

PostgreSQL: Introducción



Mario González T.
mgonzalez@mandriva.com

- **Definición**
- **Historia**
- **Características elementales**
- **Métodos de conexión**
- **Soporte**

Definición

- **Base de datos Open Source relacional; modelo cliente-servidor.**
- **Es la base de datos open source más avanzada del mundo. Usada en empresas, universidades, instituciones públicas y privadas,...**
- **Licencia BSD**
- **Actualmente desarrollada por *The PostgreSQL Development Group*.**

Historia

- **1977 – 1985** : Ingress (U. Berkeley)
- **1986 – 1994** : Postgres (M.Stonebraker; U. Berkeley)
- **1994 – 1995** : Postgres95 (A.Yu, J.Chen)
- **1996 – Hoy** : PostgreSQL (*Abr.:* Postgres)
 - Por favor, NO ES *posgre, post, pogre,...*

A la fecha (Octubre 2007) versión estable => 8.2.x

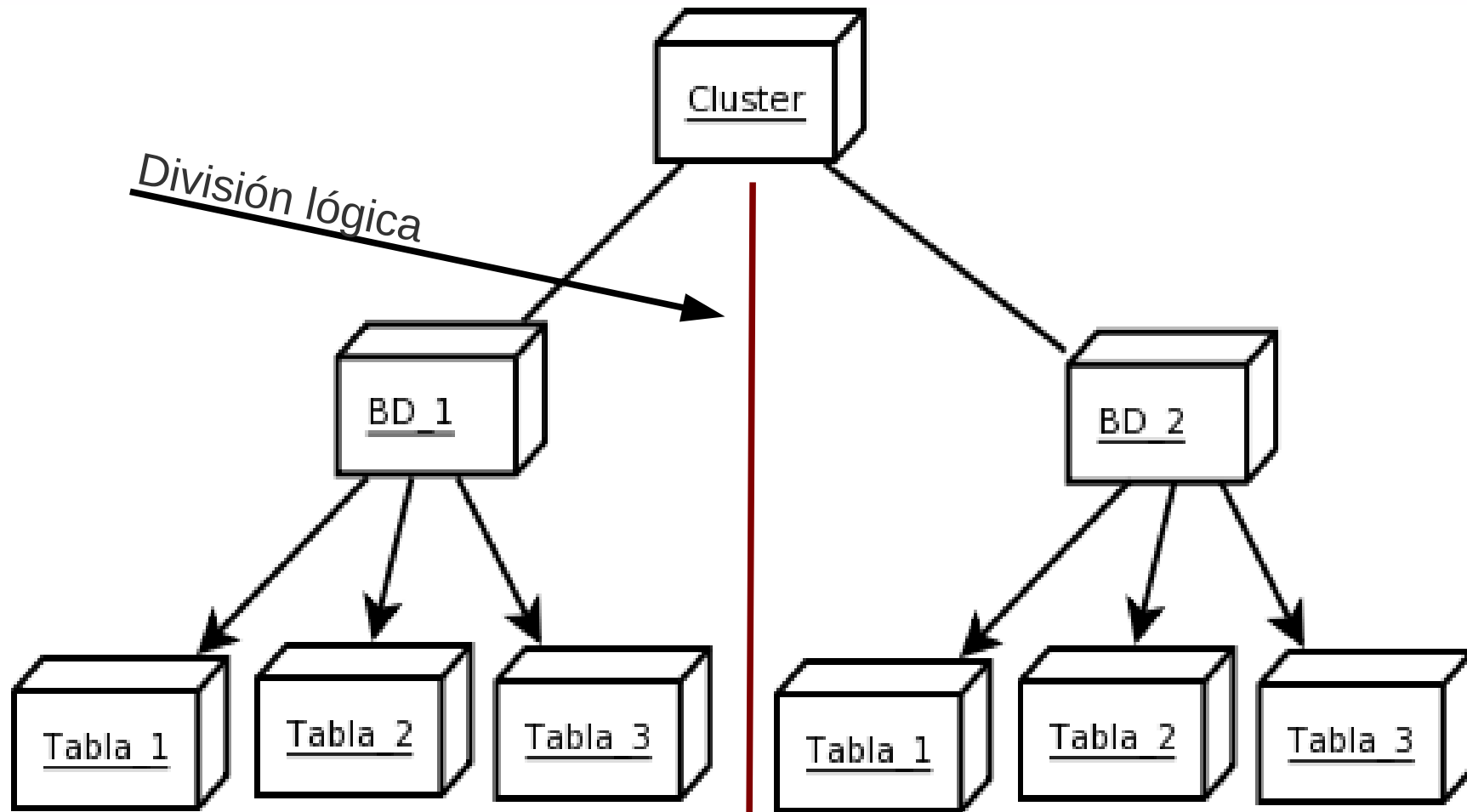
Características

- **Objeto Relacional**
- **Desarrollada implementando la mayoría del SQL92**
- **Open Source**
- **Soporte para Transacciones**
- **Integridad Referencial**
- **Múltiples lenguajes procedurales**
- **APIs para múltiples clientes**
- **Tipos de datos únicos**
- **Extensible**

- **Backend**
 - Sistema que reside en el servidor.
 - Código principalmente en C.
 - No se usa malloc, más bien rutinas implementadas por hackers de Postgres. Ej: `memory_context()`
- **Frontend**
 - Procesos que se realizan desde el (los) cliente(s).
 - En Postgres hay un terminal interactivo por defecto.
 - Hay una librería llamada `libpq` para el frontend
 - El nombre del terminal es *psql*
 - Puede usar hebras (`safe-threads`)
 - Los bindings son portados principalmente usando `libpq`:
 - `psycopg`, `libpq++`

- **Cluster**
 - **Colección de bases de datos.**
 - Cada *cluster* reside dentro de un directorio
 - Cada *cluster* es manejado por un postmaster
- **Postmaster**
 - **Proceso encargado de escuchar conexiones**
 - Por cada cliente se realiza un `fork()`
 - Cada conexión es un nuevo proceso
 - No hay hebras (no threads!)
- **Schema (Esquema)**
 - División lógica dentro de una DB
 - Por defecto se crea un esquema llamado “public”
 - Agregado en v7.3 de postgres

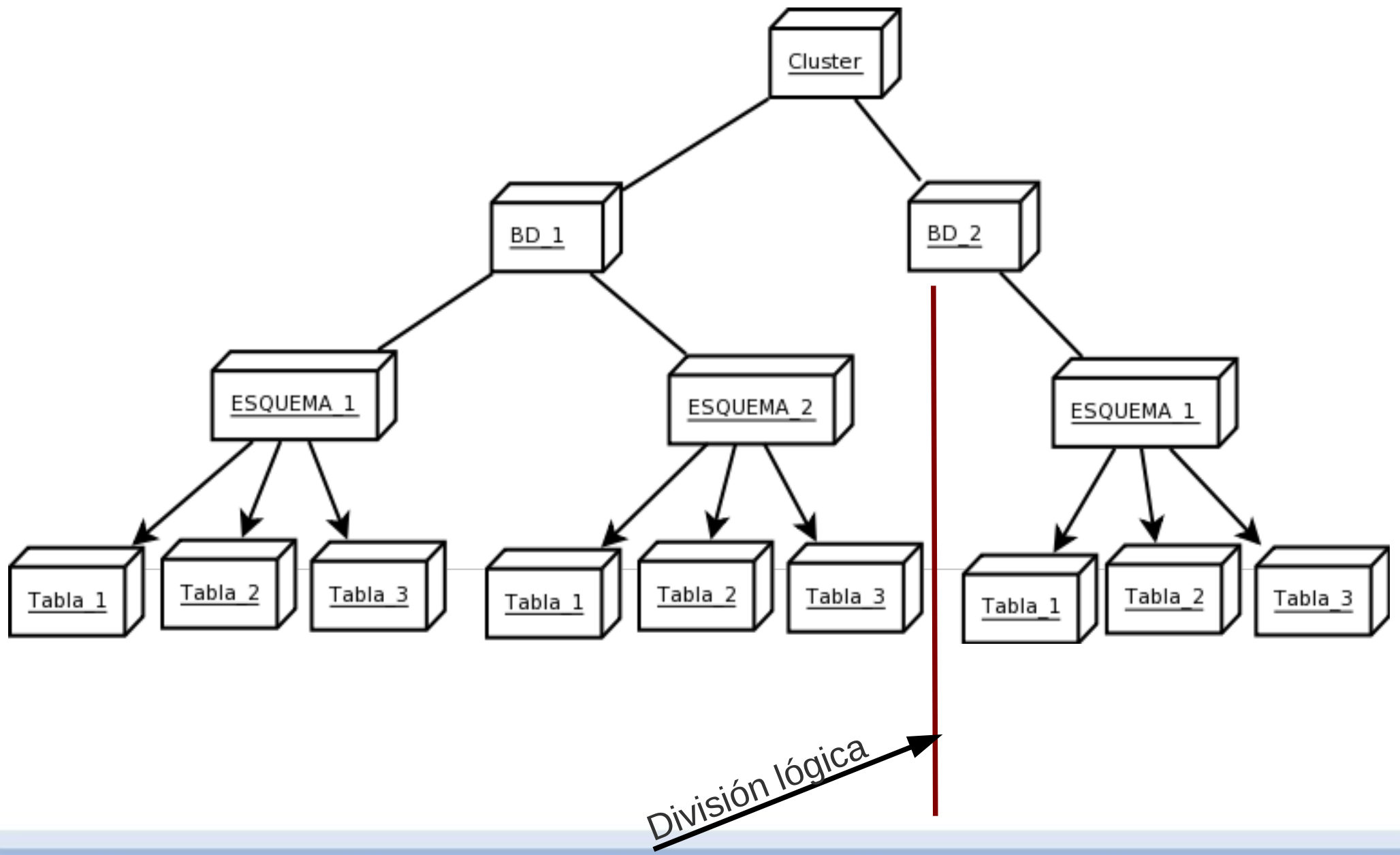
Características – Estructura - PostgreSQL versión < 7.3



SELECT b2_1.tabla_1.campo1

no funciona?! :-)

Características – Estructura - PostgreSQL >= 7.3 (7.4 ya funcionaba bien)



Características – Estructura - PostgreSQL >= 7.3 (7.4 ya funcionaba bien)

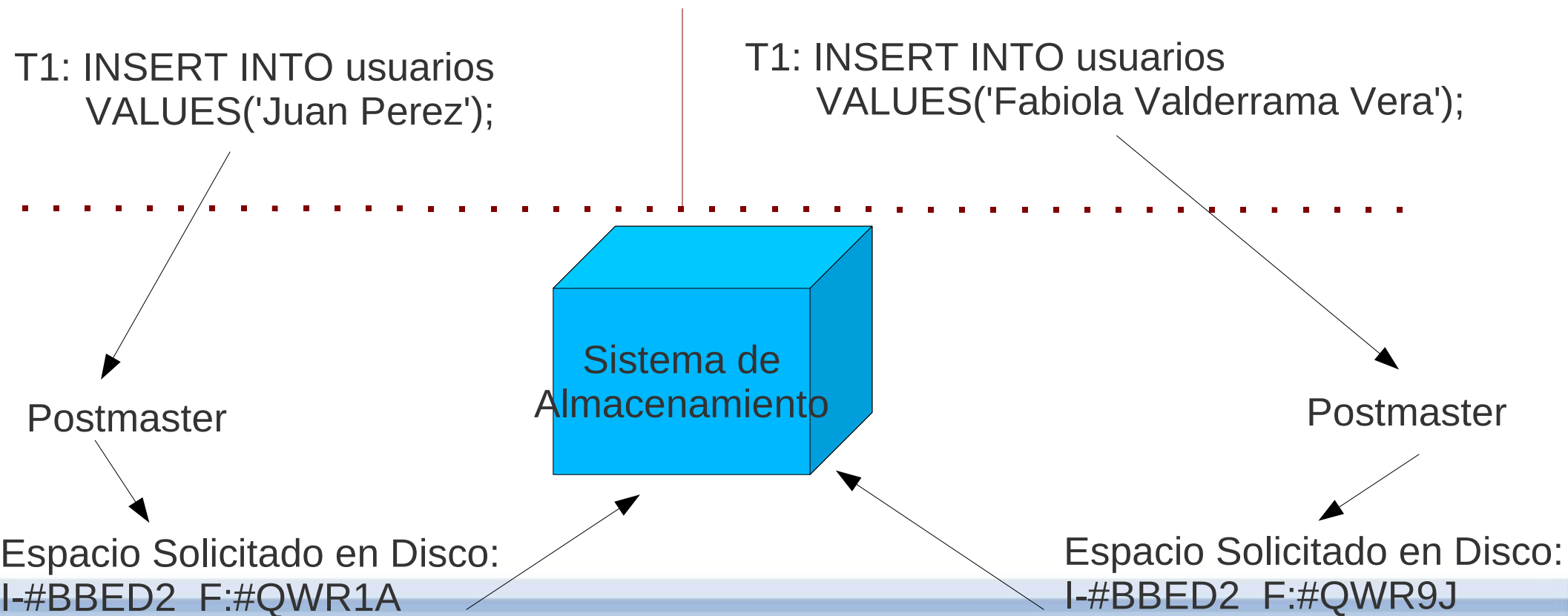
```
mario@localhost:~$ createdb test
CREATE DATABASE
mario@localhost:~$ psql test
Welcome to psql 8.2.5, the PostgreSQL interactive terminal.
```

```
Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with psql commands
      \g or terminate with semicolon to execute query
      \q to quit
```

```
test=# CREATE SCHEMA uno;
test=# CREATE SCHEMA dos;
test=# CREATE TABLE uno.tabla1 (
       numero INTEGER PRIMARY KEY
);
test=# CREATE TABLE dos.tabla1 (
       numero INTEGER PRIMARY KEY REFERENCES uno.tabla1
);
```

▪ Problema:

- ¿Cómo permito la concurrencia para múltiples usuarios?
- Ej: Si hay más de un usuario en la misma tabla realizando modificaciones, puedo sufrir corrupción de datos



- **Posibles soluciones:**
 - Bloquear la tabla completa con acceso limitado
 - Bloquear sólo algunos campos
- **Tengo un nuevo problema:**
 - Rendimiento es pésimo
 - Usuarios deben esperar a que el lock desaparezca
 - Aún para lecturas! (SQLServer ?)
- **Solución en PostgreSQL:**
 - **MVCC (Multi-versioning and Locking)**
 - Crear “copias” dentro de una transacción
 - No bloquear HASTA que sea necesario
 - Las lecturas no son bloqueadas
 - El bloqueo se hace en un *COMMIT*
 - Si hay *ROLLBACK* el resto de las transacciones nunca “sufrieron” pérdida de rendimiento ni esperas en el acceso a los datos

- **ROLLBACK**
 - Transacción es abortada
 - Posibles cambios ignorados
- **COMMIT**
 - Los datos son modificados físicamente en el *Sistema de Almacenamiento* (Disco duro, Fiber Channel,...)
 - El UPDATE es un UPDATE+INSERT
 - DELETE es un UPDATE
- **Al modificar el sistema de archivo (INSERT, UPDATE o DELETE) las tuplas afectadas se “marcan” como obsoletas (tuplas muertas)**
- **Bueno, ¿Y ese espacio, no se utiliza más?**
 - VACUUM

- **Antes de escribir al sistema de archivo, es necesario saber:**
 - ¿Cuántos bytes voy a escribir?
 - ¿Cuales son los bloques físicos disponibles?
 - 1ero: FSM (Free Space Map)
 - 2do: Sistema de Archivo
- **FSM**
 - Tuplas muertas
 - Tabla con los bloques disponibles
- **VACUUM**
 - Revisa cuales son las tuplas muertas en el sistema de archivos
 - VACUUM ANALIZE es nuestro amigo :-)
 - VACUUM FULL no tanto :-)
- **Autovacuum (Escrito por Alvaro Herrera)**

- **TCP/IP**
 - **IPV4, IPV6**
 - **Socket directo**
 - **ODBC, JDBC**
- **Socket UNIX**
- **Se restringen según archivo pg_hba.conf**
- **Un trozo de código de ejemplo:**

```
# This file controls: which hosts are allowed to connect, how clients
# are authenticated, which PostgreSQL user names they can use, which
# databases they can access. Records take one of these forms:
#
# local    DATABASE USER METHOD [OPTION]
# host     DATABASE USER CIDR-ADDRESS METHOD [OPTION]
# hostssl  DATABASE USER CIDR-ADDRESS METHOD [OPTION]
# hostnossl DATABASE USER CIDR-ADDRESS METHOD [OPTION]
```

Sin importar el método, veremos como usar psql

▪ La ayuda de *psql*

```
mario@localhost:~$ psql --help
```

This is psql 8.2.5, the PostgreSQL interactive terminal.

Usage:

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

General options:

- d DBNAME specify database name to connect to (default: "mario")
- c COMMAND run only single command (SQL or internal) and exit
- f FILENAME execute commands from file, then exit
- 1 ("one") execute command file as a single transaction
- l list available databases, then exit
- v NAME=VALUE set psql variable NAME to VALUE
- X do not read startup file (~/.psqlrc)
- help show this help, then exit
- version output version information, then exit

- **Informal**
 - [#postgresql #postgresql-es](http://irc.freenode.org)
 - www.postgresql.org
 - <http://archives.postgresql.org/pgsql-es-ayuda/>
 - pgsql-es-ayuda@postgresql.org
- **Comercial**
 - **CommandPrompt (CMD)**
 - www.commandprompt.com
 - **EnterpriseDB (EDB)**
 - **EDB** www.enterprisedb.com
- **O comprar soporte a alguna distribución:**
 - www.mandriva.com
 - www.redhat.com
 - **Ubuntu ??**

PostgreSQL: Introducción



Mario González T
mgonzalez@mandriva.com