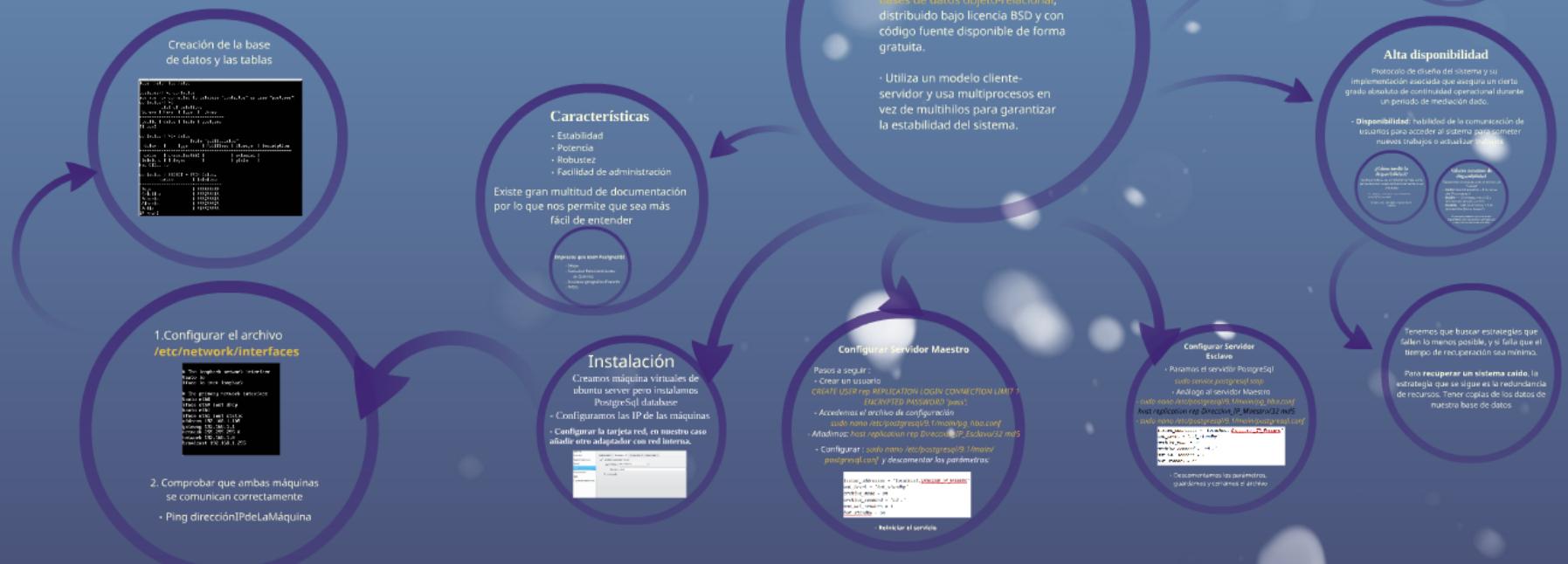
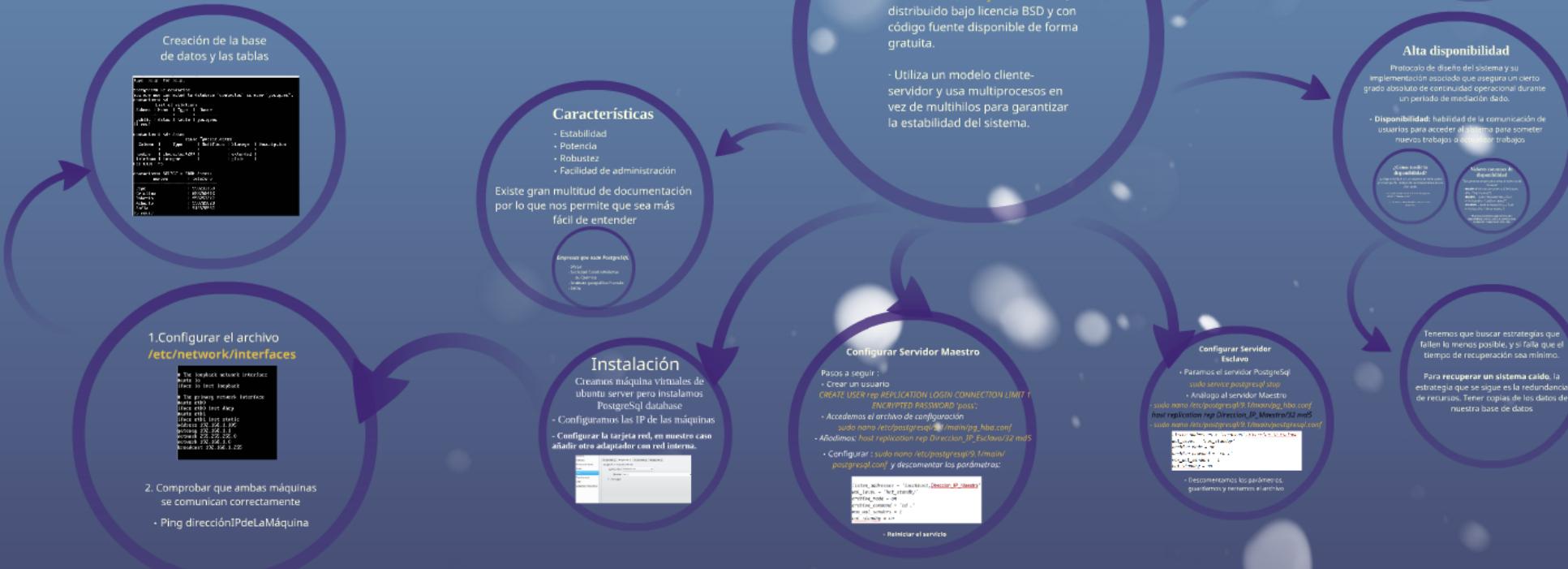


High Availability, Load Balancing y Replication con PostgreSQL



High Availability, Load Balancing y Replication con PostgreSQL



Prezi

PostgreSQL

- Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con código fuente disponible de forma gratuita.
- Utiliza un modelo cliente-servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema.

Disponible para todas las plataformas; Linux, Unix, Windows

Es manejado por una comunidad de desarrolladores, se denomina PGDG

Soportar muchos tipos de lenguajes, así como tiene una extensa biblioteca de interfaces por lo que podemos interpretar varios lenguajes

Características

- Estabilidad
- Potencia
- Robustez
- Facilidad de administración

Existe gran multitud de documentación por lo que nos permite que sea más fácil de entender

Empresas que usan PostgreSQL

- Skype
- Sociedad Estadounidense de Química
- Instituto geográfico Francés
- IMDb

Empresas que usan PostgreSQL

- Skype
- Sociedad Estadounidense
de Química
- Instituto geográfico Francés
- IMDb

Alta disponibilidad

Protocolo de diseño del sistema y su implementación asociada que asegura un cierto grado absoluto de continuidad operacional durante un periodo de mediación dado.

- **Disponibilidad:** habilidad de la comunicación de usuarios para acceder al sistema para someter nuevos trabajos o actualizar trabajos

¿Cómo medir la disponibilidad?

La disponibilidad de un sistema se mide como porcentaje del tiempo de funcionamiento en un año dado.

- Se puede medir en términos de "tiempo en activo" y "tiempo caldo"
- El tiempo de inactividad se conoce como *downtime*

Valores comunes de disponibilidad

Típicamente enunciado como el número de "nueves"

- **99,9%**=43.8 minutos/mes u 8,76 horas/año ("tres nueves")
- **99,99%** = 4.38 minutos/mes o 52.6 minutos/año ("cuatro nueves")
- **99,999%** = 0.44 minutos/mes o 5.26 minutos/año ("cinco nueves")

El principal problema que tiene la alta disponibilidad son las caídas de servicios y los tiempos de recuperación ante ellos



¿Cómo medir la disponibilidad?

La disponibilidad de un sistema se mide como porcentaje del tiempo de funcionamiento en un año dado.

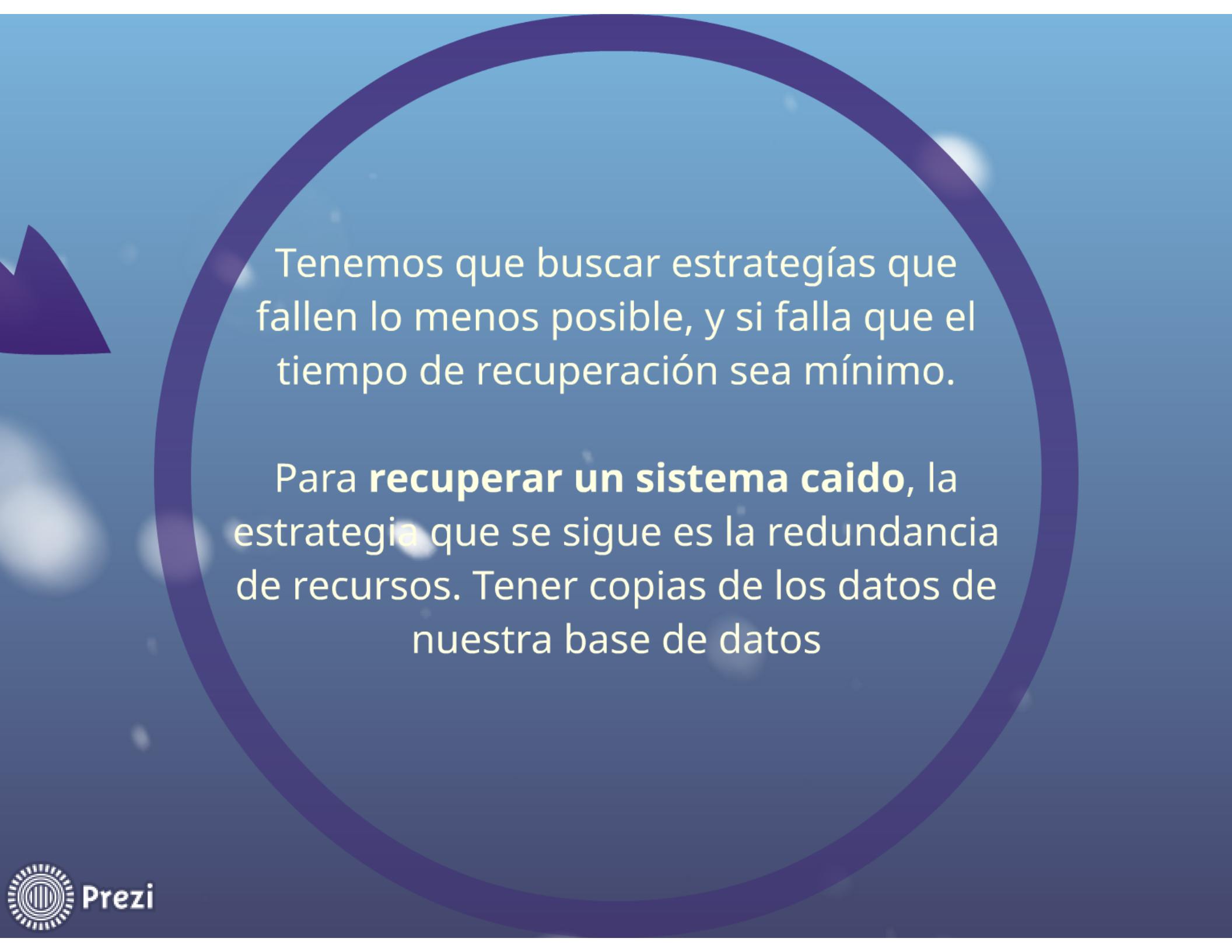
- Se puede medir en términos de "tiempo en activo" y "tiempo caído"
- El tiempo de inactividad se conoce como *downtime*

Valores comunes de disponibilidad

Típicamente enunciado como el número de "nueves"

- **99,9%**=43.8 minutos/mes u 8,76 horas/año ("tres nueves")
- **99,99%** = 4.38 minutos/mes o 52.6 minutos/año ("cuatro nueves")
- **99,999%** = 0.44 minutos/mes o 5.26 minutos/año ("cinco nueves")

El principal problema que tiene la alta disponibilidad son las caídas de servicios y los tiempos de recuperación ante ellos



Tenemos que buscar estrategías que fallen lo menos posible, y si falla que el tiempo de recuperación sea mínimo.

Para **recuperar un sistema caido**, la estrategia que se sigue es la redundancia de recursos. Tener copias de los datos de nuestra base de datos

Replicación en PostgreSQL

Es el proceso por el cual copiamos y mantenemos actualizados los datos en varios nodos de las bases de datos.

PROBLEMA: no siempre las réplicas tiene el mismo tamaño que la base original.

Los tipos de Replicación son:

- Replicación de transacciones
- Replicación por bitácoras
- Replicación por formateos específicos

Balanceo de Cargas en PostgreSQL

Para el balanceo usamos las consultas distribuidas . En PostgreSQL se llama Multi-Server Parallel Query Execution.

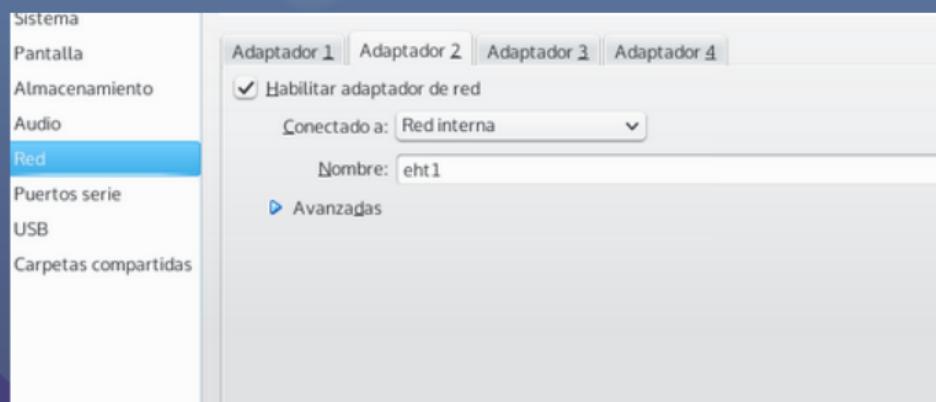
El objetivo es evitar la sobrecarga. Cuando no tenemos servidores idénticos haremos balanceo en las capacidades que cada servidor tenga.

Por ejemplo si tenemos dos servidores y cada uno con distinto numero de procesadores. El de mayor número tendrá más carga.

Instalación

Creamos máquina virtuales de
ubuntu server pero instalamos
PostgreSql database

- Configuramos las IP de las máquinas
- **Configurar la tarjeta red, en nuestro caso añadir otro adaptador con red interna.**



1. Configurar el archivo **/etc/network/interfaces**

```
# The loopback network interface
#auto lo
iface lo inet loopback

# The primary network interface
#auto eth0
iface eth0 inet dhcp
#auto eth1
iface eth1 inet static
address 192.168.1.105
gateway 192.168.1.1
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
```

2. Comprobar que ambas máquinas se comunican correctamente

- Ping direcciónIPdeLaMáquina

Creación de la base de datos y las tablas

```
Type "help" for help.

postgres=# \c contactos
You are now connected to database "contactos" as user "postgres".
contactos=# \d
      List of relations
 Schema | Name | Type | Owner
-----+-----+-----+
 public | datos | table | postgres
(1 row)

contactos=# \d+ datos
           Table "public.datos"
 Column | Type | Modifiers | Storage | Description
-----+-----+-----+-----+
 nombre | character(20) |          | extended |
 telefono | integer |          | plain    |
Has OIDs: no

contactos=# SELECT * FROM datos;
     nombre   | telefono
-----+-----+
 Pepe        | 958233459
 Cristina    | 658765412
 Roberto     | 958753212
 Alberto     | 958789878
 Sofia       | 912678562
(5 rows)
```



Configurar Servidor Maestro

Pasos a seguir :

- Crear un usuario

*CREATE USER rep REPLICATION LOGIN CONNECTION LIMIT 1
ENCRYPTED PASSWORD 'pass';*

- Accedemos el archivo de configuración

sudo nano /etc/postgresql/9.1/main/pg_hba.conf

- Añadimos: *host replication rep Direccion_IP_Esclavo/32 md5*

- Configurar : *sudo nano /etc/postgresql/9.1/main/postgresql.conf* y descomentar los parámetros:

```
listen_addresses = 'localhost,Direccion_IP_Maestro'  
wal_level = 'hot_standby'  
archive_mode = on  
archive_command = 'cd .'  
max_wal_senders = 1  
hot_standby = on
```

- Reiniciar el servicio



Configurar Servidor Esclavo

- Paramos el servidor PostgreSQL

sudo service postgresql stop

- Análogo al servidor Maestro

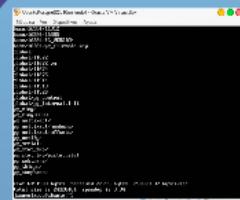
- *sudo nano /etc/postgresql/9.1/main/pg_hba.conf*

host replication rep Direccion_IP_Maestro/32 md5

- *sudo nano /etc/postgresql/9.1/main/postgresql.conf*

```
listen_addresses = 'localhost,Direccion_IP_Esclavo'  
wal_level = 'hot_standby'  
archive_mode = on  
archive_command = 'cd .'  
max_wal_senders = 1  
hot_standby = on
```

- Descomentamos los parámetros, guardamos y cerramos el archivo



Replicar la base de datos inicial

- Escribimos las siguientes órdenes

```
psql -c "SELECT pg_start_backup('initial_backup');"
```

```
rsync -cva --inplace --exclude=*pg_xlog* /var/lib/postgresql/9.1/main/ Direccion_IP_Eslavo:/var/lib/postgresql/9.1/main/
```

Ponemos `SELECT pg_stop_backup();`

UbuntuPostgreSQL [Corriendo] - Oracle VM VirtualBox

Máquina Ver Dispositivos Ayuda

```
base/16384/11912
base/16384/16385
base/16384/PG_VERSION
base/16384/pg_filenode.map
global/
global/11672
global/11672_vm
global/11674
global/11675
global/11914
global/11919
global/11920
global/pg_control
global/pg_internal.init
pg_clog/
pg_clog/0000
pg_multixact/
pg_multixact/members/
pg_multixact/offsets/
pg_notify/
pg_serial/
pg_stat_tmp/
pg_stat_tmp/pgstat.stat
pg_subtrans/
pg_tblspc/
pg_twophase/

sent 6151900 bytes received 22795 bytes 252028.37 bytes/sec
total size is 24328464 speedup is 3.94
juanuelasco@ubuntu:~$
```

Ctrl Derecho

Terminando la configuración

Por último deberemos crear un archivo de recovery o de recuperación. Para ello escribimos:

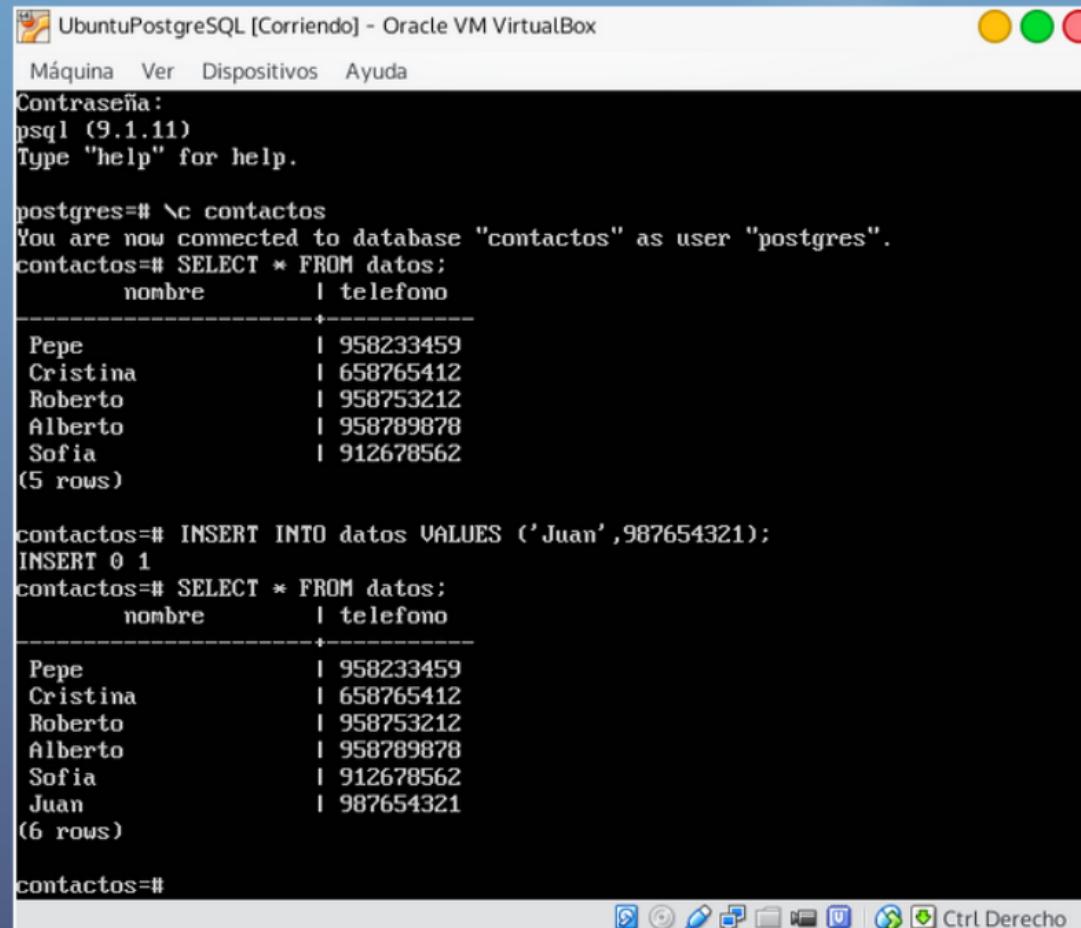
```
sudo nano /var/lib/postgresql/9.1/main/recovery.conf
```

Y escribimos dentro lo siguiente

```
standby_mode='on'  
primary_conninfo='host=192.168.1.106 port=5432  
user=juanvelasco password=tucontraseña'  
trigger_file='/tmp/postgresql.trigger.5432'
```



MAESTRO



UbuntuPostgreSQL [Corriendo] - Oracle VM VirtualBox

Máquina Ver Dispositivos Ayuda

Contraseña:

```
psql (9.1.11)
Type "help" for help.
```

```
postgres=# \c contactos
You are now connected to database "contactos" as user "postgres".
contactos=# SELECT * FROM datos;
    nombre        | telefono
+-----+-----+
| Pepe      | 958233459
| Cristina   | 658765412
| Roberto    | 958753212
| Alberto    | 958789878
| Sofia      | 912678562
(5 rows)

contactos=# INSERT INTO datos VALUES ('Juan',987654321);
INSERT 0 1
contactos=# SELECT * FROM datos;
    nombre        | telefono
+-----+-----+
| Pepe      | 958233459
| Cristina   | 658765412
| Roberto    | 958753212
| Alberto    | 958789878
| Sofia      | 912678562
| Juan       | 987654321
(6 rows)

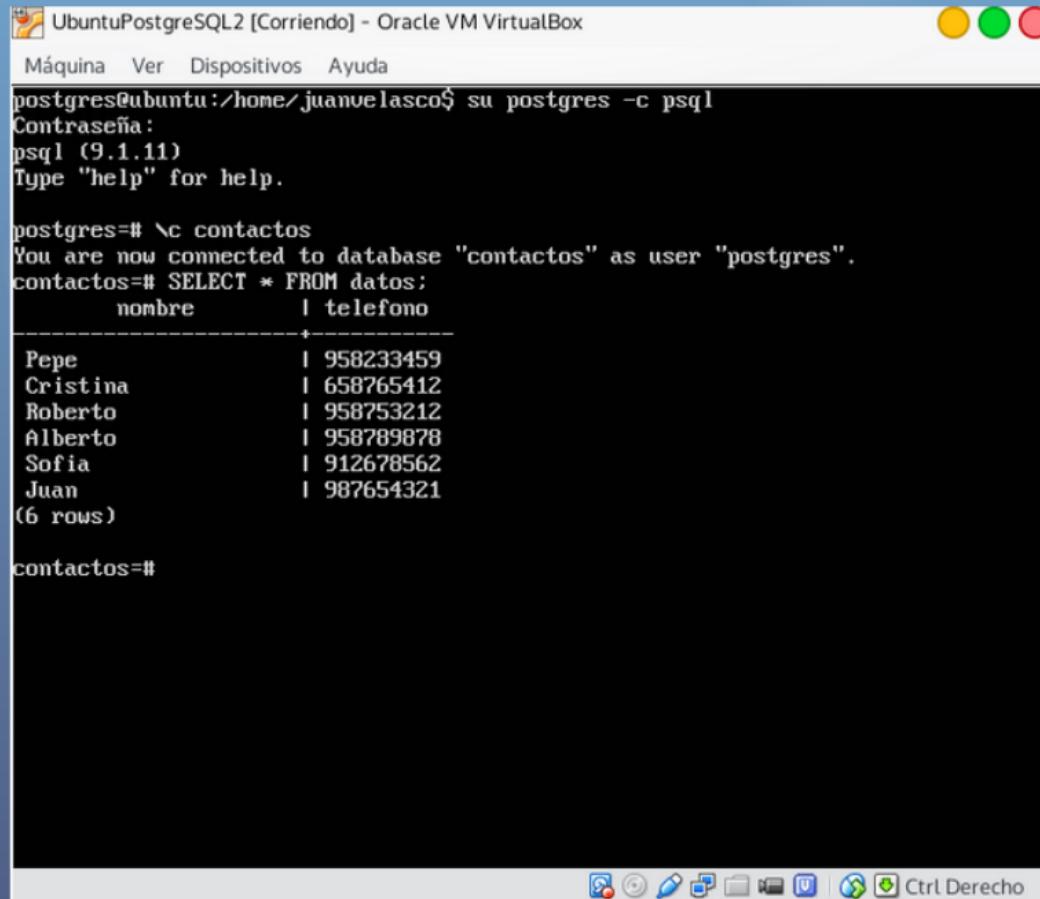
contactos=#

```

ESCLAVO

Ctrl Derecho

ESCLAVO



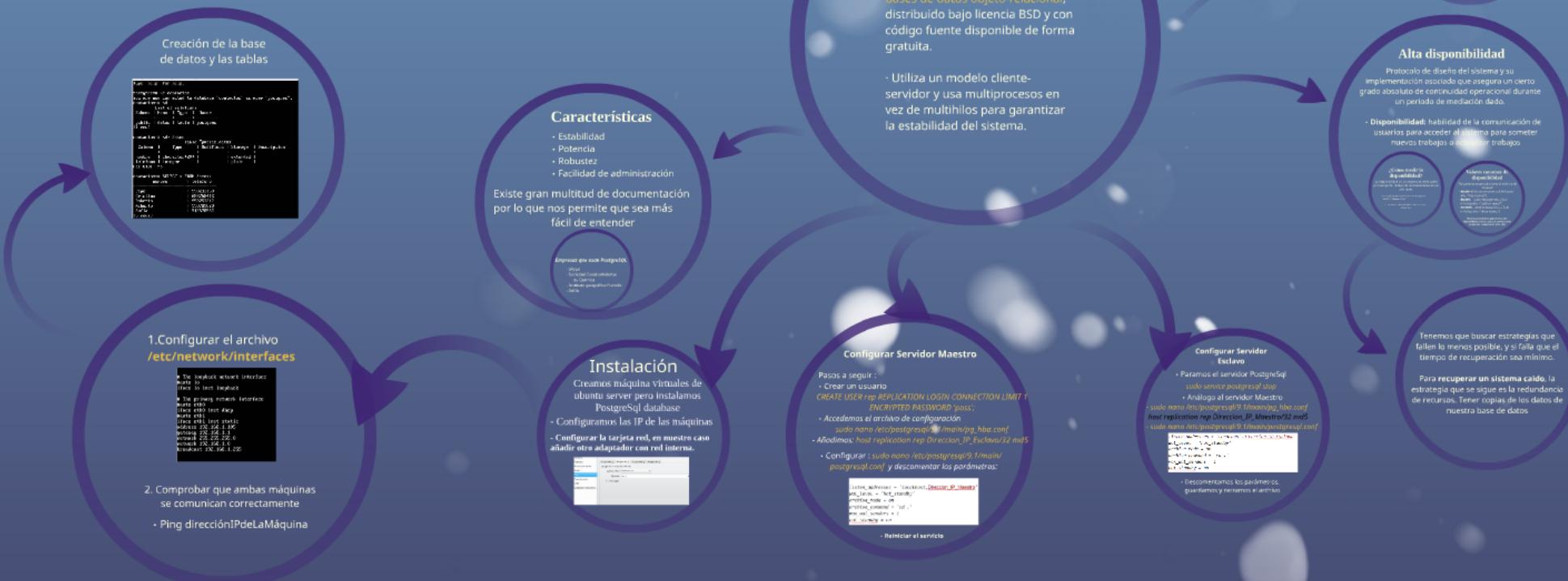
The screenshot shows a terminal window titled "UbuntuPostgreSQL2 [Corriendo] - Oracle VM VirtualBox". The window contains the following PostgreSQL session:

```
postgres@ubuntu:/home/juanuelasco$ su postgres -c psql
Contraseña:
psql (9.1.11)
Type "help" for help.

postgres=# \c contactos
You are now connected to database "contactos" as user "postgres".
contactos=# SELECT * FROM datos;
   nombre    | telefono
+-----+-----+
  Pepe      | 958233459
Cristina   | 658765412
Roberto    | 958753212
Alberto    | 958789878
Sofia      | 912678562
Juan       | 987654321
(6 rows)

contactos=#
```

High Availability, Load Balancing y Replication con PostgreSQL



Prezi