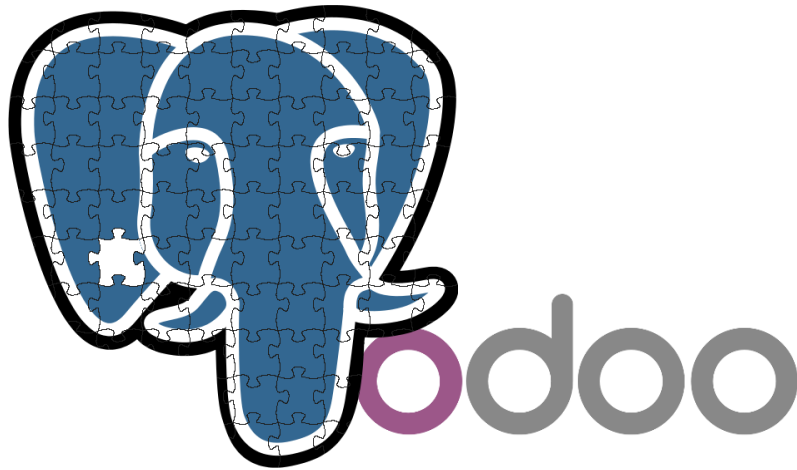




# OPERACIÓN SOBRE BASES DE DATOS EN ODOO



**Realizado por:**

Daniel Rodríguez Fernández

**Madrid 04 de Noviembre de 2022**

## Contenido

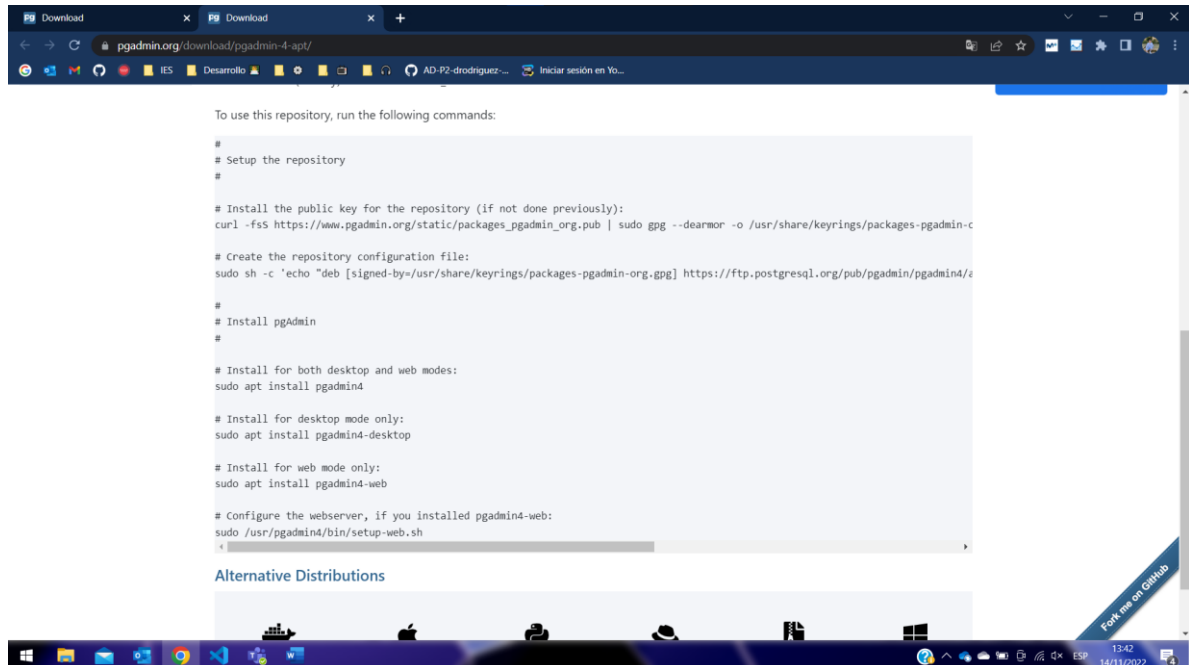
1. Máquina Virtual .....	3
1.1. Instala y configura al menos una herramienta de gestión de base de datos en local y una en remoto: .....	3
1.1.1. pgAdmin4.....	3
1.1.2. DBeaver .....	5
1.2. Realización de una copia de seguridad de la base de datos mediante Odoo. ....	10
1.3. Realización de un script que realice una copia de seguridad de la base de datos todos los domingos a las 12 de la noche.....	11
1.4. Realiza una consulta mediante psql de todos los usuarios del sistema odoo y de todos los productos .....	12
1.5. Realiza las mismas consultas mediante la herramienta o herramientas de gestión de la base de datos que hayas instalado (pgAdmin o DBeaver).....	13
2. Docker.....	14
2.1. Conéctate por consola al contenedor de Docker de PostgreSQL y haz una consulta sencilla con psql. 14	
3. Conclusiones.....	15

# 1. Máquina Virtual

## 1.1. Instala y configura al menos una herramienta de gestión de base de datos en local y una en remoto:

### 1.1.1. pgAdmin4

En primer lugar accedemos a la web oficial de pgAdmin en el caso de que no conozcamos el proceso de descarga en: <https://pgadmin.org/download/pgadmin-4-apt/> (esto es para el caso de usar distribución de Linux).



Iniciamos la máquina virtual, una vez iniciada la máquina virtual comenzamos a seguir las instrucciones de la web oficial. Lo primero que tenemos que hacer es instalar la clave pública del repositorio.

```
daniel@drodriguez:~$ curl -fsS https://www.pgadmin.org/static/packages_pgadmin_org.pub | sudo gpg --dearmor -o /usr/share/keyrings/packages-pgadmin-org.gpg
File '/usr/share/keyrings/packages-pgadmin-org.gpg' exists. Overwrite? (y/N) Y
daniel@drodriguez:~$
```

Ahora creamos el archivo de configuración del repositorio.

```
daniel@drodriguez:~$ sudo sh -c 'echo "deb [signed-by=/usr/share/keyrings/packages-pgadmin-org.gpg] https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/$lsb_release -cs pgadmin4 main" > /etc/apt/sources.list.d/pgadmin4.list && apt update'
Ign:1 http://nightly.odoo.com/15.0/nightly/deb / InRelease
Des:2 http://nightly.odoo.com/15.0/nightly/deb / Release [1.188 B]
Obj:3 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Des:4 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Des:5 http://nightly.odoo.com/15.0/nightly/deb / Release.gpg [833 B]
Des:6 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease [99,8 kB]
Des:7 http://es.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Des:8 http://nightly.odoo.com/15.0/nightly/deb / Packages [2.698 B]
Des:9 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [690 kB]
Des:10 https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/jammy pgadmin4 InRelease [4.217 B]
Des:11 http://es.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [159 kB]
Des:12 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [10,8 kB]
Des:13 http://es.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [417 kB]
Des:14 http://es.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [63,9 kB]
Des:15 http://es.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [580 B]
Des:16 http://es.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [747 kB]
Des:17 https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/jammy pgadmin4/main amd64 Packages [4.528 B]
Des:18 http://es.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [123 kB]
Des:19 https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/jammy pgadmin4/main all Packages [3.895 B]
Des:20 http://es.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [6.744 B]
Des:21 http://es.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [9.368 B]
Des:22 http://es.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [132 B]
Des:23 http://es.archive.ubuntu.com/ubuntu jammy-security/main amd64 Packages [462 kB]
Des:24 http://es.archive.ubuntu.com/ubuntu jammy-security/main Translation-en [102 kB]
Des:25 http://es.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [276 kB]
Des:26 http://es.archive.ubuntu.com/ubuntu jammy-security/restricted Translation-en [57,6 kB]
Des:27 http://es.archive.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [506 kB]
Des:28 http://es.archive.ubuntu.com/ubuntu jammy-security/universe Translation-en [77,1 kB]
Descargados 4.258 kB en 4s (1.179 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se pueden actualizar 93 paquetes. Ejecute 'apt list --upgradable' para verlos.
E: http://nightly.odoo.com/15.0/nightly/deb / Release.gpg Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
daniel@drodriguez:~$
```

Ahora podemos instalar pgadmin4.

Instalación del webserver.

100

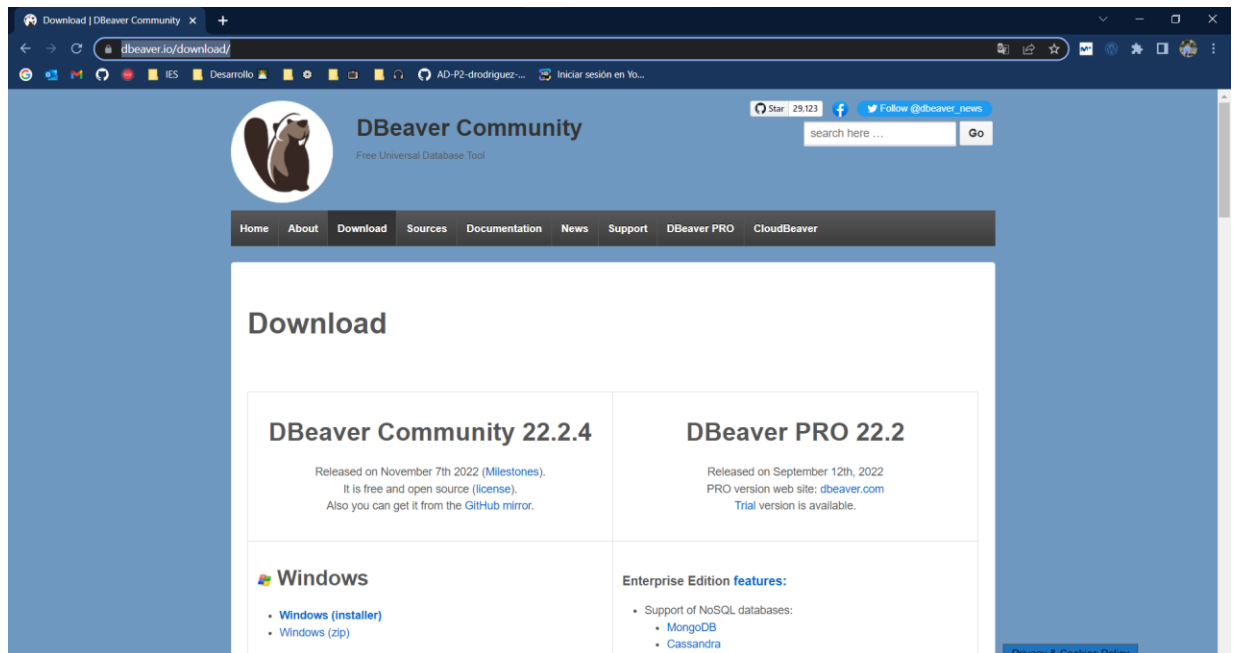
Para comprobar que ha funcionado correctamente hay que crear un túnel a nuestra

\_\_\_\_\_

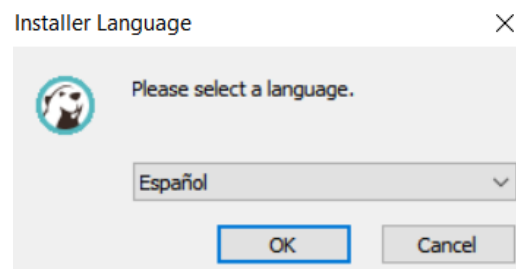
---

### 1.1.2. DBeaver

Primero para instalar DBeaver tenemos que entrar en la web oficial (<https://dbeaver.io/download/>), una vez estamos aquí pulsamos en descargar para la versión de Windows.



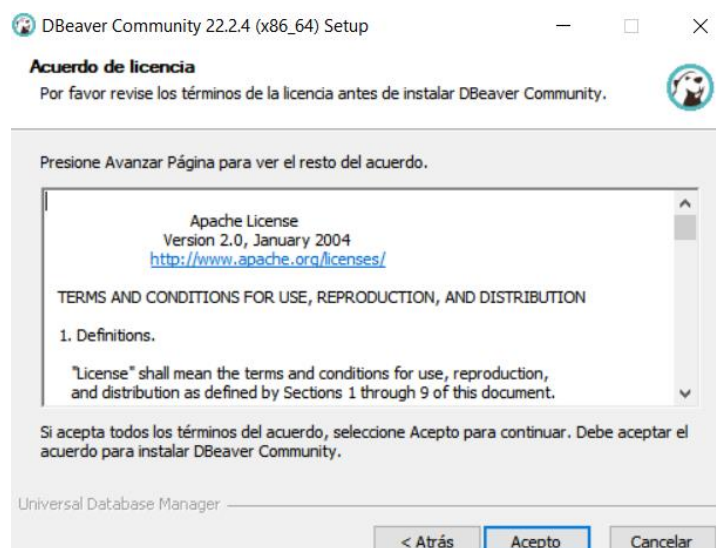
Una vez hemos pulsado en Windows (Installer), comenzamos con la instalación. Lo primero tendremos que indicar el lenguaje una vez se ejecute su instalador.



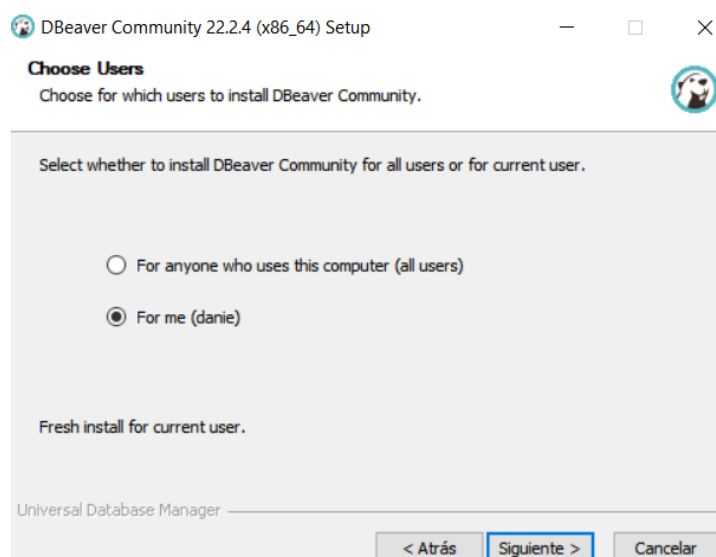
Indicamos “**siguiente**”.



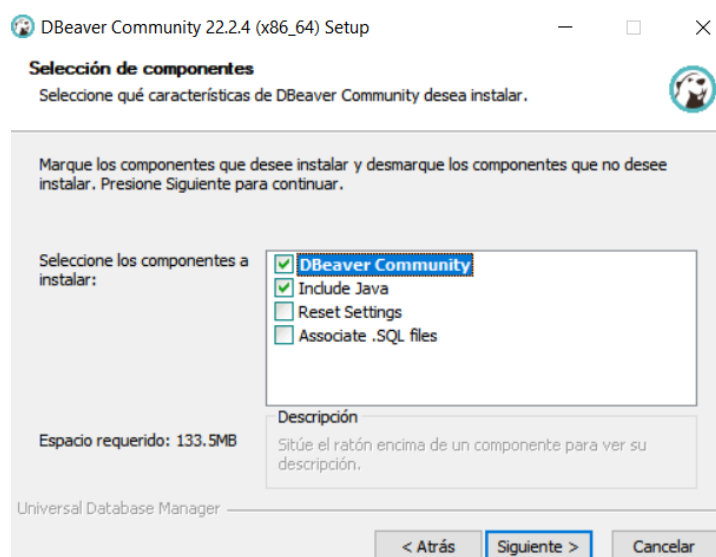
Ahora aceptamos el contrato de licencia.



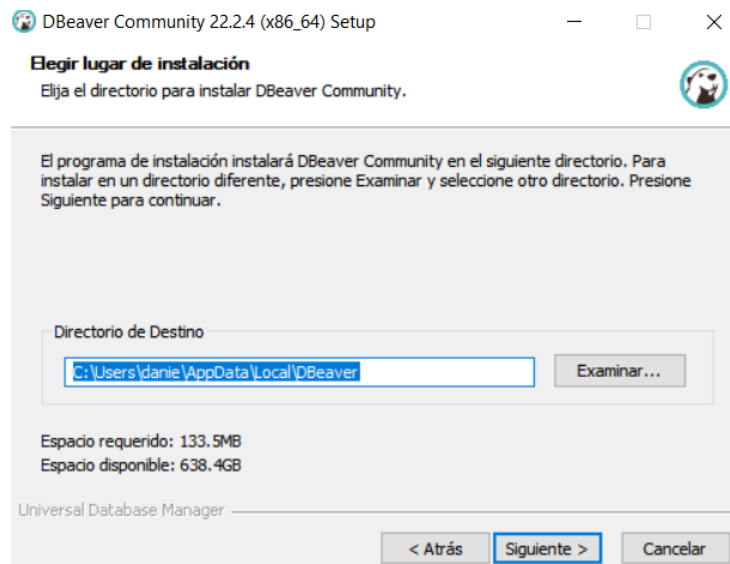
Pulsamos “*siguiente*”.



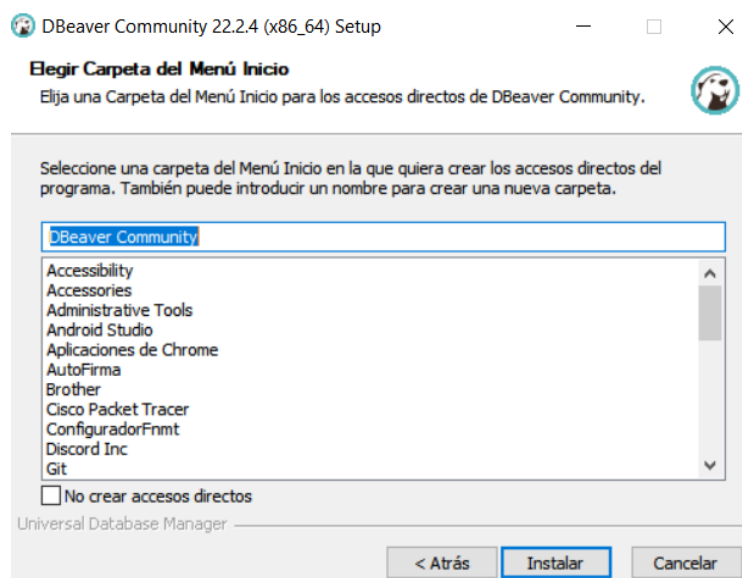
Dejamos los componentes por defecto a instalar.



Indicamos lugar de instalación y pulsamos en “**siguiente**”.



Indicamos el nombre de la carpeta.



Por último finalizamos la instalación.

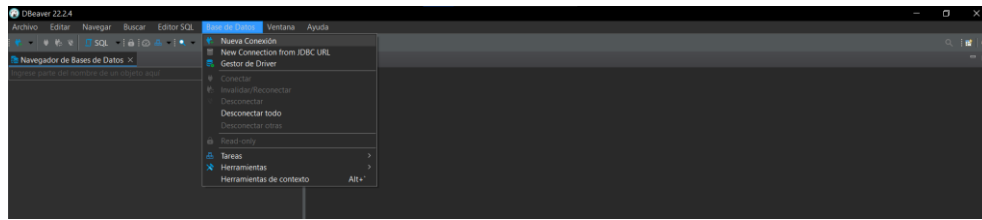




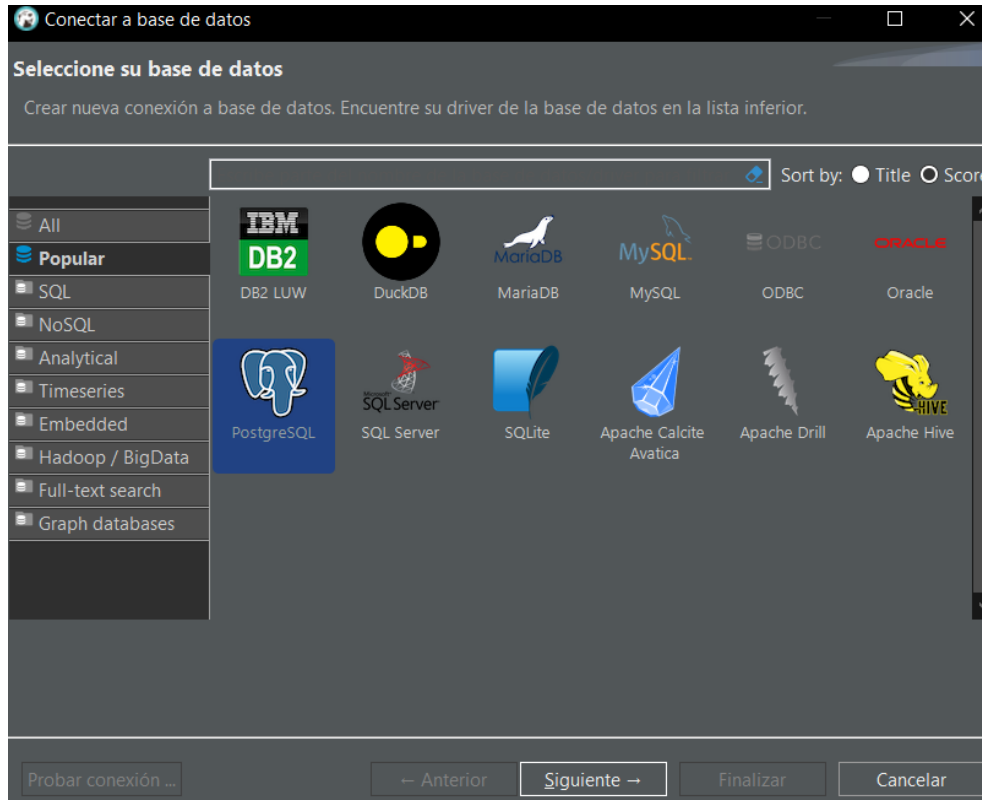
## OPERACIÓN SOBRE BASES DE DATOS EN ODOO

Daniel Rodríguez Fernández

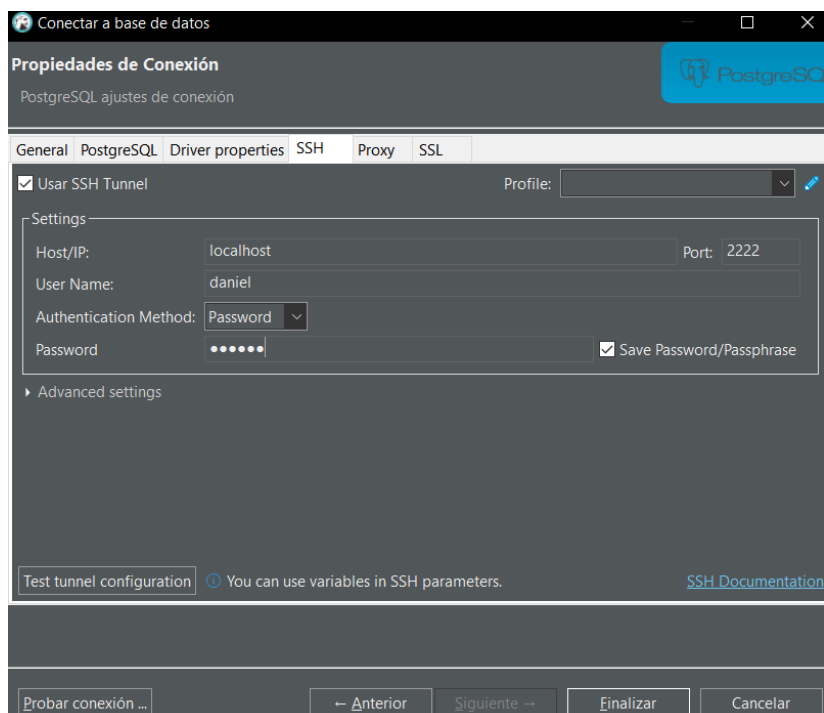
En el menú superior pulsamos sobre “Base de datos” y aquí en “Nueva conexión”.



Ahora indicamos el tipo de BBDD.

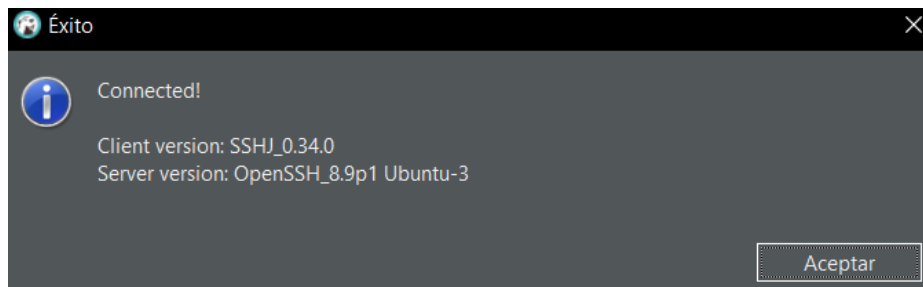


Configuramos la conexión ssh.

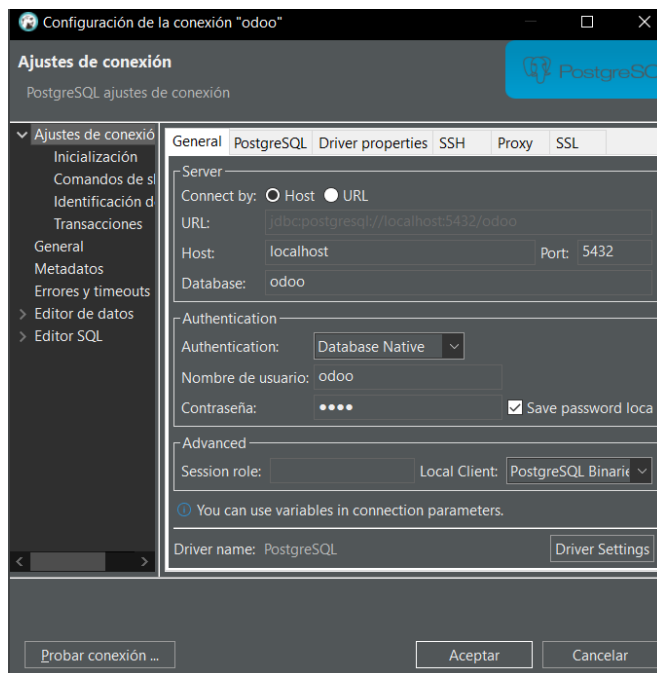




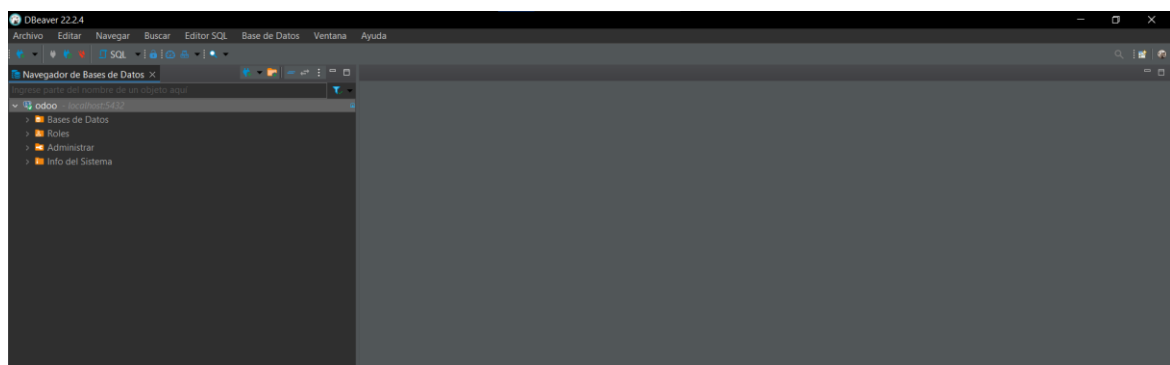
Hacemos un test de tunnel configuration.



Ahora aplicamos la configuración general de la base de datos.

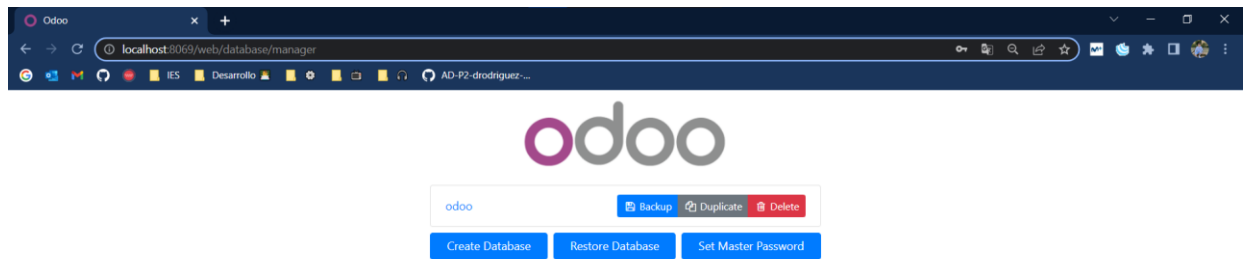


Podemos comprobar como hemos establecido la conexión.



## 1.2. Realización de una copia de seguridad de la base de datos mediante Odoo.

Entramos en la dirección `localhost:8069/web/database/manager` y ahora pulsamos sobre **“backup”**.



Introducimos la password de la base de datos y el formato en el que queremos obtener la copia de seguridad.

### Backup Database

Master Password

....

Database Name

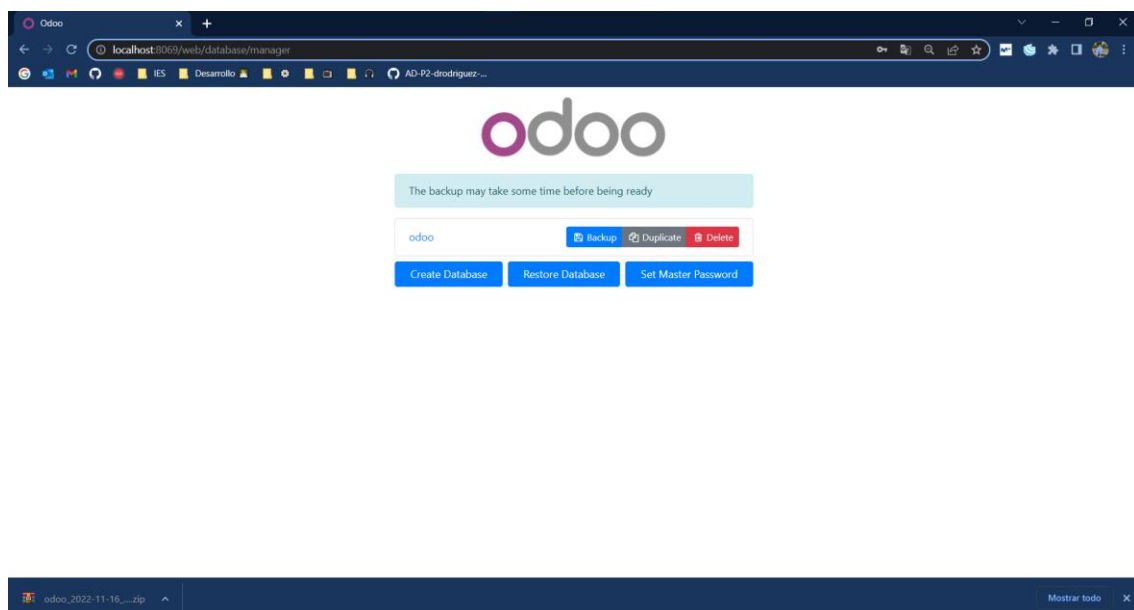
odoo

Backup Format

zip (includes filestore)

Backup

Se nos descarga un archivo zip con la copia de seguridad realizada.



### 1.3. Realización de un script que realice una copia de seguridad de la base de datos todos los domingos a las 12 de la noche

Accedemos con el usuario odoo con el comando "su odoo".

```
daniel@drodriguez:~$ su odoo
Password:
odoo@drodriguez:/home/daniel$ cd ~
odoo@drodriguez:~$
```

Introducimos el siguiente comando. Donde creamos el archivo script de la copia de seguridad.

```
odoo@drodriguez:~$ echo "pg_dump odoo | gzip -c > copiaDDBB_$(date+"%y%m%d_%H%M").gz" > tareacopia.sh
date+%y%m%d_%H%M: command not found
```

Ejecutamos el crontab -e para programar las tareas.

```
odoo@drodriguez:~$ crontab -e
no crontab for odoo - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.basic
 3. /usr/bin/vim.tiny
 4. /bin/ed

Choose 1-4 [1]: 1
crontab: installing new crontab
```

Editamos el archivo crontab y agregamos la nueva tarea programada para la copia de seguridad.

```
daniel@drodriguez: ~
GNU nano 6.2 /tmp/crontab.v0GJT/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
1 00 * * * /home/odoo/tareacopia.sh
```

## 1.4. Realiza una consulta mediante psql de todos los usuarios del sistema odoo y de todos los productos

Ejecutamos el comando `psql odoo`, para acceder a las consultas sobre la bbdd de odoo y después pasamos la `Select` para consultar los usuarios. Filtramos la `Select` para que nos quepan los datos en pantalla en este caso podemos ver el id, Login y la password y así podemos ver el contenido de forma clara.

```
odoo@drolidruvez:~$ psql odoo
psql (14.5 (Ubuntu 14.5-0ubuntu0.22.04.1))
Type "help" for help.

odoo=> select id,login,password from res_users;
 id |      login      | password
-----+-----+-----
  2 | odoo            | $pbkdf2-sha512$25000$6322dq619j7HGONc6937/w$AGRhuWl9UuhmkiGjM002axwxi1TuHolD9MTE00wx8SRWoJvVjamsSOAISotJCswBd3kGhIcm5fJ8XuNiw6/6bQ
  1 | _system_       |
  3 | default        |
  6 | demo           | $pbkdf2-sha512$25000$FEIoxdjbew9BSAnh/38zBg$9/OKPRrHISBKFxfCJujoQHMH6Tq1Vmu0YygMcU09JLInJIWqTazWMT.Tzz4.JuatEL.n.teG9dN1B3k1JGsog
  4 | public         | $pbkdf2-sha512$25000$5pzznrMwo1TKGcP431vr/Q$FNFvS9GqZtGjGMnXH9p7cYBw/ZC.2Ewr/91Q/zeS/McGnWqnK5kevZihy1oskWUxb0sbrxMIF93Lbs.zblAMnQ
  5 | portaltemplate |
  7 | portal         | $pbkdf2-sha512$25000$TanVunfuHYNwjrH2vldqTQ$0kDjRv1bc02woJPZj/S3aFCOFmE9ZzhKsd8qu0roIIAEqvnXVY/z19mDgHtm63u5sTPeeGTTmWpidGUy9gCnPW
(7 rows)

odoo=>
```

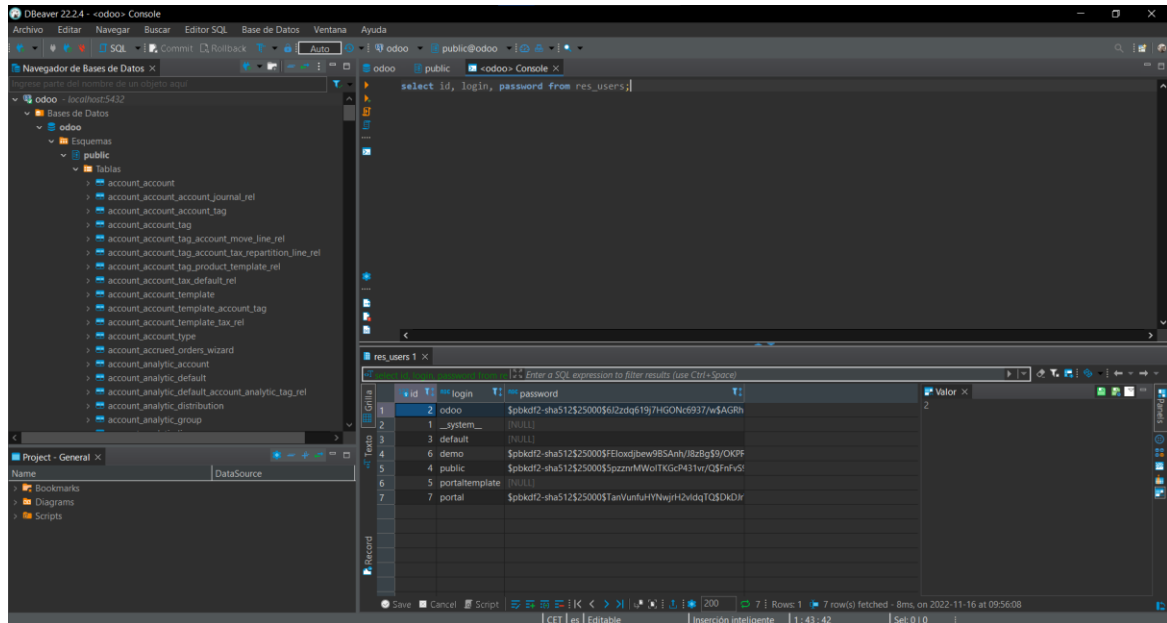
Para ver todos los productos ejecutamos la `Select` sobre la tabla `product_product`. Podemos filtrar las columnas para que podamos ver bien el contenido de la tabla solo con los datos que necesitamos.

```
odoo=> select id, default_code from product_product;
 id | default_code
-----+-----
  5 | FURN_7777
  6 | FURN_8888
 20 | E-COM10
  1 |
  3 |
  4 |
 38 | 556
 16 | E-COM06
 18 | E-COM08
 15 | DESK0004
 37 | DNF-PROD-0100
 40 |
 41 |
 13 | FURN_0097
 14 | FURN_0098
 24 | E-COM13
  7 | FURN_9999
  8 | FURN_7800
 12 | FURN_0096
 17 | E-COM07
 19 | E-COM09
 21 | E-COM11
 23 | E-COM12
 29 | FURN_7888
 30 | FURN_0789
 32 | FURN_8855
 33 | FURN_8220
 34 | FURN_6741
 35 | FURN_8999
 25 | FURN_0269
 36 | FURN_5555
 26 | FURN_1118
 27 | FURN_8900
 28 | FURN_9001
 31 | FURN_6666
 39 | 556-01
(36 rows)

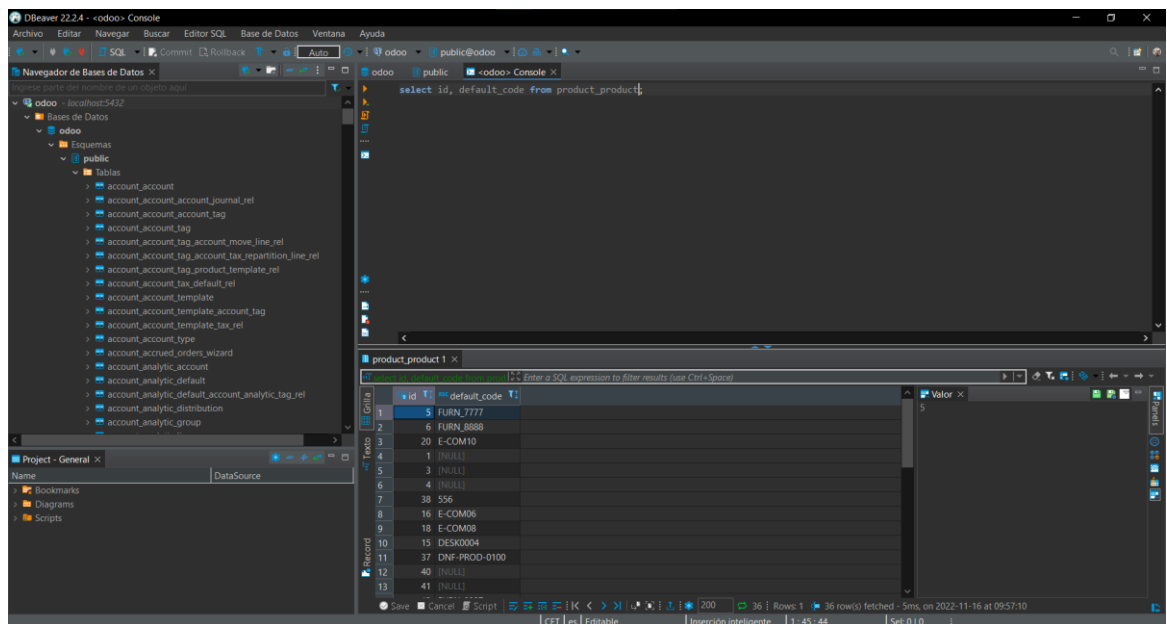
odoo=>
```

## 1.5. Realiza las mismas consultas mediante la herramienta o herramientas de gestión de la base de datos que hayas instalado (pgAdmin o DBeaver).

Nos vamos a Editor SQL una vez que estamos en este menú aquí hacemos click sobre Open SQL Console, se nos abre la consola y le pasamos la consulta para mostrar los usuarios.



Nos vamos a Editor SQL una vez que estamos en este menú aquí hacemos click sobre Open SQL Console, se nos abre la consola y le pasamos la consulta para mostrar los productos.



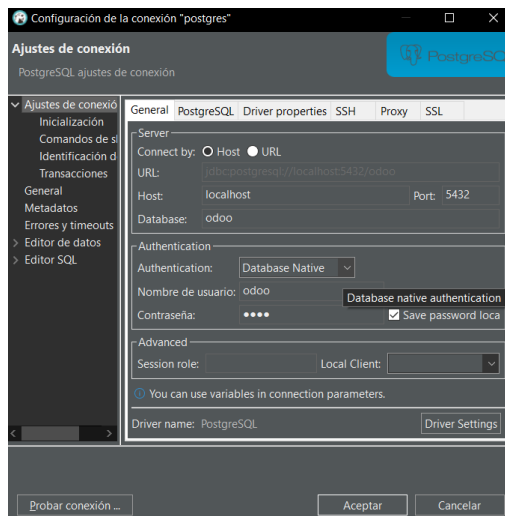
## 2. Docker

### 2.1. Conéctate por consola al contenedor de Docker de PostgreSQL y haz una consulta sencilla con psql.

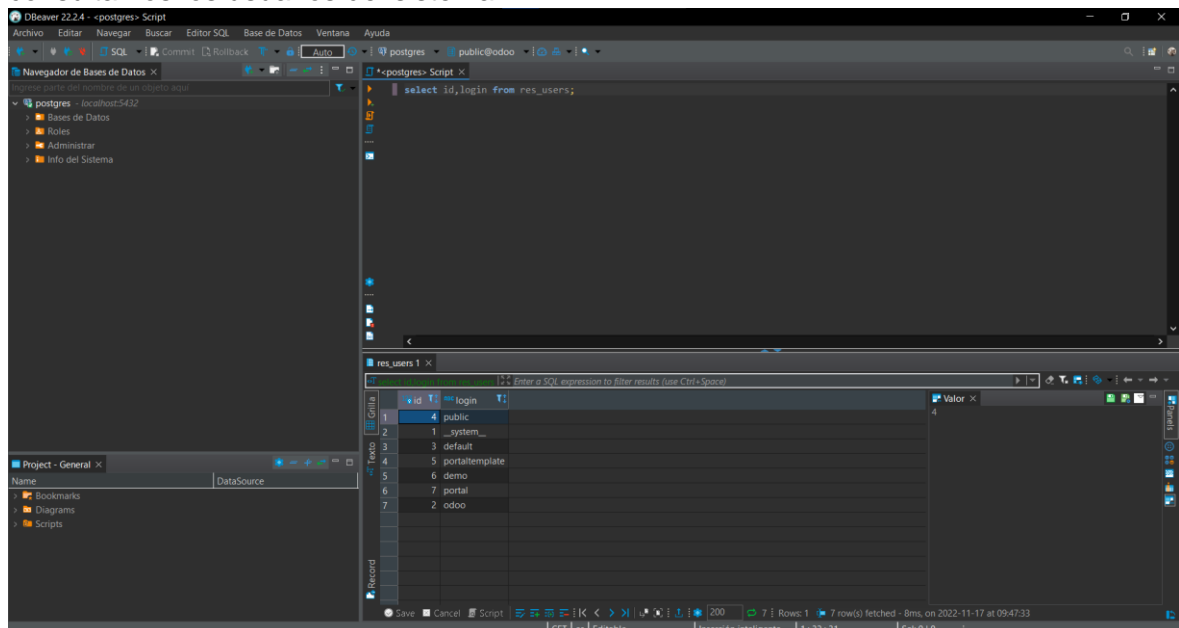
Ejecutamos dos contenedores, el primero será postgres y le asignamos el puerto en su caso es el 5432 y además le asignamos un volumen para que sus datos sean persistentes y no perdamos los datos, y el segundo contenedor será el de odoo le asignamos el puerto que será 8069 y además también le asignamos un volumen para que sus datos sean persistentes.

```
PS C:\Users\danie> docker run -d --name db postgres:13 -e POSTGRES_USER=odoo -e POSTGRES_PASSWORD=odoo -e POSTGRES_DB=postgres -p 5432:5432 -v odoo-db:/var/lib/postgresql/data
PS C:\Users\danie> docker run -v odoo-data:/var/lib/odoo -d -p 8069:8069 --name odoo --link db:db -t odoo
PS C:\Users\danie>
```

Realizamos la conexión le indicamos puerto que será 5432, nombre de la base de datos, nombre de usuario y contraseña



A continuación ejecutamos una consulta sencilla en la base de datos de odoo, en este caso consultamos los usuarios del sistema.



### 3. Conclusiones

Durante la práctica he podido las posibilidades que tengo para gestionar las bases de datos de postgre, tanto desde una maquina virtual como desde docker en contenedores.

He podido realizar copias de seguridad programadas sobre las bases de datos, hacer consultas psql sobre la base de datos.