



Realización de la práctica:

- 1) En primer lugar, tras haber iniciado sesión en la máquina virtual, el primer paso fue la instalación de Postgres:

```
usuario@ubuntu: ~  
This message is shown once once a day. To disable it please create the  
/home/usuario/.hushlogin file.  
usuario@DESKTOP-7TTIV4R:~$ ssh usuario@10.6.128.119  
The authenticity of host '10.6.128.119 (10.6.128.119)' can't be established.  
ECDSA key fingerprint is SHA256:w1NR1Pj6bJ1W14R9URVfBALfb/T4akPSJb6dgPN91zY.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '10.6.128.119' (ECDSA) to the list of known hosts.  
usuario@10.6.128.119's password:  
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-81-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Thu Oct 21 12:04:20 UTC 2021  
  
System load:  0.03          Processes:      147  
Usage of /:   34.9% of 19.56GB  Users logged in: 1  
Memory usage: 7%           IPv4 address for ens3: 10.6.128.119  
Swap usage:   0%  
  
0 updates can be applied immediately.  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
Last login: Thu Oct 21 12:03:37 2021  
usuario@ubuntu:~$ sudo apt-get install postgresql
```

Imagen [1]: Instalación de Postgres

- 2) Tras la instalación, accedemos al super usuario (\$ sudo su postgres) y empleamos un comando especial (\$ psql -U postgres) para poder ejecutar los comandos SQL necesarios.

```
usuario@ubuntu: ~  
  
System load:  0.0          Processes:      147  
Usage of /:   36.0% of 19.56GB  Users logged in: 1  
Memory usage: 10%          IPv4 address for ens3: 10.6.1  
Swap usage:   0%  
  
* Super-optimized for small spaces - read how we shrank the mem  
footprint of MicroK8s to make it the smallest full K8s around  
  
https://ubuntu.com/blog/microk8s-memory-optimisation  
  
29 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable  
  
*** System restart required ***  
Last login: Thu Oct 21 12:04:21 2021 from 10.150.142.38  
usuario@ubuntu:~$ sudo apt-get install postgresql  
[sudo] password for usuario:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
postgresql is already the newest version (12+214ubuntu0.1).  
0 upgraded, 0 newly installed, 0 to remove and 29 not upgraded.  
usuario@ubuntu:~$ sudo su postgres  
postgres@ubuntu:/home/usuario$ psql -U postgres  
psql (12.8 (Ubuntu 12.8-0ubuntu0.20.04.1))  
Type "help" for help.  
  
postgres=# createuser alu0101124445
```

Imagen [2]: Acceso al superusuario y comandos SQL

- 3) A continuación, procedemos a crear la base de datos (\$ create database pract1;). Y si queremos visualizarlo, usamos el comando \l.

```
postgres=# create database pract1;
CREATE DATABASE
postgres=# \l
```

List of databases					
Name	Owner	Encoding	Collate	Ctype	Access privileges
postgres	postgres	UTF8	C.UTF-8	C.UTF-8	
pract1	postgres	UTF8	C.UTF-8	C.UTF-8	
template0	postgres	UTF8	C.UTF-8	C.UTF-8	=c/postgres + postgres=CTc/postgres
template1	postgres	UTF8	C.UTF-8	C.UTF-8	=c/postgres + postgres=CTc/postgres

(4 rows)

Imagen [3]: Creación BBDD

- 4) Ahora, nos situamos dentro de la base de datos (\$ \c pract1) y comenzamos a introducir los datos.

```
postgres=# \c pract1
You are now connected to database "pract1" as user "postgres".
pract1=# create table usuarios (
pract1(# nombre varchar(30),
pract1(# clave varchar(10)
pract1(# );
CREATE TABLE
pract1=#
```

Imagen [4]: Situar en la BBDD e introducción de los datos

```
pract1=# insert into usuarios (nombre, clave) values ('Isa','asdf');
INSERT 0 1
pract1=# insert into usuarios (nombre, clave) values ('Pablo','jfx344');
INSERT 0 1
pract1=# insert into usuarios (nombre, clave) values ('Ana','tru3fal');
INSERT 0 1
pract1=#
```

Imagen [5]: Introducción de los datos

5) Finalmente, tras haber introducido los datos, podemos hacer uso de diversos comandos:

- \$ \l → mostrar el listado de BBDD del servidor.
- 4 \dt → listar las tablas actuales de la BBDD.

Si queremos verificar la introducción de datos, emplear una secuencia en SQL

\$ select * from usuarios;

```
usuario@ubuntu: ~
-----+-----
(0 rows)

pract1=# insert into usuarios (nombre, clave) values ('Isa','asdf');
INSERT 0 1
pract1=# insert into usuarios (nombre, clave) values ('Pablo','jfx344');
INSERT 0 1
pract1=# insert into usuarios (nombre, clave) values ('Ana','tru3fal');
INSERT 0 1
pract1=# select * from usuarios;
 nombre | clave
-----+-----
 Isa    | asdf
 Pablo  | jfx344
 Ana    | tru3fal
(3 rows)

pract1=# \l
                                List of databases
  Name  | Owner  | Encoding | Collate | Ctype  | Access privileges
-----+-----+-----+-----+-----+-----
 postgres | postgres | UTF8     | C.UTF-8 | C.UTF-8 |
 pract1   | postgres | UTF8     | C.UTF-8 | C.UTF-8 |
 template0 | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres
         |          |          |          |          | postgres=CTc/postgres
 template1 | postgres | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres
         |          |          |          |          | postgres=CTc/postgres
(4 rows)
```

Imagen [6]: Prueba de comandos

```
(4 rows)

pract1=# \dt
                List of relations
 Schema | Name   | Type  | Owner
-----+-----+-----+-----
 public | usuarios | table | postgres
(1 row)
```

Imagen [7]: Prueba de comandos

Resumen de comandos utilizados:

```
create database pract1;  
\l  
createuser alu0101124445  
\q  
create table usuarios (  
nombre varchar(30),  
clave varchar(10)  
);  
\c pract1  
select * from usuarios  
select * from usuarios;  
select * from usuarios;  
insert into usuarios (nombre, clave) values ('Isa','asdf');  
insert into usuarios (nombre, clave) values ('Pablo','jfx344');  
insert into usuarios (nombre, clave) values ('Ana','tru3fal');  
select * from usuarios;  
\l  
\dt  
\s
```