

Bloque 3 Rubrica 1 :

Configurar postgresQL para que admita conexiones remotas(dual)

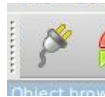
Configurar que pueda acceder conexiones remotas

instalar pgadmin

→ **sudo apt-get install pgadmin3**

Arrancamos **PGAdmin**

Creamos una nueva conexión,



New Server Registration

Properties | SSL | SSH Tunnel | Advanced

Name: local

Host: localhost

Port: 5432

Service:

Maintenance DB: postgres

Username: postgres

Password:

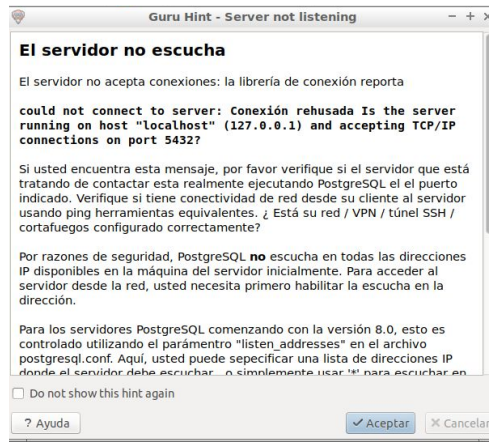
Store password: ☒

Colour:

Group: Servers

? Ayuda Aceptar Cancelar

nos dice que el servidor no escucha.



Lo que tengo que hacer es **asegurarnos de que no esté apagado postgres**

→ **sudo service postgresql start**

Ahora sale otro error:

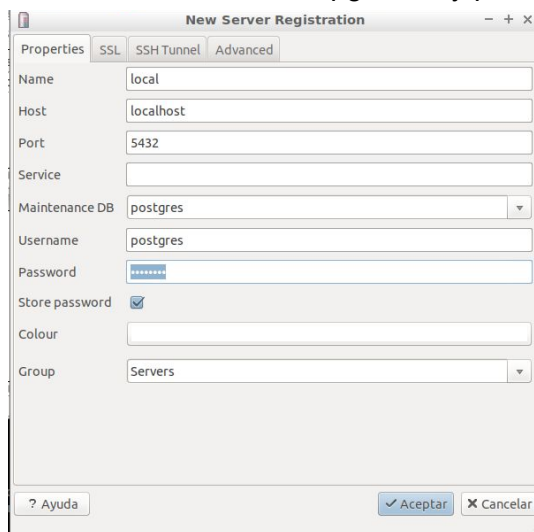


Aquí el problema es que **pgadmin no se conecta a un postgres sin contraseña**
Para eso, tenemos que cambiarle la contraseña.

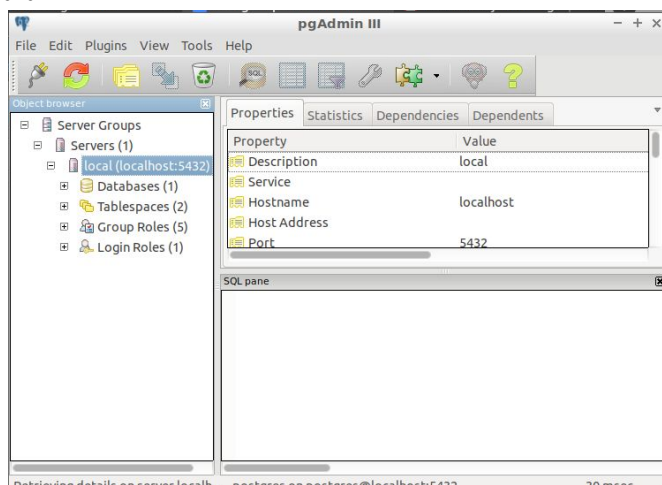
Cambiar contraseña a postgres

- **sudo -u postgres psql postgres**
- **\password postgres**
- introducimos la contraseña (en mi caso "postgres")
- **\q** (para salir)

una vez hecho, vamos a pgadmin y ponemos la contraseña



y ya tendríamos la conexión.




Cuando se instala postgres por defecto no permite conexiones remotas. Esto quiere decir que si deseas conectarte desde otra máquina diferente no es posible. Si se está desarrollando una aplicación, por lo general, es necesario el uso de conexiones remotas.

A continuación, se explican los simples pasos que se deben ejecutar para cumplir con este requerimiento:

Paso 1.

Editar el archivo **postgresql.conf** que se encuentra en la ruta **/etc/postgresql/10/main**

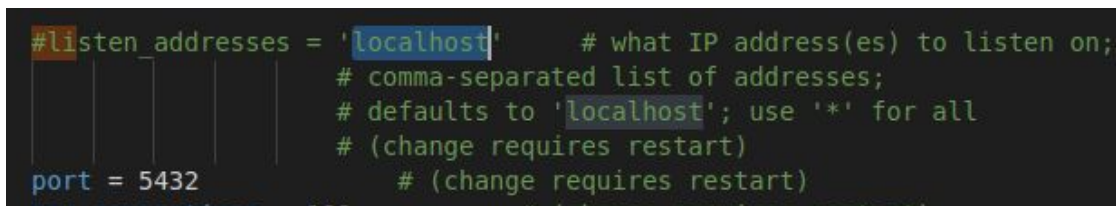
Vamos a editar el archivo (→ code postgresql.conf)



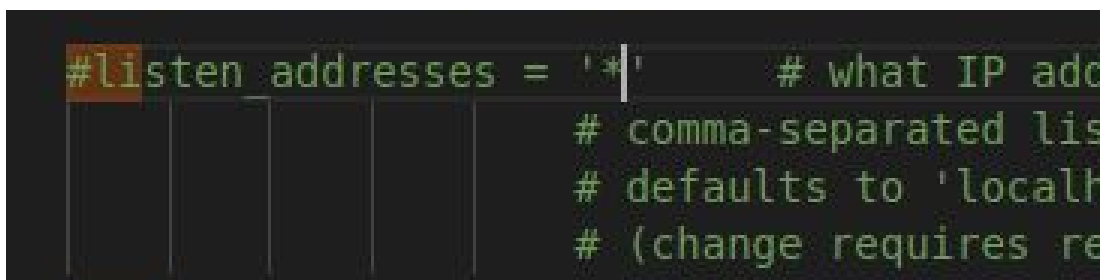
```
jonathan@jonathan: /etc/postgresql/10/main
jonathan@jonathan: ~
jonathan@jonathan: /etc/postgresql/10/main 150x36
~ cd /etc/postgresql/10/main
@ /etc/postgresql/10/main ls
conf.d environment pg_ctl.conf pg_hba.conf pg_ident.conf postgresql.conf start.conf
@ /etc/postgresql/10/main code postgresql.conf
```

Editamos la línea **#listen_addresses = 'localhost'**

Y la cambiamos por **#listen_addresses = '*'**

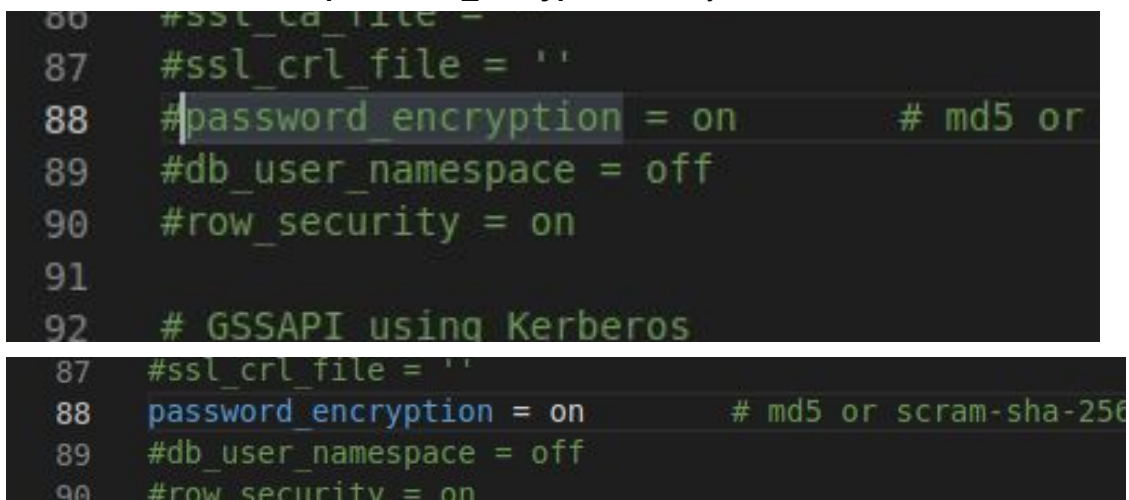


```
#listen_addresses = 'localhost' # what IP address(es) to listen on;
# comma-separated list of addresses;
# defaults to 'localhost'; use '*' for all
# (change requires restart)
port = 5432 # (change requires restart)
```



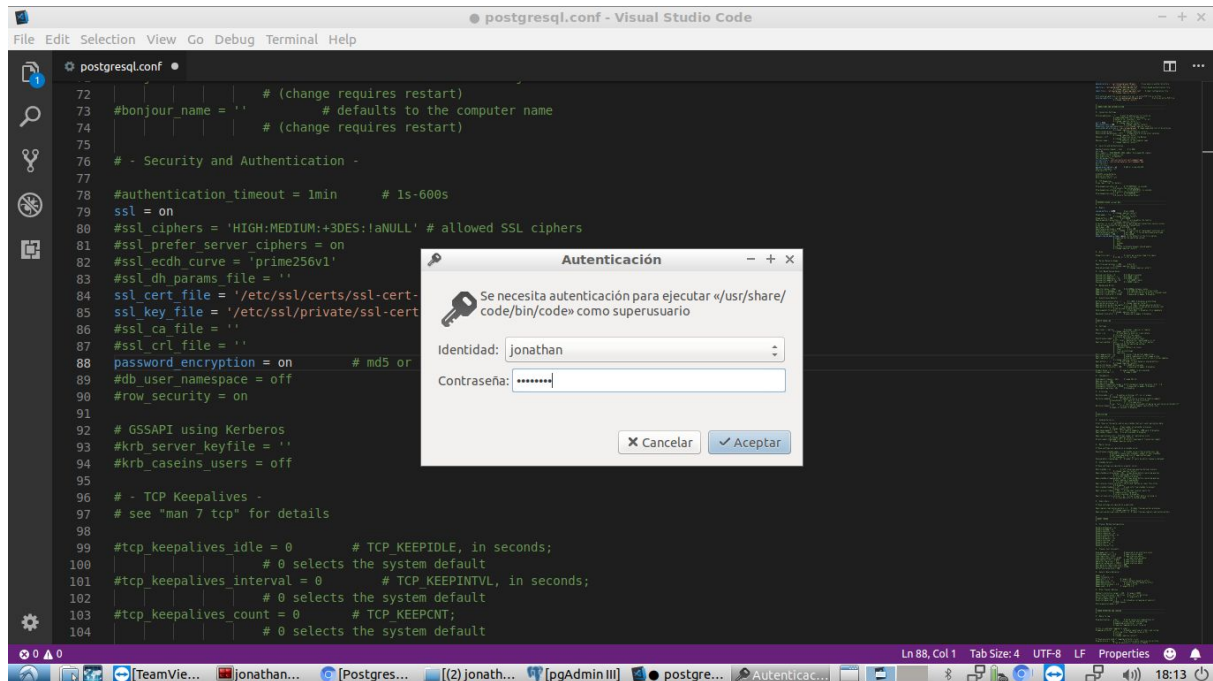
```
#listen_addresses = '*' # what IP address(es) to listen on;
# comma-separated list of addresses;
# defaults to 'localhost'; use '*' for all
# (change requires restart)
```

Ahora vamos a la línea **#password_encryption = on** y la descomentamos

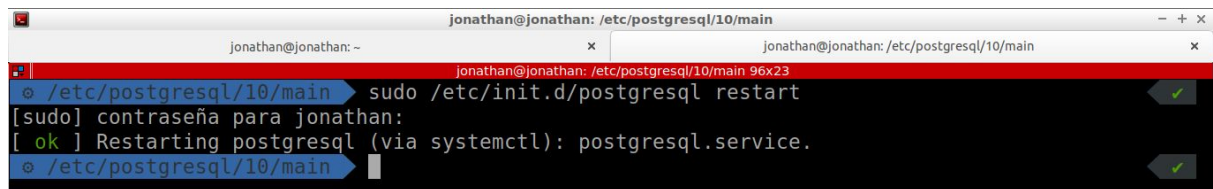


```
86 #ssl_ca_file = ''
87 #ssl_crl_file = ''
88 #password_encryption = on # md5 or scram-sha-256
89 #db_user_namespace = off
90 #row_security = on
91
92 # GSSAPI using Kerberos
87 #ssl_crl_file = ''
88 password_encryption = on # md5 or scram-sha-256
89 #db_user_namespace = off
90 #row_security = on
```

Guardamos el archivo



Y reiniciamos postgres



Paso 2

Se debe modificar lista de acceso, la cual permite establecer relaciones de confianza para ciertos equipos y redes.

Nos vamos a `/etc/postgresql/10/main` y editamos `pg_hba.conf` (`sudo vim pg_hba.conf`)

```
jonathan@jonathan: /etc/postgresql/10/main
jonathan@jonathan: /etc/postgresql/10/main 75x19
~ ➔ cd /etc/postgresql/10/main
* /etc/postgresql/10/main ➔ ls
conf.d      pg_ctl.conf  pg_ident.conf  start.conf
environment pg_hba.conf  postgresql.conf
* /etc/postgresql/10/main ➔ sudo vim pg_hba.conf
```

Al final del archivi ponemos **host all all 0.0.0.0 0.0.0.0 md5**

```
sudo vim pg_hba.conf
sudo vim pg_hba.conf 84x21
1 pg_hba.conf
17 #
16 # Database administrative login by Unix domain socket
15 local all postgres peer
14
13 # TYPE DATABASE USER ADDRESS METHOD
12
11 # "local" is for Unix domain socket connections only
10 local all all peer
9 # IPv4 local connections:
8 host all all 127.0.0.1/32 md5
7 # IPv6 local connections:
6 host all all ::1/128 md5
5 # Allow replication connections from localhost, by a user with the
4 # replication privilege.
3 local replication all peer
2 host replication all 127.0.0.1/32 md5
1 host replication all ::1/128 md5
100 host all all 0.0.0.0 0.0.0.0 md5
:wg
```

Hacemos un restart

```
jonathan@jonathan: /etc/postgresql/10/main
jonathan@jonathan: /etc/postgresql/10/main 84x21
* /etc/postgresql/10/main ➔ sudo /etc/init.d/postgresql restart
[ ok ] Restarting postgresql (via systemctl): postgresql.service.
* /etc/postgresql/10/main ➔
```