----
postgres  | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 |
template0 | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres
          |          |              |              |              | : postgres=CTc/postgres
template1 | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres
          |          |              |              |              | : postgres=CTc/postgres
(3 filas)

postgres=# \q
-bash-4.1$

```

# Bases de datos en PostgreSQL (IV)

- PostgreSQL también permite crear una base de datos mediante el comando del sistema `createdb`.

```
createdb [-E codificación] [-l idioma] [-O propietario] [-T esquema] nombre
```

- -E indica la codificación de caracteres que usara la nueva base de datos.
- -l indica el idioma de la base de datos.
- -O indica que usuario será el propietario de la nueva base de datos.
- -T patrón con el que crear la base de datos.
- Y borrar una base de datos con el comando `dropdb`.

```
dropdb nombre
```

# Bases de datos en PostgreSQL (V)

```

root@localhost:~
Archivo Editar Ver Buscar Terminal Ayuda
postgres=# \l ←
Listado de base de datos
  Nombre | Dueño   | Codificación | Collation | Ctype   | Privilegios
-----+-----+-----+-----+-----+-----
 postgres | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 |
 template0 | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres
                                     : postgres=CTc/postgres
 template1 | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres
                                     : postgres=CTc/postgres
(3 filas)

postgres=# \q ←
-bash-4.1$ createdb -l es_ES.UTF-8 -T template0 db1 ←
Contraseña:
-bash-4.1$ psql ←
Contraseña:
psql (8.4.20)
Digite «help» para obtener ayuda.

postgres=# \l ←
Listado de base de datos
  Nombre | Dueño   | Codificación | Collation | Ctype   | Privilegios
-----+-----+-----+-----+-----+-----
    db1   | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 | ←
 postgres | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 |
 template0 | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres
                                     : postgres=CTc/postgres
 template1 | postgres | UTF8         | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres
                                     : postgres=CTc/postgres
(4 filas)

postgres=#

```

# Bases de datos en PostgreSQL (VI)

```
root@localhost:~  
Archivo Editar Ver Buscar Terminal Ayuda  
postgres=# \l  
Listado de base de datos  
+-----+-----+-----+-----+-----+-----+  
| Nombre | Dueño | Codificación | Collation | Ctype | Privilegios |  
+-----+-----+-----+-----+-----+-----+  
| db1 | postgres | UTF8 | es_ES.UTF-8 | es_ES.UTF-8 | |  
| postgres | postgres | UTF8 | es_ES.UTF-8 | es_ES.UTF-8 | |  
| template0 | postgres | UTF8 | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres  
| | | | | | : postgres=CTc/postgres  
| template1 | postgres | UTF8 | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres  
| | | | | | : postgres=CTc/postgres  
+-----+-----+-----+-----+-----+-----+  
(4 filas)  
  
postgres=# \q  
-bash-4.1$ dropdb db1  
Contraseña:  
-bash-4.1$ psql  
Contraseña:  
psql (8.4.20)  
Digite «help» para obtener ayuda.  
  
postgres=# \l  
Listado de base de datos  
+-----+-----+-----+-----+-----+-----+  
| Nombre | Dueño | Codificación | Collation | Ctype | Privilegios |  
+-----+-----+-----+-----+-----+-----+  
| postgres | postgres | UTF8 | es_ES.UTF-8 | es_ES.UTF-8 | |  
| template0 | postgres | UTF8 | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres  
| | | | | | : postgres=CTc/postgres  
| template1 | postgres | UTF8 | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres  
| | | | | | : postgres=CTc/postgres  
+-----+-----+-----+-----+-----+-----+  
(3 filas)  
  
postgres=#
```

# Tablas en PostgreSQL (I)

- Las tablas en PostgreSQL se crean dentro de bases de datos existentes.
- Para utilizar una determinada base de datos es necesario conectarse indicando el nombre de la base de datos que se desea usar:

```
psql [-U usuario] [-W] basedatos
```

- -U permite especificar un usuario distinto al actual en el S.O.
  - -W indica que se solicite la contraseña.
- El comando que se utiliza para crear una tabla es:

```
CREATE nombre_tabla(elementos);
```

# Tablas en PostgreSQL (II)

- Existen comandos adicionales muy útiles para manejar las tablas:
  - `\d`: Lista las tablas existentes en la base de datos.
  - `\d nombre_tabla`: Muestra los atributos (columnas) de la tabla.



# Tablas en PostgreSQL (III)

```
root@localhost:~  
Archivo Editar Ver Buscar Terminal Ayuda  
-bash-4.1$ psql db1  
Contraseña:  
psql (8.4.20)  
Digite «help» para obtener ayuda.  
  
db1=# CREATE TABLE tabla1(a INTEGER NOT NULL, PRIMARY KEY(a));  
NOTICE: CREATE TABLE / PRIMARY KEY creará el índice implícito «tabla1_pkey» para la tabla «tabla1»  
CREATE TABLE  
db1=# CREATE TABLE tabla2(b INTEGER NOT NULL, FOREIGN KEY(b) REFERENCES tabla1(a));  
CREATE TABLE  
db1=# \d  
Listado de relaciones  
Esquema | Nombre | Tipo | Dueño  
-----+-----+-----+-----  
public | tabla1 | tabla | postgres  
public | tabla2 | tabla | postgres  
(2 filas)  
  
db1=# \d tabla1  
Tabla «public.tabla1»  
Columna | Tipo | Modificadores  
-----+-----+-----  
a | integer | not null  
Índices:  
"tabla1_pkey" PRIMARY KEY, btree (a)  
Referenciada por:  
TABLE "tabla2" CONSTRAINT "tabla2_b_fkey" FOREIGN KEY (b) REFERENCES tabla1(a)  
  
db1=# \d tabla2  
Tabla «public.tabla2»  
Columna | Tipo | Modificadores  
-----+-----+-----  
b | integer | not null  
Restricciones de llave foránea:  
"tabla2_b_fkey" FOREIGN KEY (b) REFERENCES tabla1(a)  
  
db1=#
```



# Usuarios en PostgreSQL (I)

- Los usuarios en PostgreSQL se crean con el comando:

```
CREATE ROLE nombre [LOGIN] [ENCRYPTED |  
UNENCRYPTED] PASSWORD 'clave';
```

- -LOGIN indica que el usuario puede acceder a las bases de datos.
- ENCRYPTED|UNENCRYPTED indica si la contraseña se almacena cifrada o no.

- Se listan con el comando:

```
SELECT * FROM pg_user;
```

- Y se borran con el comando:

```
DROP ROLE nombre;
```

# Usuarios en PostgreSQL (II)

```

root@localhost:~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
-bash-4.1$ psql ←
Contraseña:
psql (8.4.20)
Digite «help» para obtener ayuda.

postgres=# SELECT * FROM pg_user; ←
 username | usesysid | usecreatedb | usesuper | usecatupd | passwd | valuntil | useconfig
-----+-----+-----+-----+-----+-----+-----+-----
 postgres |      10 | t          | t        | t         | ***** |          | 
(1 fila)

postgres=# CREATE ROLE usuariol LOGIN ENCRYPTED PASSWORD 'mi_clave'; ←
CREATE ROLE
postgres=# SELECT * FROM pg_user; ←
 username | usesysid | usecreatedb | usesuper | usecatupd | passwd | valuntil | useconfig
-----+-----+-----+-----+-----+-----+-----+-----
 postgres |      10 | t          | t        | t         | ***** |          | 
 usuariol |    16407 | f          | f        | f         | ***** |          | ←
(2 filas)

postgres=# DROP ROLE usuariol; ←
DROP ROLE
postgres=# SELECT * FROM pg_user; ←
 username | usesysid | usecreatedb | usesuper | usecatupd | passwd | valuntil | useconfig
-----+-----+-----+-----+-----+-----+-----+-----
 postgres |      10 | t          | t        | t         | ***** |          | 
(1 fila)

postgres=# █

```

# Usuarios en PostgreSQL (III)

- PostgreSQL también permite crear usuarios mediante el comando del sistema `createuser`.

```
createuser [-s|-S] [-d|-D] [-r|-R] [-P [-E|-N]]  
nombre
```

- `-s|-S` indica si el usuario será administrador o no.
- `-d|-D` indica si podrá crear bases de datos o no.
- `-r|-R` indica si podrá crear otros usuarios o no.
- `-P` indica que se solicite el password del usuario.
- `-E|-N` indican que el password se cifre o no.
- Y borrar un usuario con el comando `dropuser`.

```
dropuser nombre
```

# Usuarios en PostgreSQL (IV)

```
root@localhost:~  
Archivo Editar Ver Buscar Terminal Ayuda  
-bash-4.1$ createuser -P -E usuario1  
Ingrese la contraseña para el nuevo rol:  
Ingrésela nuevamente:  
¿Será el nuevo rol un superusuario? (s/n) n  
¿Debe permitírsele al rol la creación de bases de datos? (s/n) n  
¿Debe permitírsele al rol la creación de otros roles? (s/n) n  
Contraseña:  
-bash-4.1$ psql  
Contraseña:  
psql (8.4.20)  
Digite «help» para obtener ayuda.  
  
postgres=# SELECT * FROM pg_user;  
username | usesysid | usecreatedb | usesuper | usecatupd | passwd | valuntil | useconfig  
-----+-----+-----+-----+-----+-----+-----+-----  
postgres |      10 | t          | t        | t         | ***** |          |  
usuario1 |    16408 | f          | f        | f         | ***** |          |  
(2 filas)  
  
postgres=# \q  
-bash-4.1$ dropuser usuario1  
Contraseña:  
-bash-4.1$ psql  
Contraseña:  
psql (8.4.20)  
Digite «help» para obtener ayuda.  
  
postgres=# SELECT * FROM pg_user;  
username | usesysid | usecreatedb | usesuper | usecatupd | passwd | valuntil | useconfig  
-----+-----+-----+-----+-----+-----+-----+-----  
postgres |      10 | t          | t        | t         | ***** |          |  
(1 fila)  
  
postgres=#
```

# Copias de seguridad en PostgreSQL (I)

- PostgreSQL permite hacer copias de seguridad con el comando `pg_dump`.
  - El comando posee múltiples opciones, proporcionando por defecto un fichero en formato texto.
    - `--create`: Crear al principio de la copia los comandos necesarios para crear la base de datos.
    - `-c`: Crear en la copia los comandos necesarios para asegurar que al restaurarla no existan los objetos que se restauran.
    - `-f`: Fichero de salida, si se omite se usa la salida estándar
    - `-t`: Copiar solo la tabla o tablas indicadas.

```
pg_dump basedatos > fichero_copia.sql
```

# Copias de seguridad en PostgreSQL (II)

- La copia de seguridad se puede con el comando `pg_restore`.
  - El comando posee múltiples opciones:
    - `-C`: Utilizar los comandos existentes en la copia para crear la base de datos donde hacer la restauración.
    - `-c`: Utilizar los comandos existentes en la copia para asegurarse que no existen los objetos a restaurar.
    - `-d`: Restaurar en la base de datos indicada, la cual debe existir. Esta opción no tiene efecto si la opción `-C` es especificada.
    - `-t`: Restaurar solo la tabla o tablas indicadas.

```
pg_restore -d basedatos fichero_copia.sql
```



# Copias de seguridad en PostgreSQL (III)

- También es posible restaurar la base de datos utilizando el comando psql:

```
psql basedatos < fichero_copia.sql
```

- De forma general, la base de datos debe existir.