

Instalación de herramientas básicas para la asignatura (S.O. Ubuntu Server 18.04)

Doc V1.0

Instalación de nmap	2
Instalación de la base de datos POSTGRESQL 11	4
Ejemplo básico	9
Instalación de PostGIS sobre Postgresql 11	11
Instalación de pgAdmin4	13
Estructura básica de una sentencia sql	17
Uso del ambiente de trabajo	19

Instalación de nmap

Herramienta que permite verificar los puertos que tienes abiertos en tu sistema. Hacer esta verificación es importante para conocer la disponibilidad de puerto en tu máquina. Las siguientes instrucciones se ejecutan en la línea de comando.

- Actualizamos los paquetes de software de ubuntu

sudo apt-get update

- Instalar aplicación (probablemente pregunte si desea instalarlo, Y + Enter)

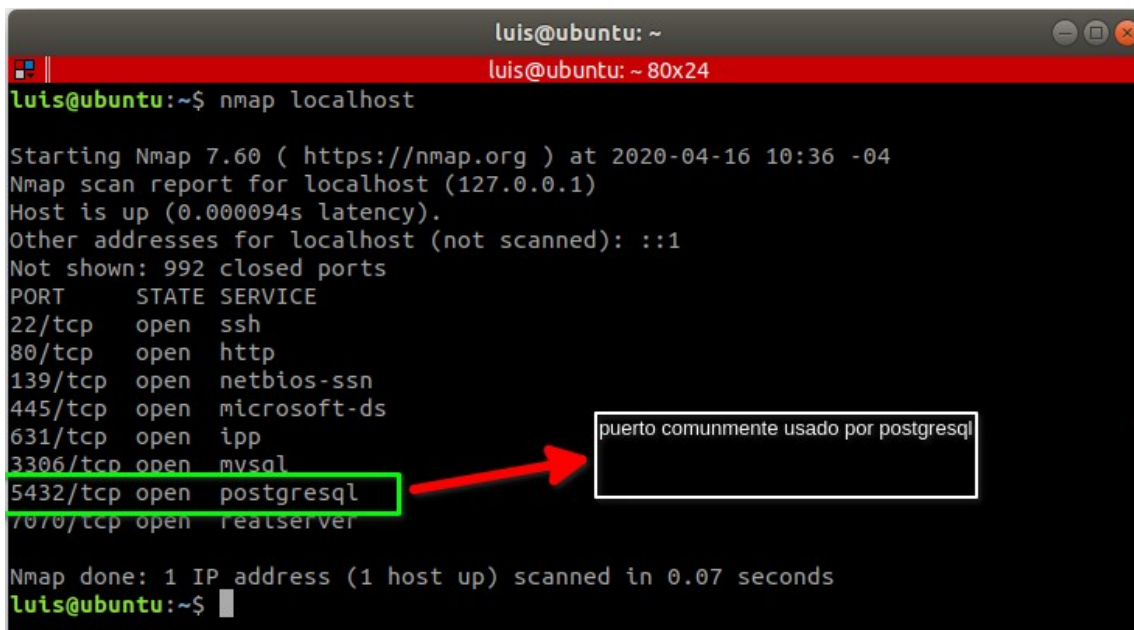
sudo apt-get install nmap

- Verificar versión instalada

nmap --version

- Uso básico

nmap localhost



```
luis@ubuntu: ~  
luis@ubuntu: ~ 80x24  
luis@ubuntu:~$ nmap localhost  
  
Starting Nmap 7.60 ( https://nmap.org ) at 2020-04-16 10:36 -04  
Nmap scan report for localhost (127.0.0.1)  
Host is up (0.000094s latency).  
Other addresses for localhost (not scanned): ::1  
Not shown: 992 closed ports  
PORT      STATE SERVICE  
22/tcp    open  ssh  
80/tcp    open  http  
139/tcp   open  netbios-ssn  
445/tcp   open  microsoft-ds  
631/tcp   open  ipp  
3306/tcp  open  mysql  
5432/tcp  open  postgresql  
7070/tcp  open  realserver  
  
Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds  
luis@ubuntu:~$
```

- También puedes verificar los puerto abiertos otros servidores, ejemplo en google.cl

nmap google.cl

```
luis@ubuntu: ~  
luis@ubuntu: ~ 80x24  
luis@ubuntu:~$ nmap google.cl  
  
Starting Nmap 7.60 ( https://nmap.org ) at 2020-04-16 10:33 -04  
Nmap scan report for google.cl (172.217.192.94)  
Host is up (0.032s latency).  
Other addresses for google.cl (not scanned): 2800:3f0:4003:c02::5e  
Not shown: 998 filtered ports  
PORT      STATE SERVICE  
80/tcp    open  http  
443/tcp   open  https  
  
Nmap done: 1 IP address (1 host up) scanned in 12.90 seconds  
luis@ubuntu:~$ ^C
```

Verificar un puerto en especifico, en este ejemplo el 3000, donde no tengo ningún sistema corriendo

nmap -p 3000 localhost

```
luis@ubuntu: ~  
luis@ubuntu: ~ 80x14  
luis@ubuntu:~$ nmap -p 3000 localhost  
  
Starting Nmap 7.60 ( https://nmap.org ) at 2020-04-16 10:49 -04  
Nmap scan report for localhost (127.0.0.1)  
Host is up (0.000089s latency).  
Other addresses for localhost (not scanned): ::1  
  
PORT      STATE SERVICE  
3000/tcp  closed ppp  
  
Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds  
luis@ubuntu:~$
```

Verificar un puerto en especifico, en este ejemplo el 7070, donde tengo una aplicación web corriendo

nmap -p 7070 localhost

```
luis@ubuntu: ~  
luis@ubuntu: ~ 80x14  
luis@ubuntu:~$ nmap -p 7070 localhost  
  
Starting Nmap 7.60 ( https://nmap.org ) at 2020-04-16 10:47 -04  
Nmap scan report for localhost (127.0.0.1)  
Host is up (0.000099s latency).  
Other addresses for localhost (not scanned): ::1  
  
PORT      STATE SERVICE  
7070/tcp  open  realserver  
  
Nmap done: 1 IP address (1 host up) scanned in 0.05 seconds  
luis@ubuntu:~$
```

para más ejemplos de uso de esta herramienta puedes visitar la página oficial de nmap o el siguiente link, donde encuentre ejemplos prácticos

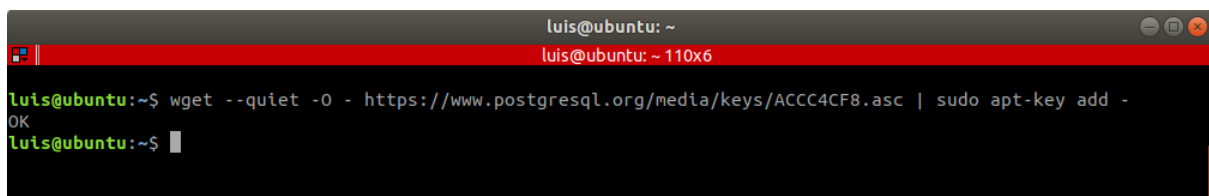
Instalación de la base de datos POSTGRESQL 11

- Lo primero es instalar dos herramientas complementarias que ayudarán en la instalación, estas son vim y wget

```
sudo apt install -y wget vim
```

- Luego agregamos el repositorio de postgresql 11 para poder instalarlo

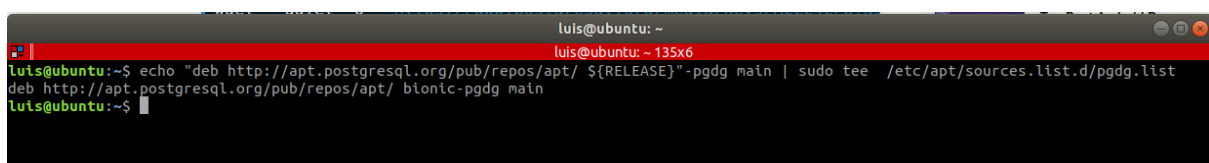
```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
```



```
luis@ubuntu: ~  
luis@ubuntu: ~ 110x6  
luis@ubuntu:~$ wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -  
OK  
luis@ubuntu:~$
```

- Importamos la key para la versión de ubuntu con la que estamos trabajando

```
RELEASE=$(lsb_release -cs)  
echo "deb http://apt.postgresql.org/pub/repos/apt/ ${RELEASE}"-pgdg  
main | sudo tee /etc/apt/sources.list.d/pgdg.list
```



```
luis@ubuntu: ~  
luis@ubuntu: ~ 135x6  
luis@ubuntu:~$ echo "deb http://apt.postgresql.org/pub/repos/apt/ ${RELEASE}"-pgdg main | sudo tee /etc/apt/sources.list.d/pgdg.list  
deb http://apt.postgresql.org/pub/repos/apt/ bionic-pgdg main  
luis@ubuntu:~$
```

- y podemos verificar que ya contamos con el punto de conexión con el repositorio de postgresql 11

```
cat /etc/apt/sources.list.d/pgdg.list
```

```
luis@ubuntu: ~  
luis@ubuntu: ~ 63x6  
luis@ubuntu:~$ cat /etc/apt/sources.list.d/pgdg.list  
deb http://apt.postgresql.org/pub/repos/apt/ bionic-pgdg main  
luis@ubuntu:~$
```

- Instalación de postgresql 11

sudo apt update

sudo apt -y install postgresql-11

- Imagen luego de la instalación, donde se remarca la finalización de la instalación y el puerto 5432 usado por defecto por postgresql

```
luis@ubuntu: ~  
luis@ubuntu: ~ 96x21  
seleccionando el valor para shared_buffers ... 128MB  
seleccionando el huso horario por omisión ... America/Santiago  
seleccionando implementación de memoria compartida dinámica ...posix  
creando archivos de configuración ... hecho  
ejecutando script de inicio (bootstrap) ... hecho  
realizando inicialización post-bootstrap ... hecho  
sincronizando los datos a disco ... hecho  
  
Completado. Ahora puede iniciar el servidor de bases de datos usando:  
  
pg_ctlcluster 11 main start  
  
Ver Cluster Port Status Owner    Data directory          Log file  
11  main    5432 down   postgres /var/lib/postgresql/11/main /var/log/postgresql/postgresql-11-main.log  
update-alternatives: utilizando /usr/share/postgresql/11/man/man1/postmaster.1.gz para proveer /usr/share/man/man1/postmaster.1.gz (postmaster.1.gz) en modo automático  
Procesando disparadores para libc-bin (2.27-3ubuntu1) ...  
Procesando disparadores para systemd (237-3ubuntu10.39) ...  
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...  
Procesando disparadores para ureadahead (0.100.0-21) ...
```

- Verificamos que efectivamente está corriendo en el puerto señalado utilizando nmap

nmap localhost

```
luis@ubuntu: ~  
luis@ubuntu: ~ 71x19  
luis@ubuntu:~$ nmap localhost  
Starting Nmap 7.60 ( https://nmap.org ) at 2020-04-16 11:24 -04  
Nmap scan report for localhost (127.0.0.1)  
Host is up (0.00016s latency).  
Other addresses for localhost (not scanned): ::1  
Not shown: 992 closed ports  
PORT      STATE SERVICE  
22/tcp    open  ssh  
80/tcp    open  http  
139/tcp   open  netbios-ssn  
445/tcp   open  microsoft-ds  
631/tcp   open  ipp  
3306/tcp  open  mysql  
5432/tcp  open  postgresql  
7070/tcp  open  realserver  
Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds  
luis@ubuntu:~$
```

Cuando trabajas en equipo necesitarás habilitar la conexión desde otros equipos a su base de datos, para hacer eso debe editar el archivo **/etc/postgresql/11/main/postgresql.conf** En este archivo deben comentar la siguiente instrucción.

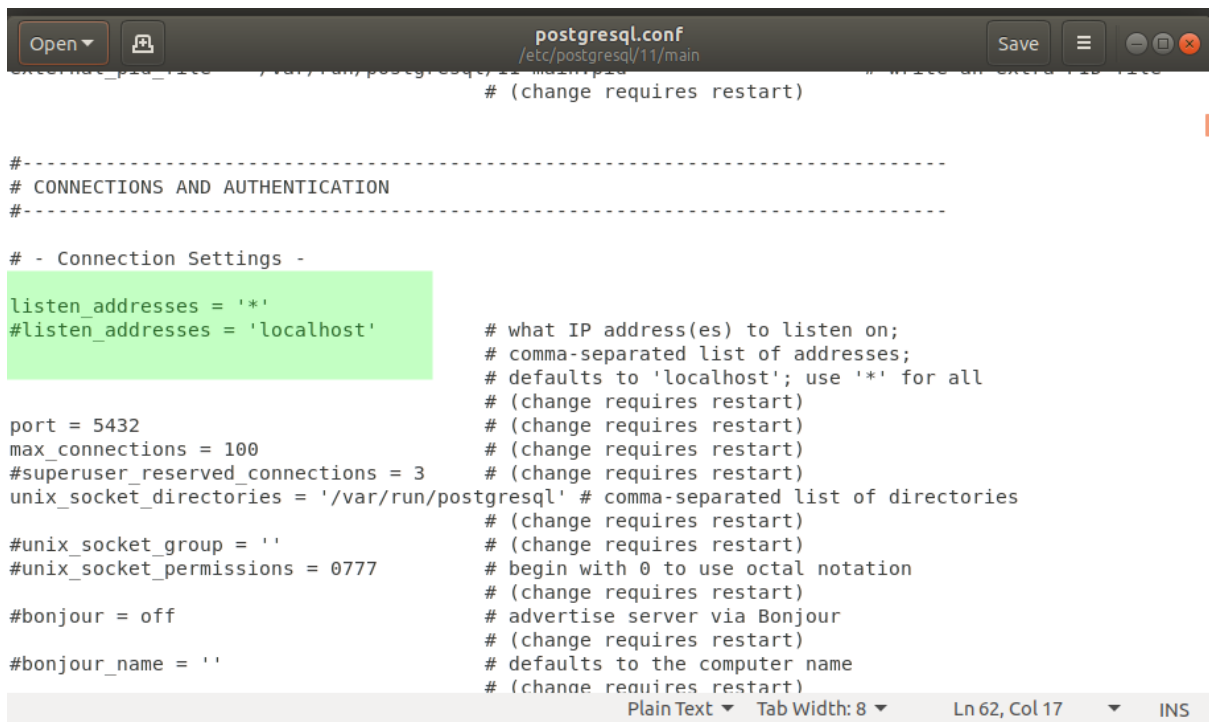
listen_addresses = 'localhost'

y dejar una nueva línea cambiando localhost por *, con lo cual permitirá a cualquiera (que pertenezca a su subred) conectarse a su base de datos (esto solo lo hacemos por facilidad para el curso, pero en producción suele dejarse un rango de IPs o IPs específicas con acceso a la DB)

La línea nueva debería verse así

listen_addresses = '*'

El archivo debería tener la siguiente apariencia (luego guardar los cambios y salir de programa)



```
Open postgresql.conf Save
/etc/postgresql/11/main
# (change requires restart)

#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -

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)
max_connections = 100           # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
                                # (change requires restart)
#unix_socket_group = ''         # (change requires restart)
#unix_socket_permissions = 0777 # begin with 0 to use octal notation
                                # (change requires restart)
#bonjour = off                 # advertise server via Bonjour
                                # (change requires restart)
#bonjour_name = ''             # defaults to the computer name
                                # (change requires restart)

Plain Text Tab Width: 8 Ln 62, Col 17 INS
```

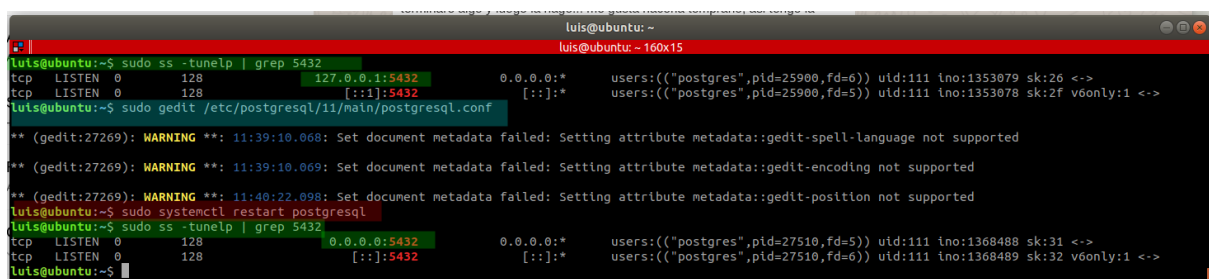
nota: para poder editar el archivo debe ser con permisos de superusuario, ejemplo:

sudo gedit /etc/postgresql/11/main/postgresql.conf

- Reiniciamos postgresql para que el cambio realizado sobre la configuración de postgresql tenga efecto

sudo systemctl restart postgresql

- Este sería el resultado de la secuencia de instrucciones realizadas, donde primero verificamos que solo localhost (**127.0.0.1:5432**) tiene acceso a la db, luego modificamos el archivo mencionado, reiniciamos el servicio de postgresql y finalmente verificamos que todo funciona correctamente y la db queda habilitada para recibir consultas desde cualquier otra máquina (**0.0.0.0:5432**)



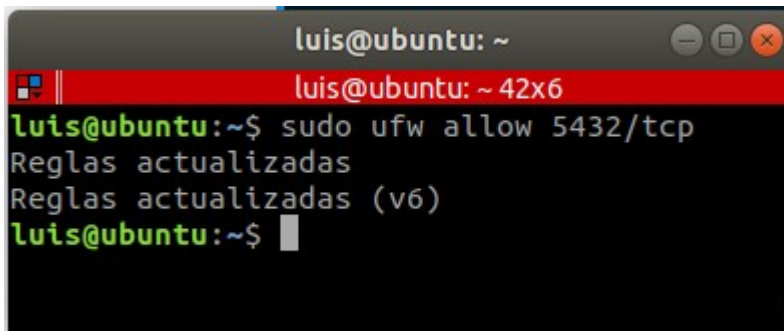
```
luis@ubuntu: ~
luis@ubuntu: ~ 160x15
luis@ubuntu:~$ sudo ss -tunelp | grep 5432
tcp LISTEN 0      128          127.0.0.1:5432      0.0.0.0:*        users:((("postgres",pid=25900,fd=6)) uid:111 ino:1353079 sk:26 <->
tcp LISTEN 0      128          [::]:5432          [::]:*          users:((("postgres",pid=25900,fd=5)) uid:111 ino:1353078 sk:2f v6only:1 <->
luis@ubuntu:~$ sudo gedit /etc/postgresql/11/main/postgresql.conf
** (gedit:27269): WARNING **: 11:39:10.068: Set document metadata failed: Setting attribute metadata::gedit-spell-language not supported
** (gedit:27269): WARNING **: 11:39:10.069: Set document metadata failed: Setting attribute metadata::gedit-encoding not supported
** (gedit:27269): WARNING **: 11:40:22.098: Set document metadata failed: Setting attribute metadata::gedit-position not supported
luis@ubuntu:~$ sudo systemctl restart postgresql
luis@ubuntu:~$ sudo ss -tunelp | grep 5432
tcp LISTEN 0      128          127.0.0.1:5432      0.0.0.0:*        users:((("postgres",pid=27510,fd=5)) uid:111 ino:1368488 sk:31 <->
tcp LISTEN 0      128          [::]:5432          [::]:*          users:((("postgres",pid=27510,fd=6)) uid:111 ino:1368489 sk:32 v6only:1 <->
luis@ubuntu:~$
```

- La instrucción para realizar la primera y última verificación es

sudo ss -tunelp | grep 5432

- y si tiene el firewall corriendo debe activar el uso del puerto de postgresql (usando firewall UFW por defecto de ubuntu)

```
sudo ufw allow 5432/tcp
```

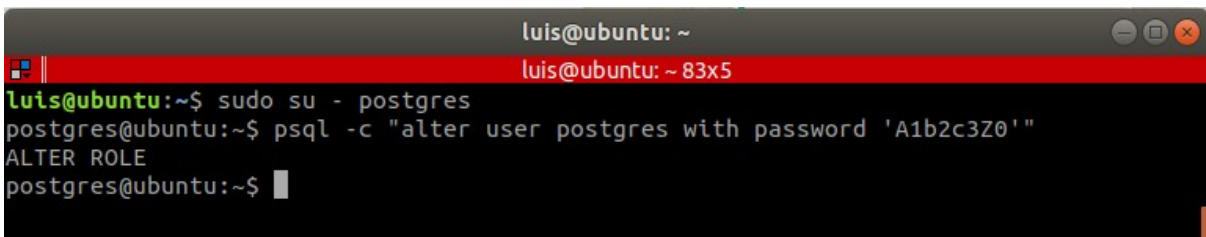


```
luis@ubuntu: ~  
luis@ubuntu: ~ 42x6  
luis@ubuntu:~$ sudo ufw allow 5432/tcp  
Reglas actualizadas  
Reglas actualizadas (v6)  
luis@ubuntu:~$
```

- Una vez instalado postgresql es importante establecer una contraseña robusta para el usuario administrador de postgresql (user: postgres)

```
sudo su - postgres
```

```
psql -c "alter user postgres with encrypted password 'A1b2c3Z0' "
```

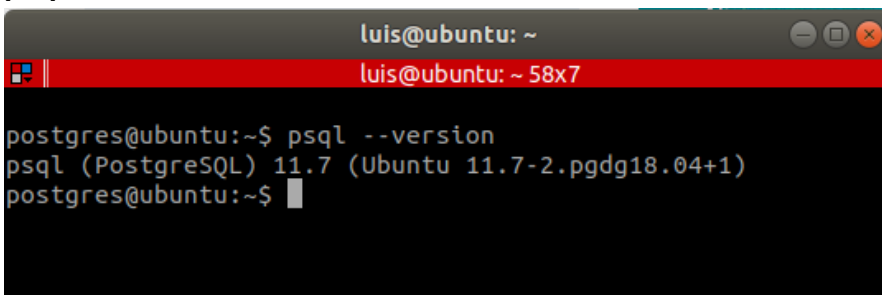


```
luis@ubuntu: ~  
luis@ubuntu: ~ 83x5  
luis@ubuntu:~$ sudo su - postgres  
postgres@ubuntu:~$ psql -c "alter user postgres with password 'A1b2c3Z0'"  
ALTER ROLE  
postgres@ubuntu:~$
```

- Verificar versión de postgresql instalada

```
sudo su - postgres
```

```
psql --version
```



```
luis@ubuntu: ~  
luis@ubuntu: ~ 58x7  
postgres@ubuntu:~$ psql --version  
psql (PostgreSQL) 11.7 (Ubuntu 11.7-2.pgdg18.04+1)  
postgres@ubuntu:~$
```

- otra alternativa

```
sudo su - postgres
```

```
psql
```

```
SELECT version();
```

```
exit;
```


Ejemplo básico

Pensemos que se quiere crear una base de datos del personal de una empresa, contemplando un usuario genérico para los asistentes sociales y una tabla empleados

- primero debe estar bajo el perfil de usuario administrador postgres

```
sudo su - postgres
```

- creamos el usuario genérico

```
createuser dbusr_asistsoc
```

- creamos la base de datos de recursos humanos

```
createdb db_rekursoshumanos -O dbusr_asistsoc
```

- verificamos que se creó correctamente y se asignó al usuario

```
psql -l | grep db_rekursoshumanos
```

- Ingresamos al ambiente de postgresql donde se ejecutan las instrucciones sql, ejecutar

```
psql
```

- establecemos una contraseña para el usuario creado

```
alter user dbusr_asistsoc with encrypted password 'Pwd3astSoc1';
```

- asigno los privilegios para usar la db al usuario dbusr_asistsoc

```
grant all privileges on database db_rekursoshumanos to dbusr_asistsoc;
```

- salir y volver a ingresar ahora con el usuario dbusr_asistsoc (la opción -W permite que ingresen la contraseña por la línea de comandos)

```
exit;
```

```
psql -h localhost -d db_rekursoshumanos -U dbusr_asistsoc -W  
(ingresar contraseña solicitada)
```

- creamos la tabla empleados donde de psql

```
create table personal (id int, nombre text, apellido text, cargo text);
```

- Insertamos algunos registros

```
insert into personal (id,nombre,apellido,cargo)
values (1,'Pedro','Piedras','subgerente');
```

```
insert into personal (id,nombre,apellido,cargo)
values (2,'Mario','Mariano','programador');
```

```
insert into personal (id,nombre,apellido,cargo)
values (3,'Mirta','Miranda','programador');
```

```
insert into personal (id,nombre,apellido,cargo)
values (4,'Arturo','Astudillo','arquitecto');
```

```
insert into personal (id,nombre,apellido,cargo)
values (5,'Maria','Marea','gerente');
```

- realizamos consultas básicas.

```
Select * from personal;
```

```
select count(*) total_empleados from personal;
```

```
select * from personal where cargo='gerente';
```

```
select count(*) cantidad_programadores
from personal
where cargo='programador';
```

The screenshot shows a terminal window with the following content:

```

luis@ubuntu: ~/Documentos/tbd
luis@ubuntu: ~/Documentos/tbd 103x27

luis@ubuntu:~/Documentos/tbd$ sudo su - postgres
postgres@ubuntu:~$ psql -h localhost -d db_recursoshumanos -U dbusr_asistsoc -W
Contraseña:
psql (12.2 (Ubuntu 12.2-2.pgdg18.04+1), servidor 11.7 (Ubuntu 11.7-2.pgdg18.04+1))
conexión SSL (protocolo: TLSv1.3, cifrado: TLS_AES_256_GCM_SHA384, bits: 256, compresión: desactivado)
Digite «help» para obtener ayuda.

db_recursoshumanos=> Select * from personal;
 id | nombre | apellido | cargo
-----+-----+-----+-----
  1 | Pedro  | Piedras  | subgerente
  2 | Mario  | Mariano  | programador
  3 | Mirta  | Miranda  | programador
  4 | Arturo | Astudillo| arquitecto
  5 | Maria  | Marea    | gerente
(5 filas)

db_recursoshumanos=> select count(*) cantidad_programadores
db_recursoshumanos-> from personal
db_recursoshumanos-> where cargo='programador';
 cantidad_programadores
-----
                        2
(1 fila)

db_recursoshumanos=>

```

Instalación de PostGIS sobre Postgresql 11

PostGIS es una extensión de Postgresql para trabajar con información geoespacial (un conjunto de funciones y procedimientos almacenados que facilitan el trabajo con información geoespacial), para más información <https://postgis.net/install/>

Nota: PostGIS será usada en secciones posteriores de este curso, pero es mejor dejarla configurada desde el inicio para no tener contratiempos más adelante.

```
sudo apt update
```

```
sudo apt install postgis postgresql-11-postgis-2.5
```

solicitará confirmación (y + Enter), demora un poco, ya que son 150MB aprox.

Se debe crear un usuario, una base de datos y una vez conectado con dicha base de datos se deben agregar las extensiones “postgis” (funciones básicas) y “postgis_topology” (funciones adicionales).

Nota: existen varias otras extensiones agregadas al instalar PostGIS, pero para el trabajo requerido en este curso, con las dos extensiones agregadas es suficiente.

- Serie de comandos para crear db y agregar las extensiones de PostGIS

```
sudo su - postgres
```

```
createuser dbusr_geologo
```

```
createdb db_geodata -O dbusr_geologo
```

```
psql
```

```
alter user dbusr_geologo with encrypted password 'Pwd0geousr0';
```

```
grant all privileges on database db_geodata to dbusr_geologo;
```

```
exit;
```

```
psql -h localhost -d db_geodata
```

(ingresar password solicitada)

```
CREATE EXTENSION postgis;
```

```
CREATE EXTENSION postgis_topology;
```

```
SELECT PostGIS_version();
```

```
exit;
```

Secuencia de pasos ejecutada

```
luis@ubuntu: ~  
luis@ubuntu: ~ 108x32  
postgres@ubuntu:~$ exit  
logout  
luis@ubuntu:~$ sudo su - postgres  
postgres@ubuntu:~$ createuser dbusr_geologo  
postgres@ubuntu:~$ createdb db_geodata -O dbusr_geologo  
postgres@ubuntu:~$ psql  
psql (12.2 (Ubuntu 12.2-2.pgdg18.04+1), servidor 11.7 (Ubuntu 11.7-2.pgdg18.04+1))  
Digite «help» para obtener ayuda.  
  
postgres=# alter user dbusr_geologo with encrypted password 'Pwd0geousr0';  
ALTER ROLE  
postgres=# grant all privileges on database db_geodata to dbusr_geologo;  
GRANT  
postgres=# exit;  
postgres@ubuntu:~$ psql -h localhost -d db_geodata  
Contraseña para usuario postgres:  
psql (12.2 (Ubuntu 12.2-2.pgdg18.04+1), servidor 11.7 (Ubuntu 11.7-2.pgdg18.04+1))  
conexión SSL (protocolo: TLSv1.3, cifrado: TLS_AES_256_GCM_SHA384, bits: 256, compresión: desactivado)  
Digite «help» para obtener ayuda.  
  
db_geodata=# CREATE EXTENSION postgis;  
CREATE EXTENSION  
db_geodata=# CREATE EXTENSION postgis_topology;  
CREATE EXTENSION  
db_geodata=# exit;  
postgres@ubuntu:~$ exit  
logout  
luis@ubuntu:~$
```

Instalación de pgAdmin4

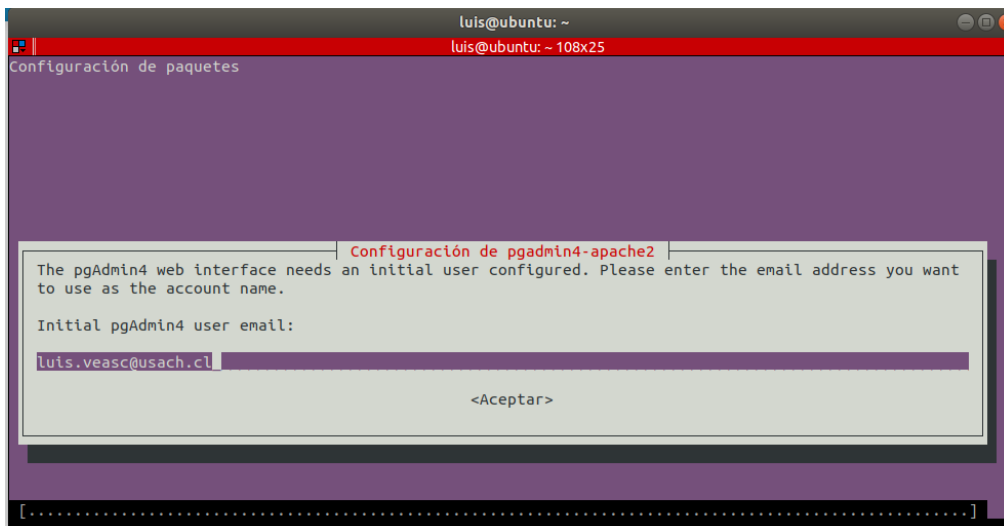
Herramienta para realizar la administración de Postgresql y ejecutar comandos sql sobre las DBs creadas. La versión 4 de pgAdmin tiene una interfaz web, por ende también se instala un servidor de aplicaciones web, en este caso apache2.

sudo apt update

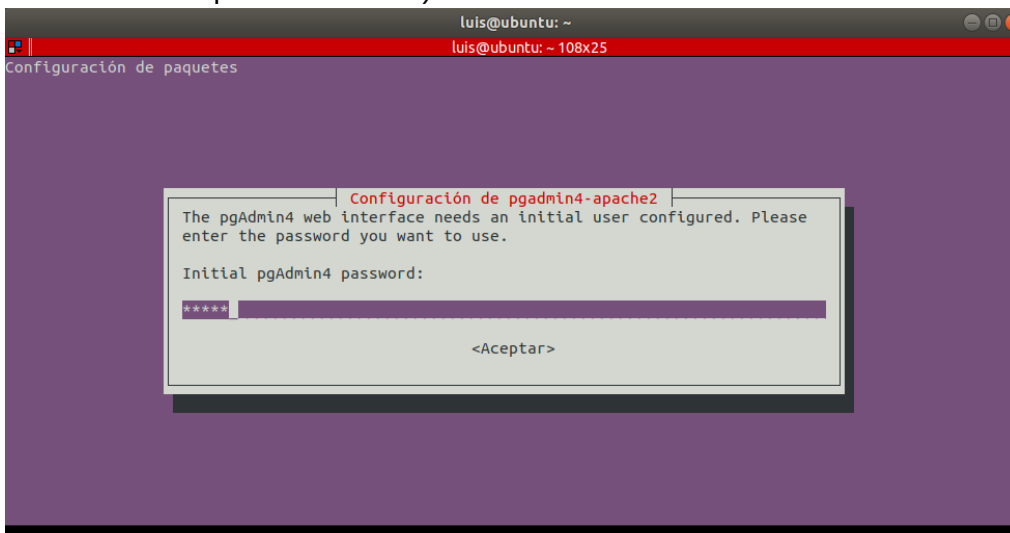
sudo apt install pgadmin4 pgadmin4-apache2

Solicitará confirmación (S + Enter)

- Durante la instalación se debe ingresar el su correo, que será usado como usuario de la aplicación.



- Luego solicitará una contraseña (no es necesario que sea la del su correo, puede ser cualquier contraseña)



- Si tiene activo el firewall debe habilitar las conexiones tipo http y https, para esto ejecute

```
sudo ufw allow http
sudo ufw allow https
```

Si se presenta algún problema durante la instalación al levantar el servidor apache2, es por que probablemente tiene algún otro servicio corriendo en los puertos 80(http) y 443 (https). En mi caso tenía **nginx** como proxy de aplicaciones web, lo que debes hacer es detener el servidor **nginx**, para luego levantar el servidor apache2.

```
sudo service nginx stop
sudo systemctl start apache2.service
```

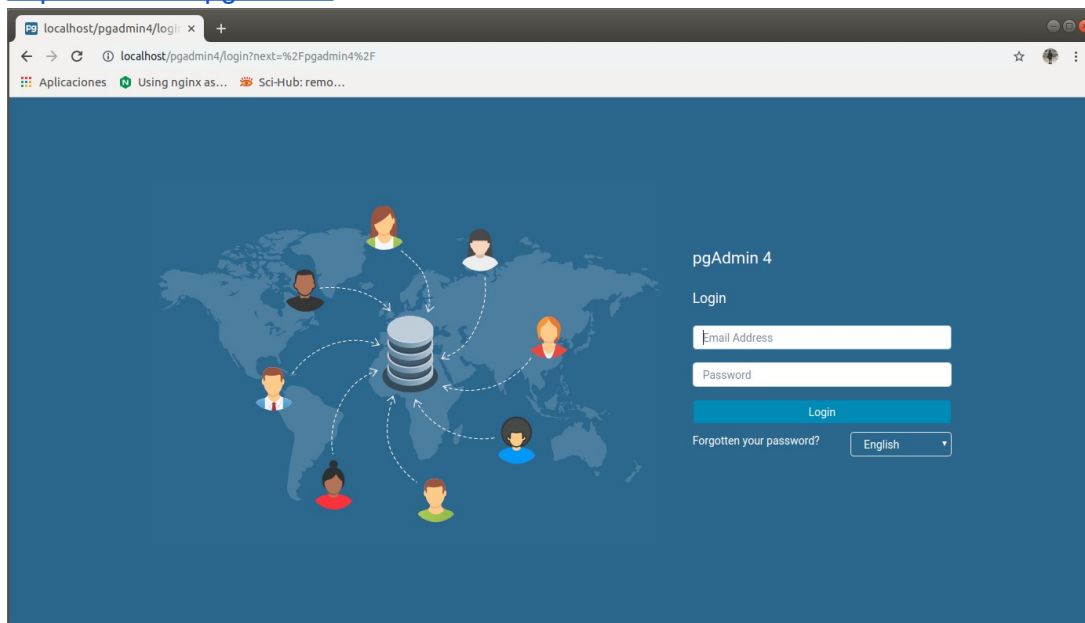
```
luis@ubuntu: ~  
luis@ubuntu: ~ 172x25  
=====
```

```
Job for apache2.service failed because the control process exited with error code.  
See 'systemctl status apache2.service' and 'journalctl -xe' for details.  
invoke-rc.d: initscript apache2, action 'restart' failed.  
● apache2.service - The Apache HTTP Server  
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)  
   Drop-In: /lib/systemd/system/apache2.service.d  
            └─apache2-systemd.conf  
   Active: failed (Result: exit-code) since Thu 2020-04-16 15:49:10 -04; 28ms ago  
   Process: 6877 ExecStart=/usr/sbin/apachectl start (code=exited, status=1/FAILURE)
```

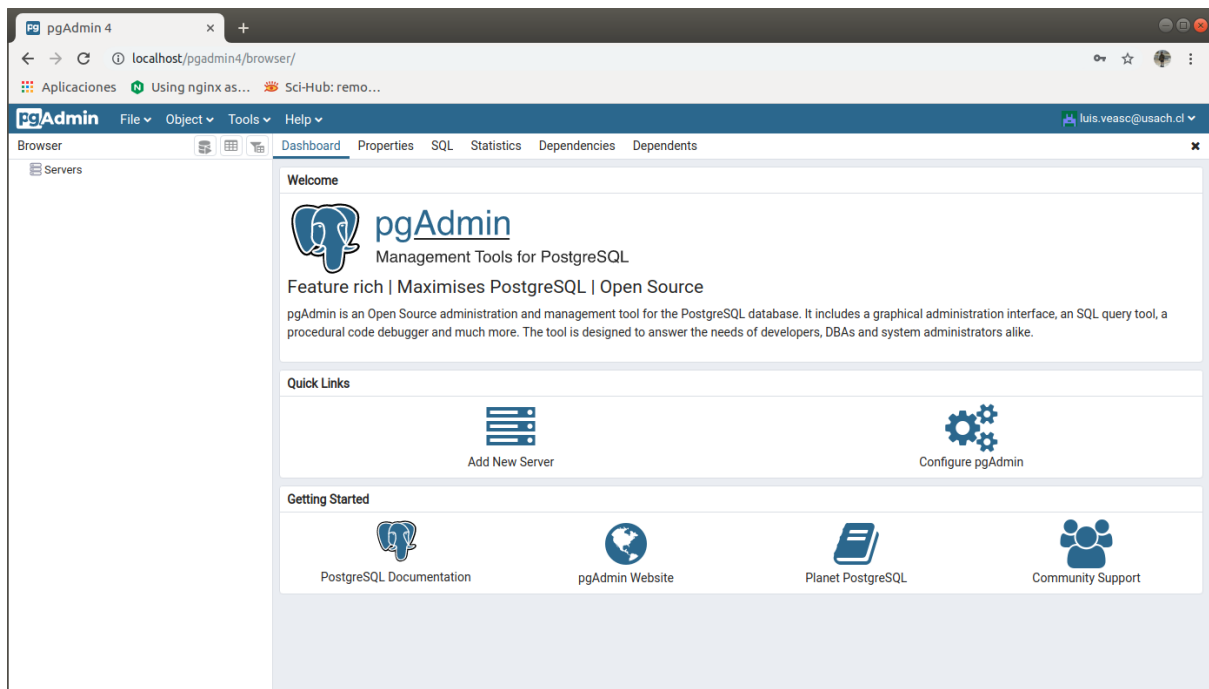
```
abr 16 15:49:10 ubuntu apachectl[6877]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' direc  
tive globally to suppress this message  
abr 16 15:49:10 ubuntu apachectl[6877]: (98)Address already in use: AH00072: make_sock: could not bind to address [::]:80  
abr 16 15:49:10 ubuntu apachectl[6877]: (98)Address already in use: AH00072: make_sock: could not bind to address 0.0.0.0:80  
abr 16 15:49:10 ubuntu apachectl[6877]: no listening sockets available, shutting down  
abr 16 15:49:10 ubuntu apachectl[6877]: AH00015: Unable to open logs  
abr 16 15:49:10 ubuntu apachectl[6877]: Action 'start' failed.  
abr 16 15:49:10 ubuntu apachectl[6877]: The Apache error log may have more information.  
abr 16 15:49:10 ubuntu systemd[1]: apache2.service: Control process exited, code=exited status=1  
abr 16 15:49:10 ubuntu systemd[1]: apache2.service: Failed with result 'exit-code'.  
abr 16 15:49:10 ubuntu systemd[1]: Failed to start The Apache HTTP Server.  
Procesando disparadores para libc-bin (2.27-3ubuntu1) ...  
Procesando disparadores para doc-base (0.10.8) ...  
Procesando 1 archivo doc-base añadido
```

Una vez está todo corriendo correctamente debe conectarse a la ruta **http://[HOST-UTILIZADO]/pgadmin4**, en mi caso

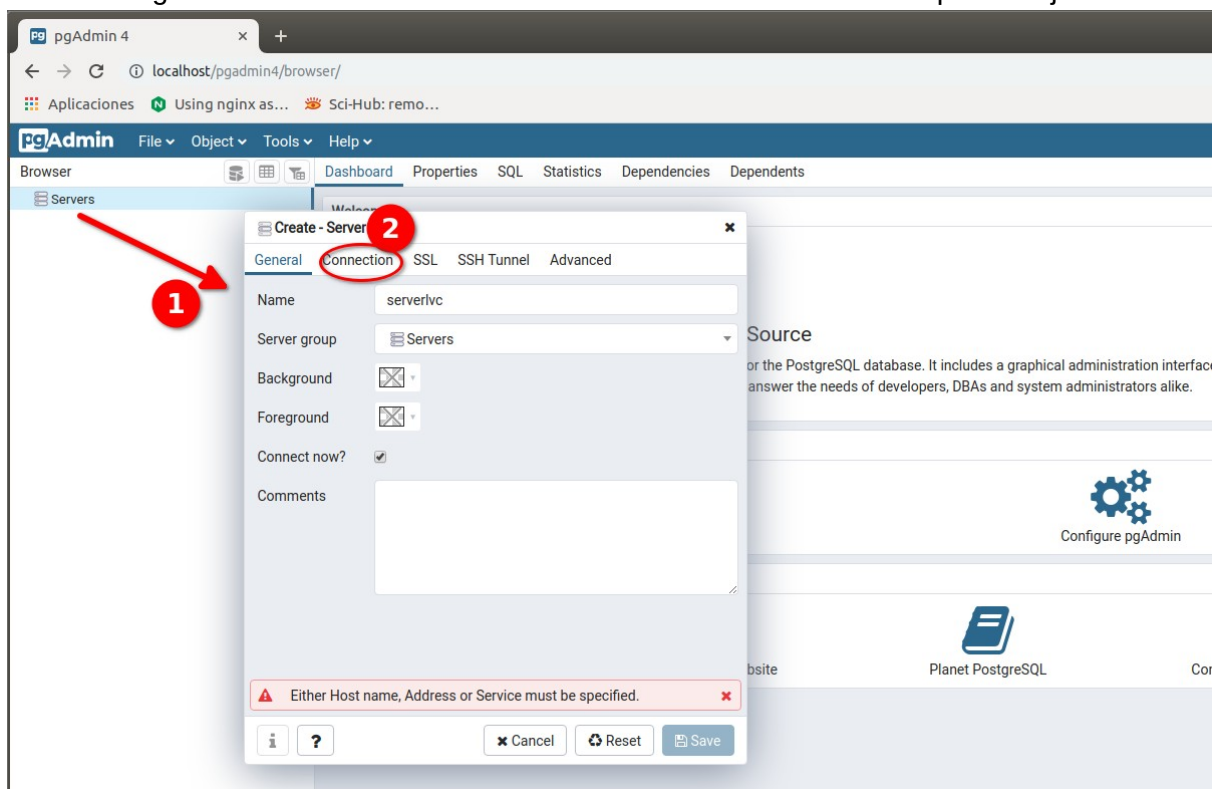
<http://localhost/pgadmin4>



- Esta será la apariencia al ingresar al sistema



- Luego debe establecer una conexión a la base de datos con la que trabajará



- Debe rellenar la información solicitada y guardar, con esto se podrá conectar a las bases de datos que maneje el usuario "postgres" en el servidor "localhost"

- Luego para crear una base de datos, tablas y relaciones ver el video **clase1_pgadmin_crea_tablas.mp4**

Estructura básica de una sentencia sql

→ Select *
→ From alumno;
↑
tabla

campos a
mostrar

→ Select rut, nombre
From alumno;

campos a
mostrar

Select rut, nombre
From alumno
→ where rut = 1;
busca inf particular

→ Select count(*) cant_alumnos
From alumno;

función que sirve para
contar tuplas

Select count(*) cant_alumnos,
idCarrera
From alumno
→ group by idCarrera
→ order by idCarrera;
agrupa por
columnas
ordena por
columnas

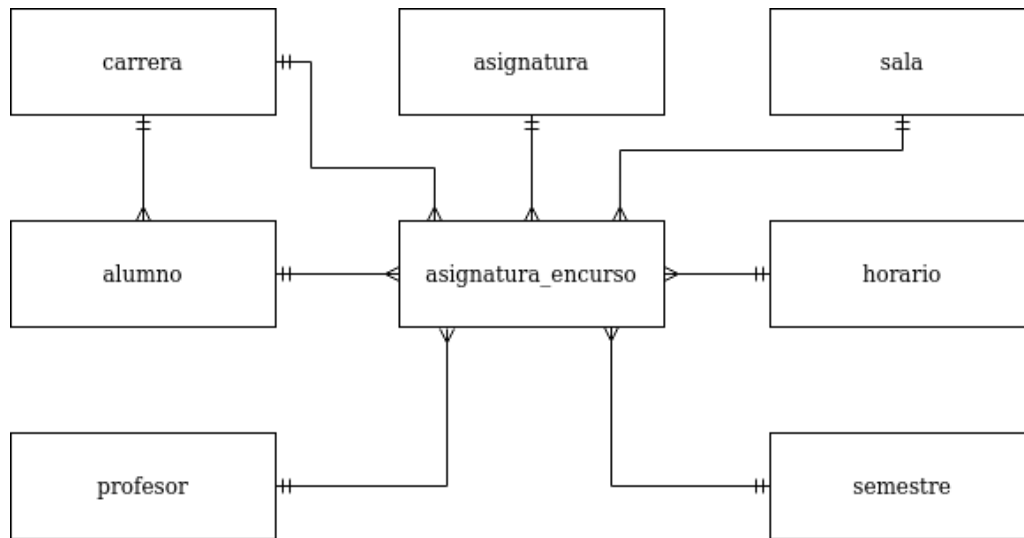
Intersección entre dos tablas

```
Select a.nombre nombre_alumno  
, b.nombre nombre_carrera  
From alumno a, carrera b  
where a.idCarrera=b.id;
```

```
Select a.nombre nombre_alumno  
, b.nombre nombre_carrera  
From alumno a  
inner join carrera b  
on a.idCarrera=b.id";
```

Uso del ambiente de trabajo

Ahora con el ambiente de trabajo desplegado crear la base de datos cursosdb basada en modelo de datos entregado en la siguiente imagen.



Las columnas por tabla son las siguientes:

carrera (id, nombre)

alumno (rut, dv, nombre, idCarrera)

asignatura (id, nombre)

horario (id, descrip)

profesor (rut, nombre, fnacimiento)

sala (codigo, capacidad, ubicacion)

semestre(codigo, descrip)

asignatura_encurso(rutalu, id, rutprof, codSala, idAsig, codSem, idHorario, idCarrera)

