



Odoo + WSL + Pycharm

Víctor Piles.

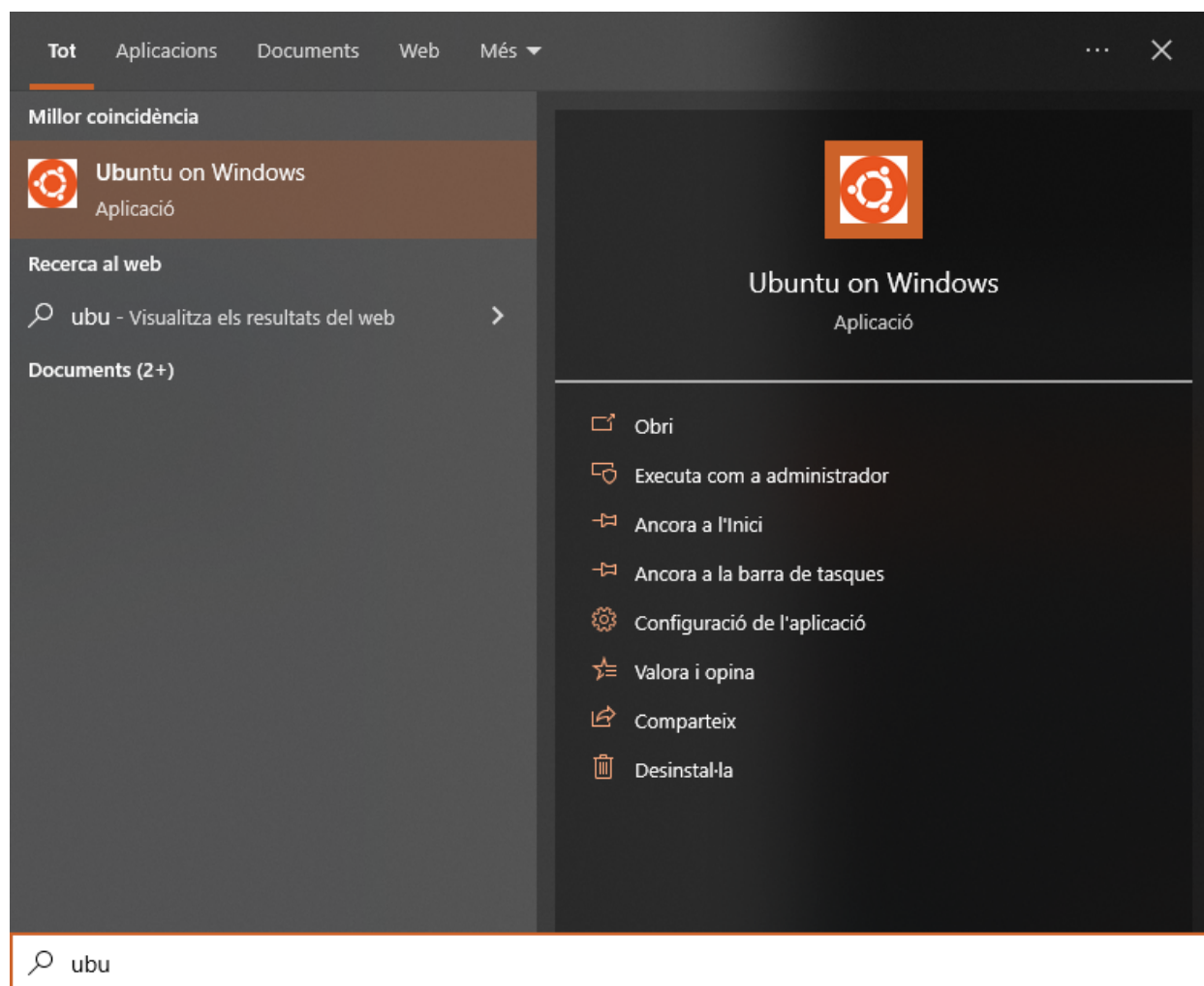
1) Instalación del WSL ([Microsoft Docs](#))

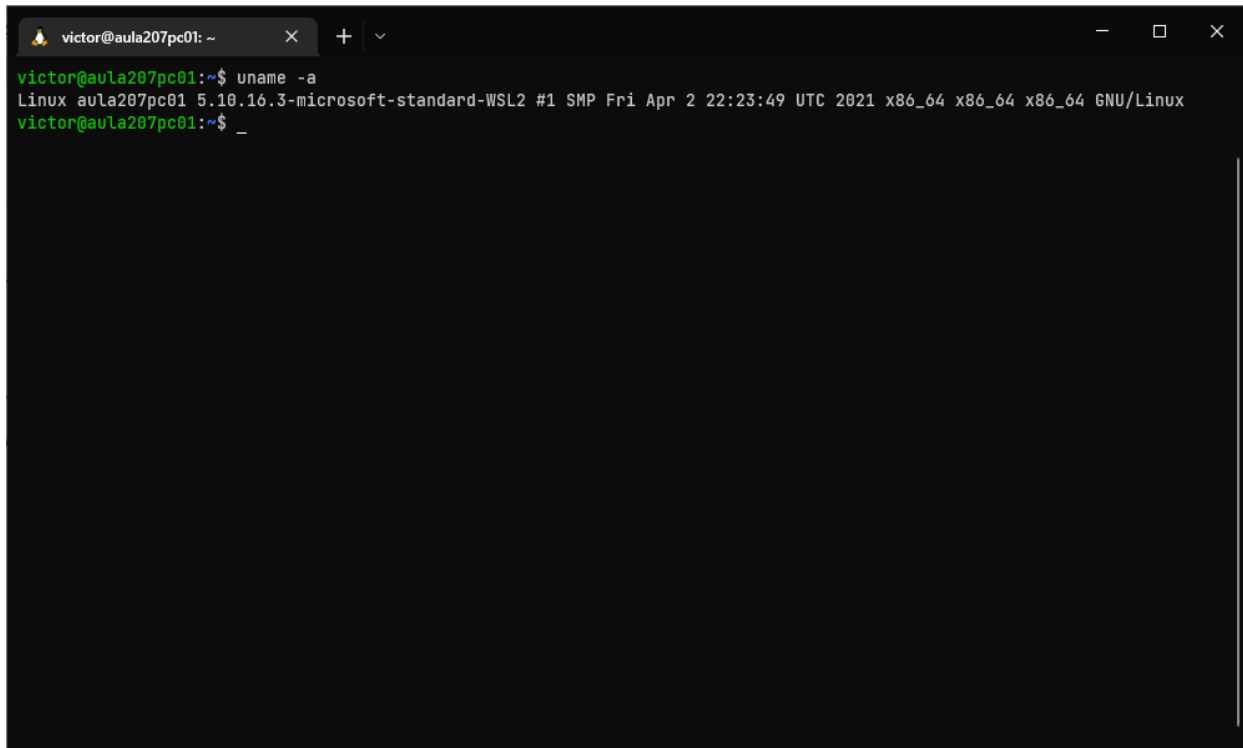
`wsl --install -d Ubuntu`

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\7J> wsl --install -d Ubuntu
Ubuntu ya está instalado.
Iniciando Ubuntu...
PS C:\Users\7J> _
```



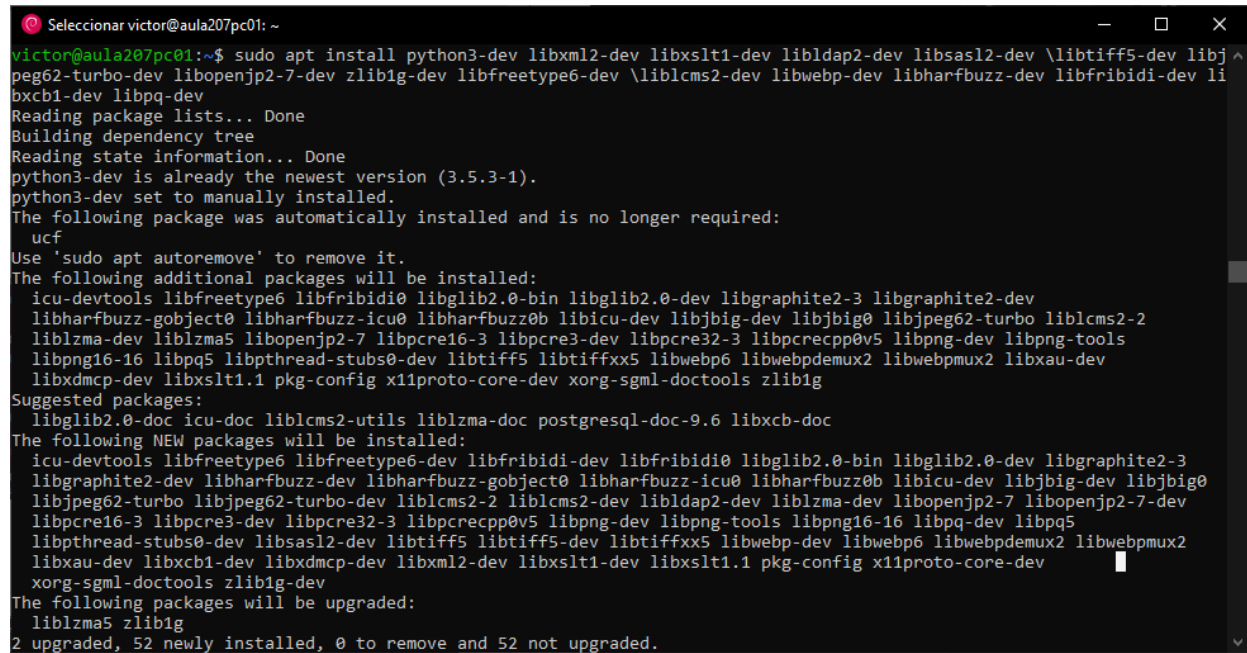
A terminal window with a dark background. The title bar shows 'victor@aula207pc01: ~' and standard window controls. The terminal text shows a user prompt 'victor@aula207pc01:~\$' followed by the command 'uname -a'. The output is 'Linux aula207pc01 5.10.16.3-microsoft-standard-WSL2 #1 SMP Fri Apr 2 22:23:49 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux'. The prompt returns to 'victor@aula207pc01:~\$' followed by a cursor line ' _\$'.

```
victor@aula207pc01:~$ uname -a
Linux aula207pc01 5.10.16.3-microsoft-standard-WSL2 #1 SMP Fri Apr 2 22:23:49 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
victor@aula207pc01:~$ _$
```

2) Instalación de Odoo ([Odoo Docs](#))

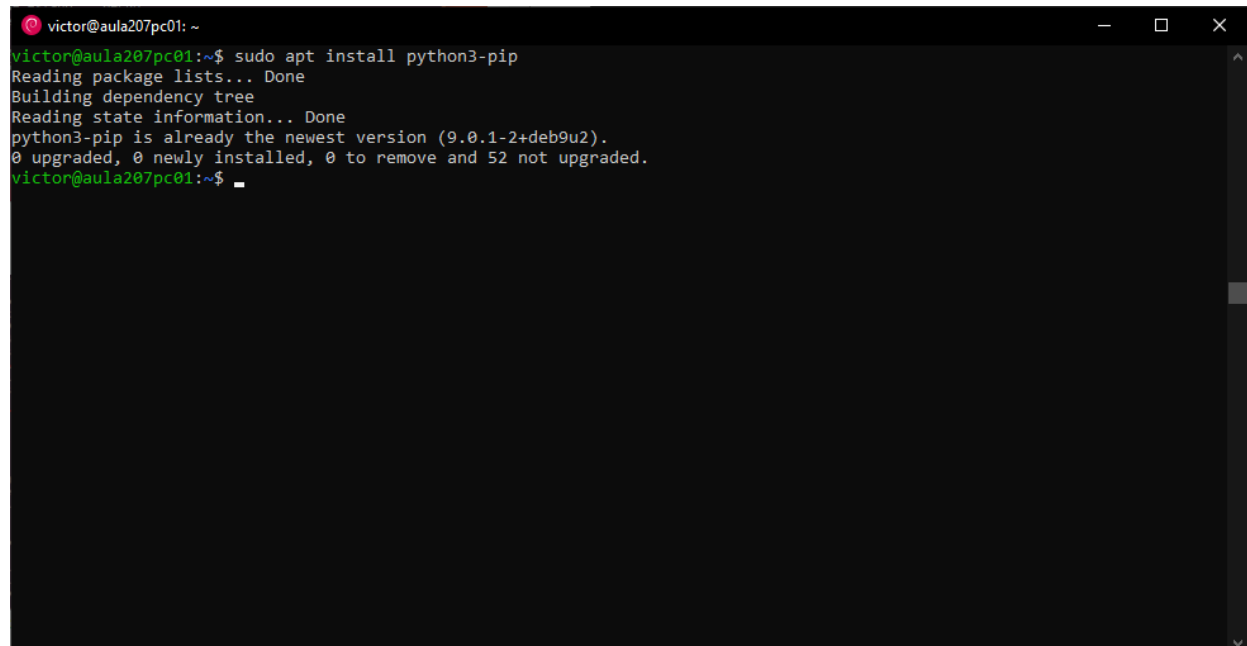
Instalamos las dependencias.

```
sudo apt install python3-dev libxml2-dev libxslt1-dev libldap2-dev  
libsasl2-dev libtiff5-dev libjpeg8-dev libopenjp2-7-dev zlib1g-dev  
libfreetype6-dev liblcms2-dev libwebp-dev libharfbuzz-dev libfribidi-dev  
libxcb1-dev libpq-dev
```



```
Seleccionar victor@aula207pc01: ~  
victor@aula207pc01:~$ sudo apt install python3-dev libxml2-dev libxslt1-dev libldap2-dev libsasl2-dev libtiff5-dev libjpeg62-turbo-dev libopenjp2-7-dev zlib1g-dev libfreetype6-dev liblcms2-dev libwebp-dev libharfbuzz-dev libfribidi-dev libxcb1-dev libpq-dev  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
python3-dev is already the newest version (3.5.3-1).  
python3-dev set to manually installed.  
The following package was automatically installed and is no longer required:  
  ucf  
Use 'sudo apt autoremove' to remove it.  
The following additional packages will be installed:  
  icu-devtools libfreetype6 libfribidi0 libglib2.0-bin libglib2.0-dev libgraphite2-3 libgraphite2-dev  
  libharfbuzz-gobject0 libharfbuzz-icu0 libharfbuzz0b libicu-dev libjbig-dev libjbig0 libjpeg62-turbo liblcms2-2  
  liblzma-dev liblzma5 libopenjp2-7 libpcre16-3 libpcre3-dev libpcre32-3 libpcrecpp0v5 libpng-dev libpng-tools  
  libpng16-16 libpq5 libpthread-stubs0-dev libtiff5 libtiffxx5 libwebp6 libwebpdemux2 libwebpmux2 libxau-dev  
  libxdmcp-dev libxslt1.1 pkg-config x11proto-core-dev xorg-sgml-doctools zlib1g  
Suggested packages:  
  libglib2.0-doc icu-doc liblcms2-utils liblzma-doc postgresql-doc-9.6 libxcb-doc  
The following NEW packages will be installed:  
  icu-devtools libfreetype6 libfreetype6-dev libfribidi-dev libfribidi0 libglib2.0-bin libglib2.0-dev libgraphite2-3  
  libgraphite2-dev libharfbuzz-dev libharfbuzz-gobject0 libharfbuzz-icu0 libharfbuzz0b libicu-dev libjbig-dev libjbig0  
  libjpeg62-turbo libjpeg62-turbo-dev liblcms2-2 liblcms2-dev libldap2-dev liblzma-dev libopenjp2-7 libopenjp2-7-dev  
  libpcre16-3 libpcre3-dev libpcre32-3 libpcrecpp0v5 libpng-dev libpng-tools libpng16-16 libpq-dev libpq5  
  libpthread-stubs0-dev libsasl2-dev libtiff5 libtiff5-dev libtiffxx5 libwebp-dev libwebp6 libwebpdemux2 libwebpmux2  
  libxau-dev libxcb1-dev libxdmcp-dev libxml2-dev libxslt1-dev libxslt1.1 pkg-config x11proto-core-dev  
  xorg-sgml-doctools zlib1g-dev  
The following packages will be upgraded:  
  liblzma5 zlib1g  
2 upgraded, 52 newly installed, 0 to remove and 52 not upgraded.
```

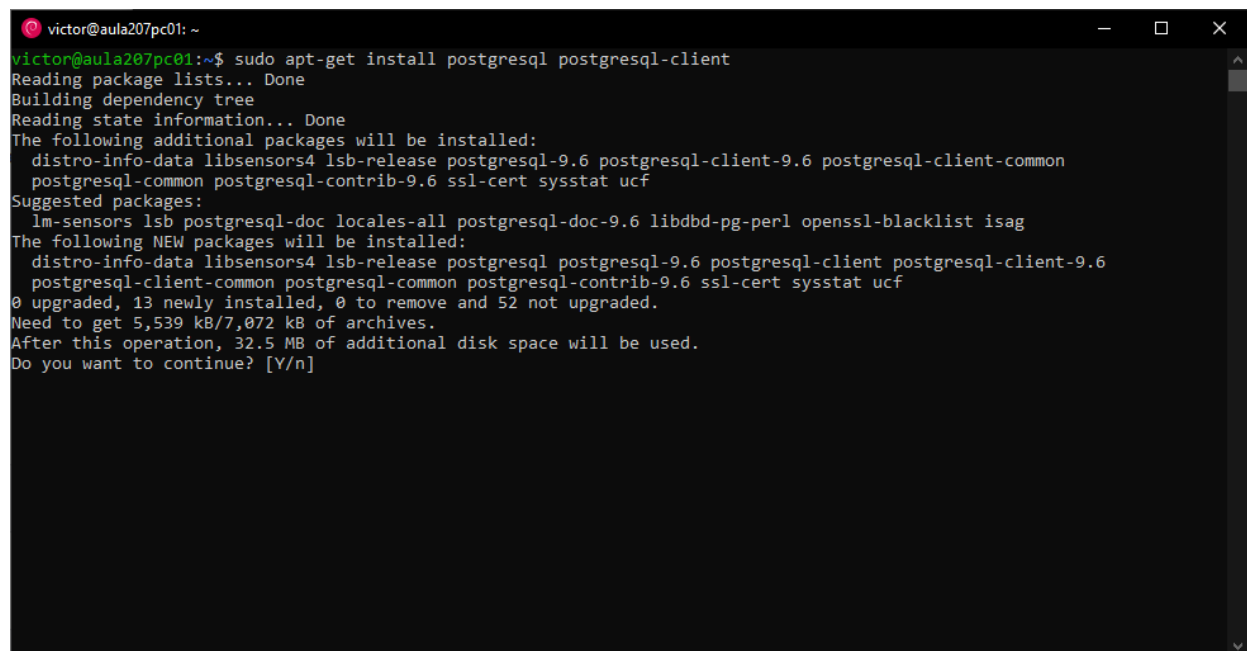
```
sudo apt install python3-pip
```



```
victor@aula207pc01: ~  
victor@aula207pc01:~$ sudo apt install python3-pip  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
python3-pip is already the newest version (9.0.1-2+deb9u2).  
0 upgraded, 0 newly installed, 0 to remove and 52 not upgraded.  
victor@aula207pc01:~$
```

PostgreSQL

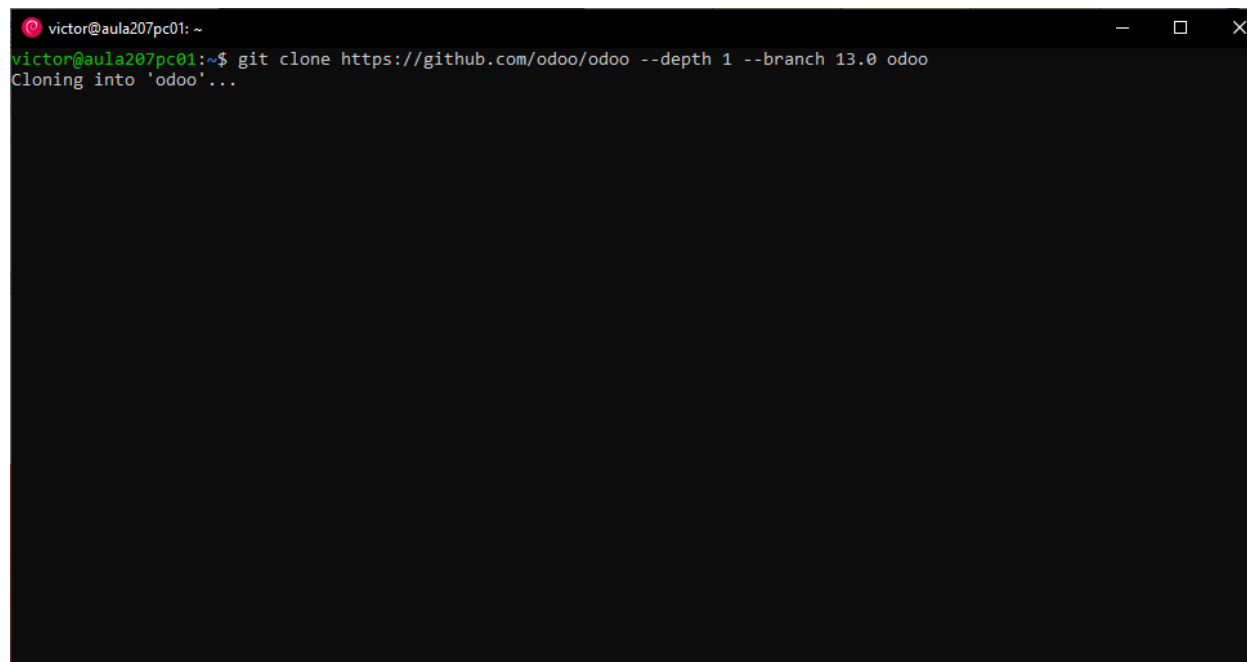
```
sudo apt-get install postgresql postgresql-client
```

A terminal window with a dark background and light text. The prompt is 'victor@aula207pc01: ~'. The user enters 'sudo apt-get install postgresql postgresql-client'. The terminal shows the progress of the installation, including reading package lists, building a dependency tree, and reading state information. It lists additional packages to be installed, suggested packages, and the new packages to be installed. It also shows the disk space requirements and asks for confirmation to continue.

```
victor@aula207pc01: ~  
victor@aula207pc01:~$ sudo apt-get install postgresql postgresql-client  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  distro-info-data libsensors4 lsb-release postgresql-9.6 postgresql-client-9.6 postgresql-client-common  
  postgresql-common postgresql-contrib-9.6 ssl-cert sysstat ucf  
Suggested packages:  
  lm-sensors lsb postgresql-doc locales-all postgresql-doc-9.6 libdbd-pg-perl openssl-blacklist isag  
The following NEW packages will be installed:  
  distro-info-data libsensors4 lsb-release postgresql postgresql-9.6 postgresql-client postgresql-client-9.6  
  postgresql-client-common postgresql-common postgresql-contrib-9.6 ssl-cert sysstat ucf  
0 upgraded, 13 newly installed, 0 to remove and 52 not upgraded.  
Need to get 5,539 kB/7,072 kB of archives.  
After this operation, 32.5 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

Clonamos el repositorio de Odoo.

```
git clone https://github.com/odoo/odoo --depth 1 --branch 13.0 odoo
```

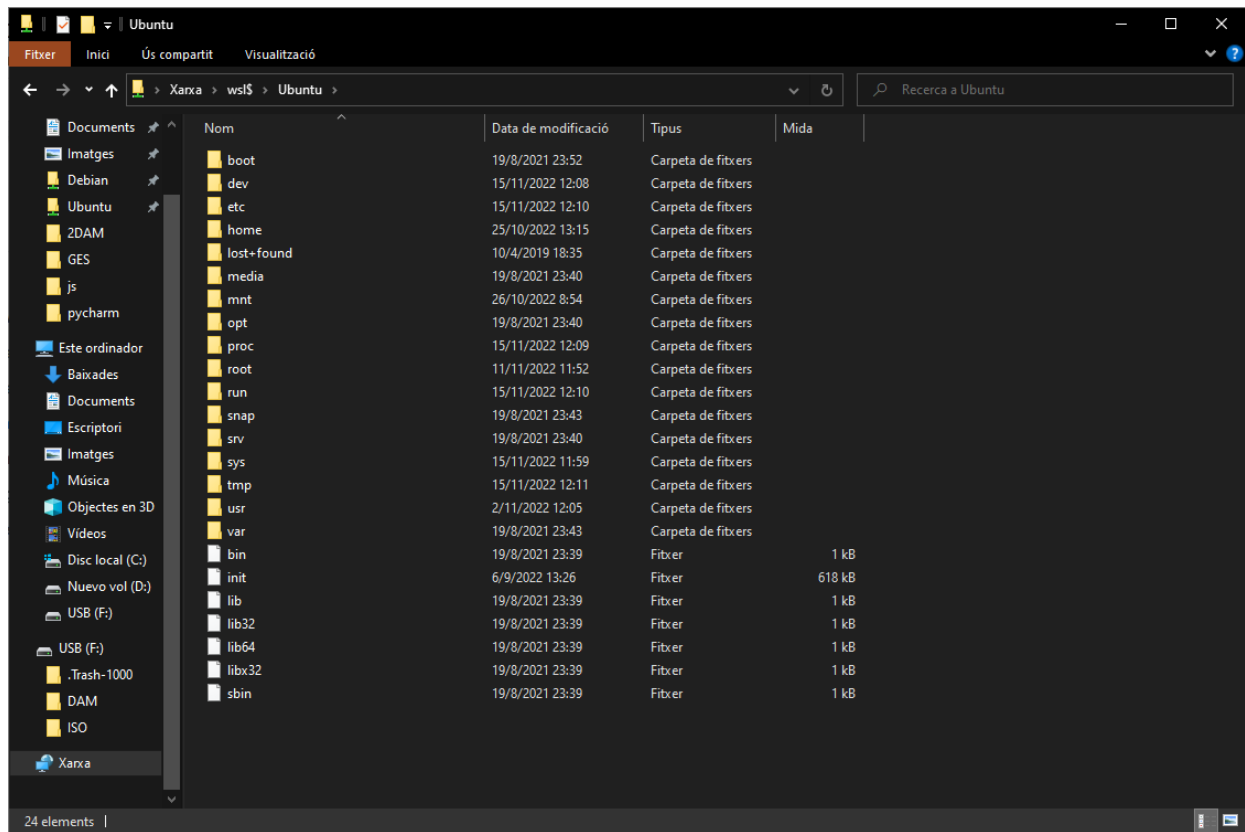
A terminal window with a dark background and light text. The prompt is 'victor@aula207pc01: ~'. The user enters 'git clone https://github.com/odoo/odoo --depth 1 --branch 13.0 odoo'. The terminal shows the progress of cloning the repository, including the message 'Cloning into 'odoo'...' and a progress bar.

```
victor@aula207pc01: ~  
victor@aula207pc01:~$ git clone https://github.com/odoo/odoo --depth 1 --branch 13.0 odoo  
Cloning into 'odoo'...
```

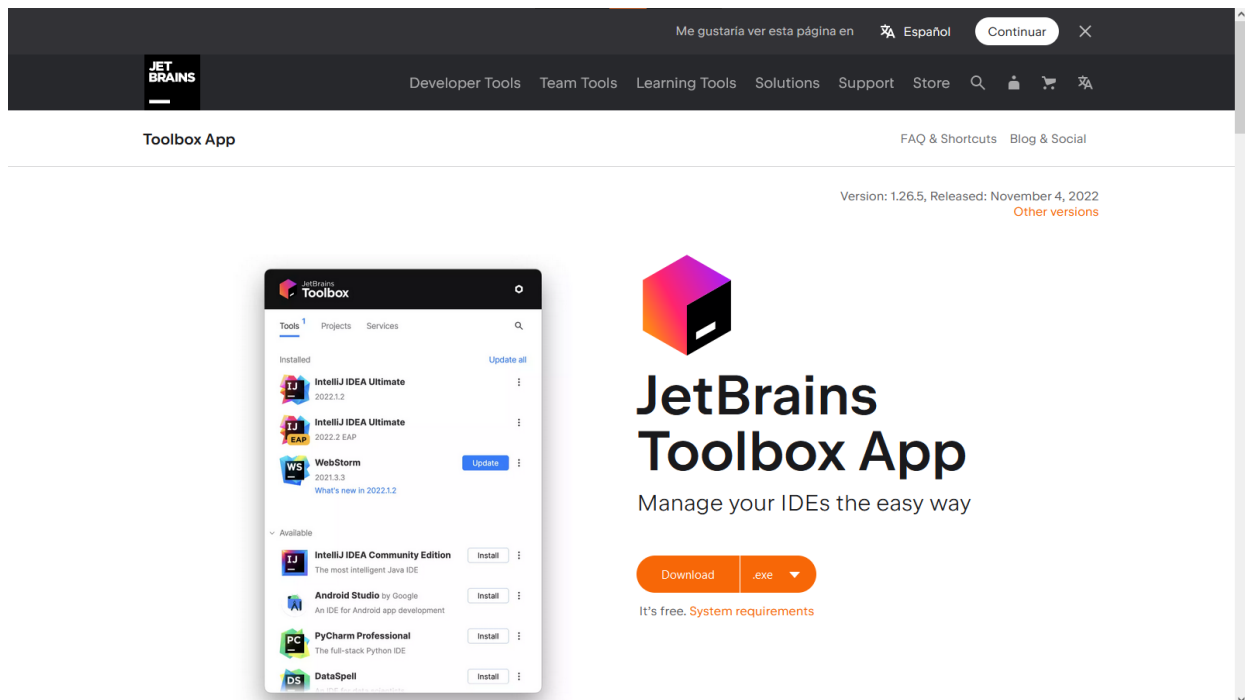
3) Pycharm en Windows 10

Desde el explorador de archivos podemos acceder a los ficheros de Ubuntu WSL.

\\wsl\$\Ubuntu

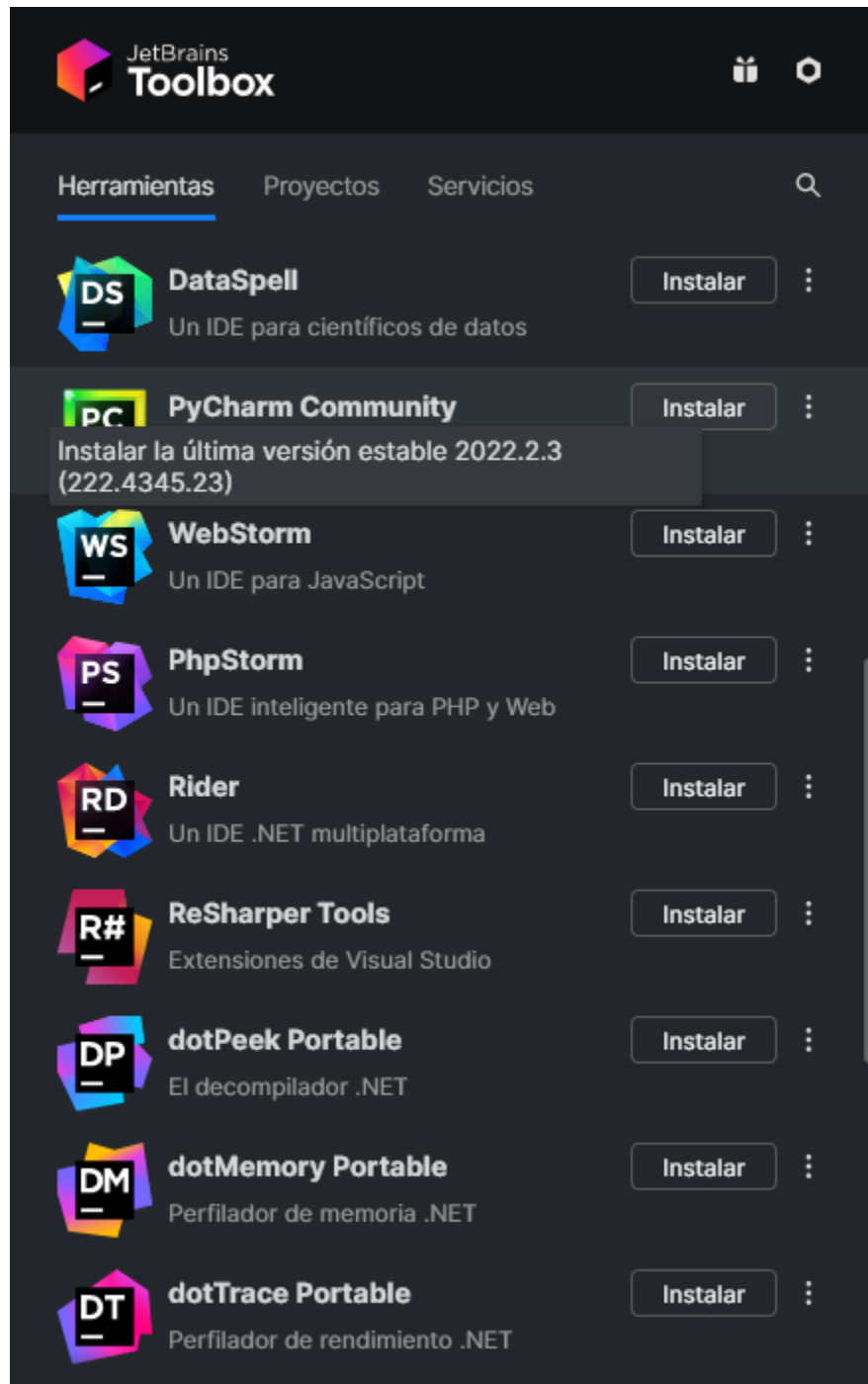


Nos bajamos JetBrains Toolbox desde [su página](#) y lo instalamos.

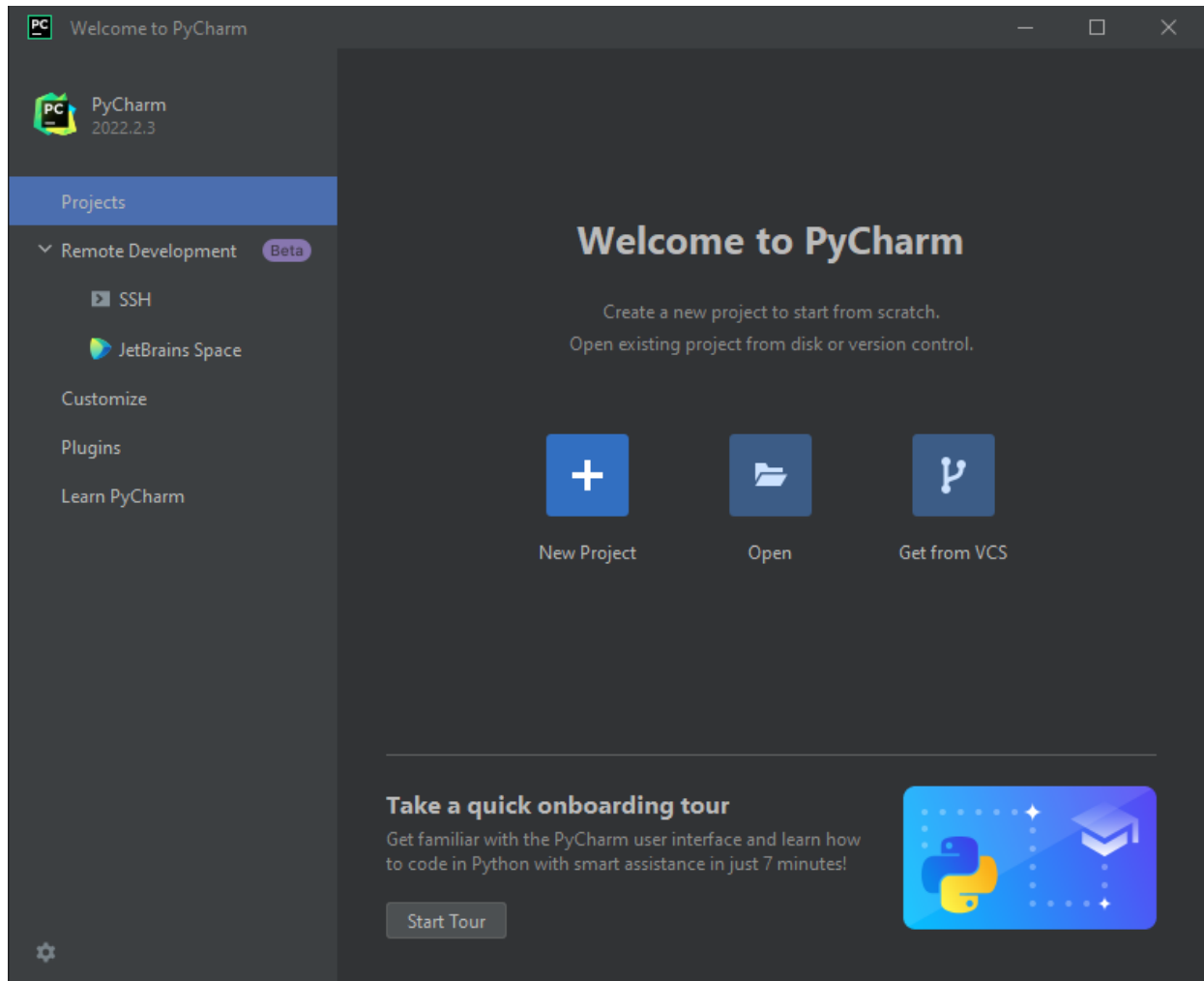


Desde la app podemos gestionar e instalar diferentes versiones de los IDE de JetBrains.

Instalamos Pycharm Community.

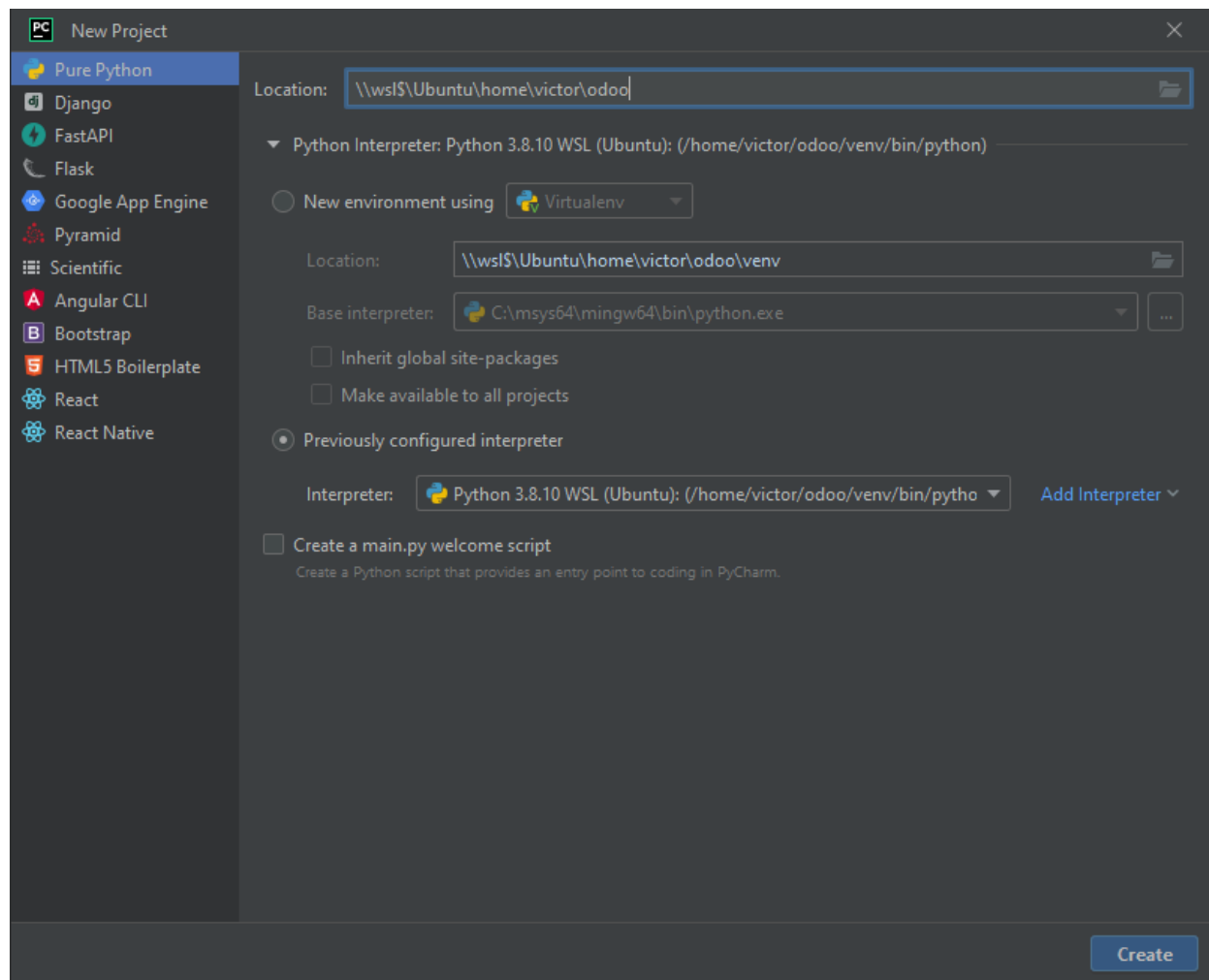


Abrimos PyCharm y le damos a "New Project".

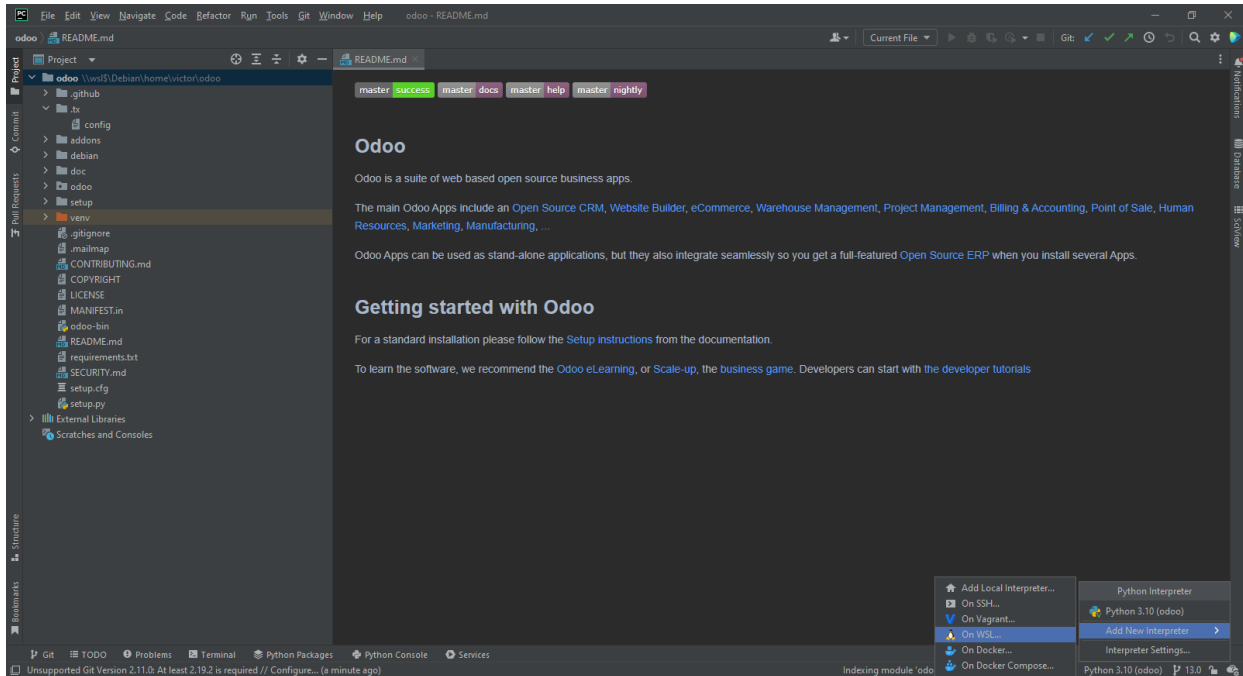


Creamos el proyecto en la carpeta “odoo” dentro de “home” en Ubuntu.

\\wsl\$\Ubuntu\home\vector\odoo



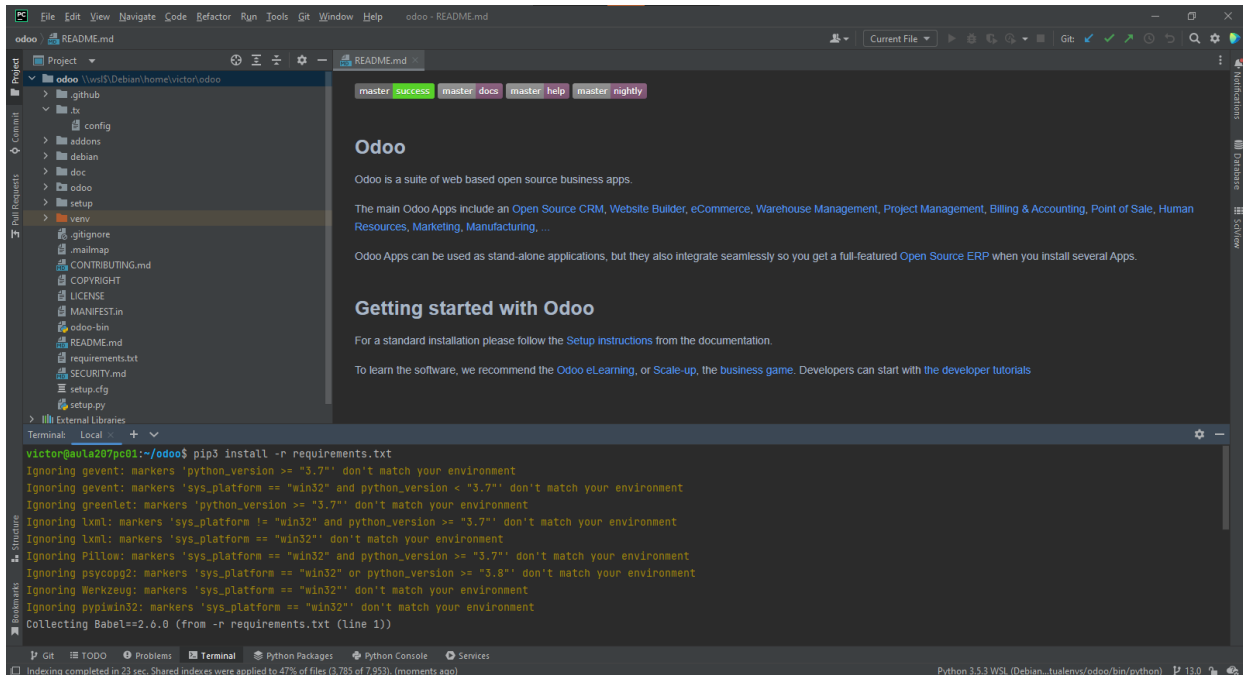
Configuramos el intérprete de Python para que use el del “virtualenv” WSL.



Instalamos las dependencias del proyecto.

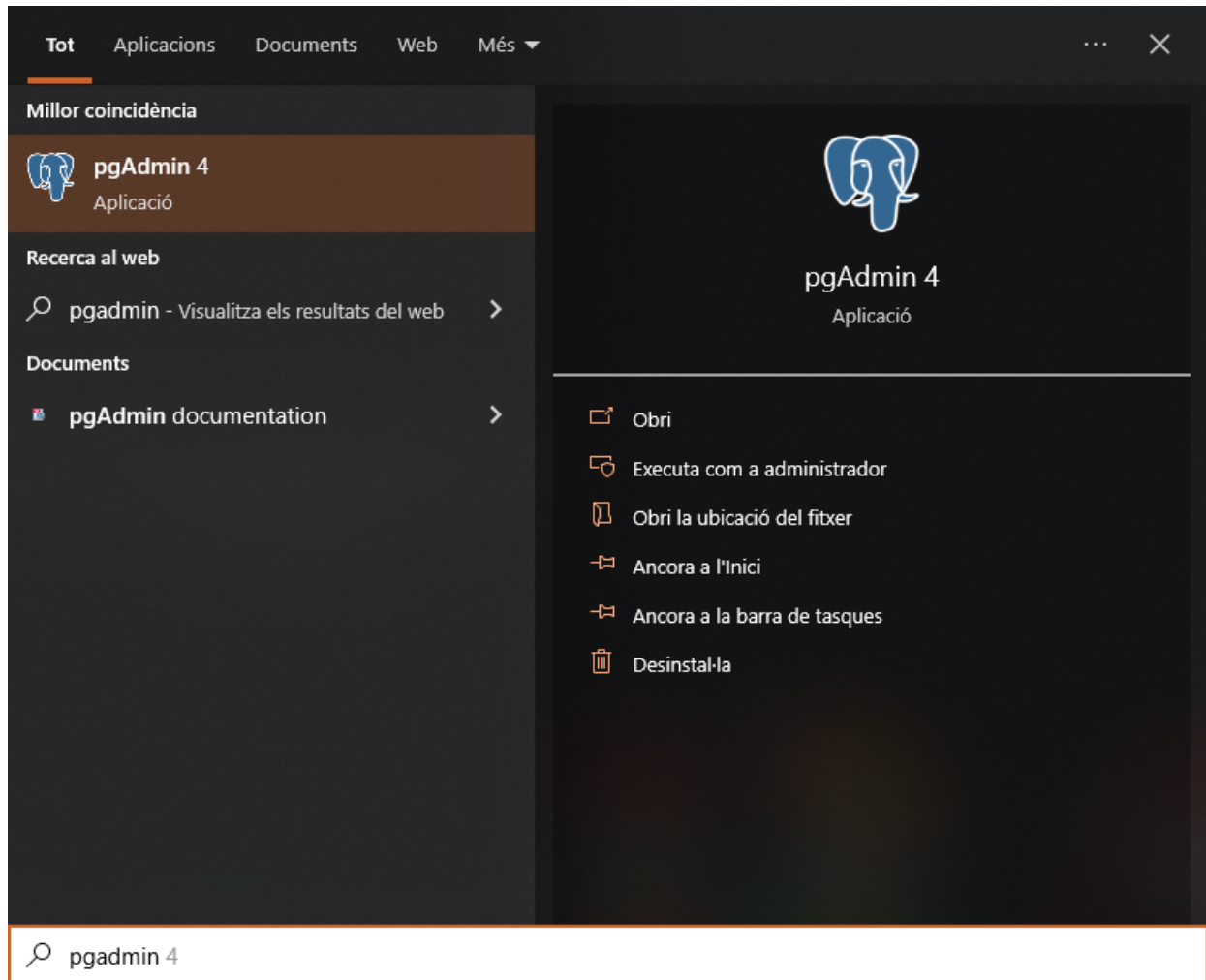
```
pip3 install -r requirements.txt
```

Nótese que el terminal que se me ha abierto es el de WSL, no el cmd de Windows. Lo hace Pycharm de forma automática.



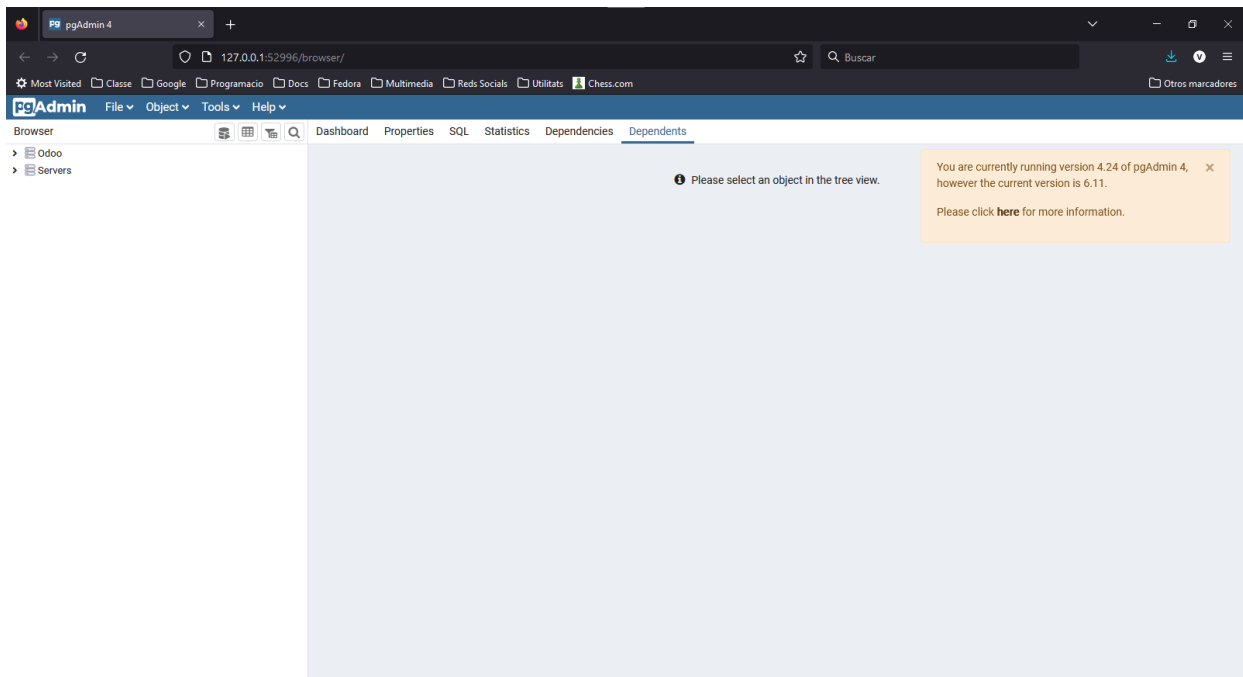
4) PGAdmin Windows 10

Descargamos PGAdmin desde su [página oficial](#) e instalamos. Después lo abrimos.



Se ejecuta en el navegador.

http://127.0.0.1:52996/browser/



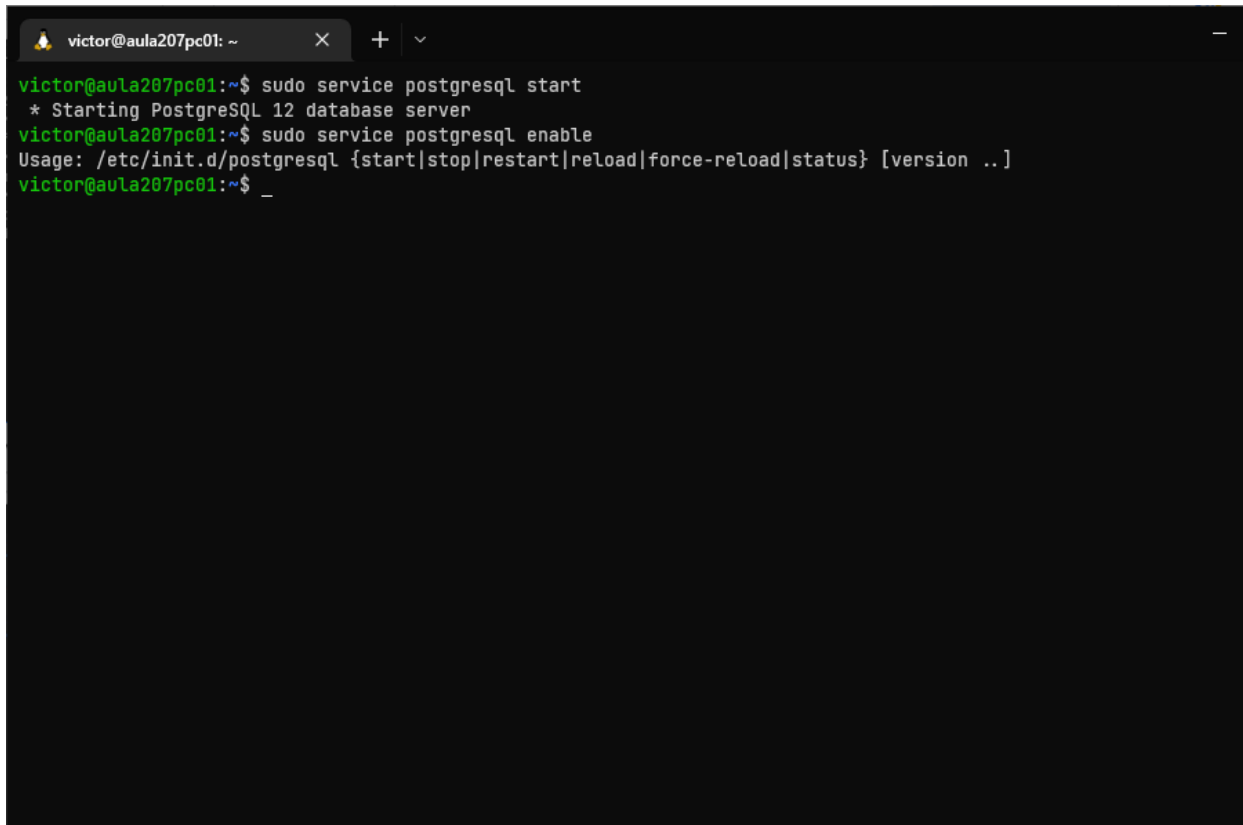
Ahora nos vamos a conectar a la base de datos del subsistema.

Encontraremos la IP del subsistema en el terminal.

ip a

```
victor@aula207pc01: ~  
victor@aula207pc01:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: bond0: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN group default qlen 1000  
    link/ether ee:b1:22:fb:b5:af brd ff:ff:ff:ff:ff:ff  
3: dummy0: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 1000  
    link/ether fe:e5:ed:bb:01:5b brd ff:ff:ff:ff:ff:ff  
4: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000  
    link/ether 00:15:5d:a8:c0:0a brd ff:ff:ff:ff:ff:ff  
    inet 172.19.231.29/20 brd 172.19.239.255 scope global eth0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::215:5dff:fea8:c00a/64 scope link  
        valid_lft forever preferred_lft forever  
5: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000  
    link/ipip 0.0.0.0 brd 0.0.0.0  
6: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000  
    link/sit 0.0.0.0 brd 0.0.0.0  
victor@aula207pc01:~$ _
```

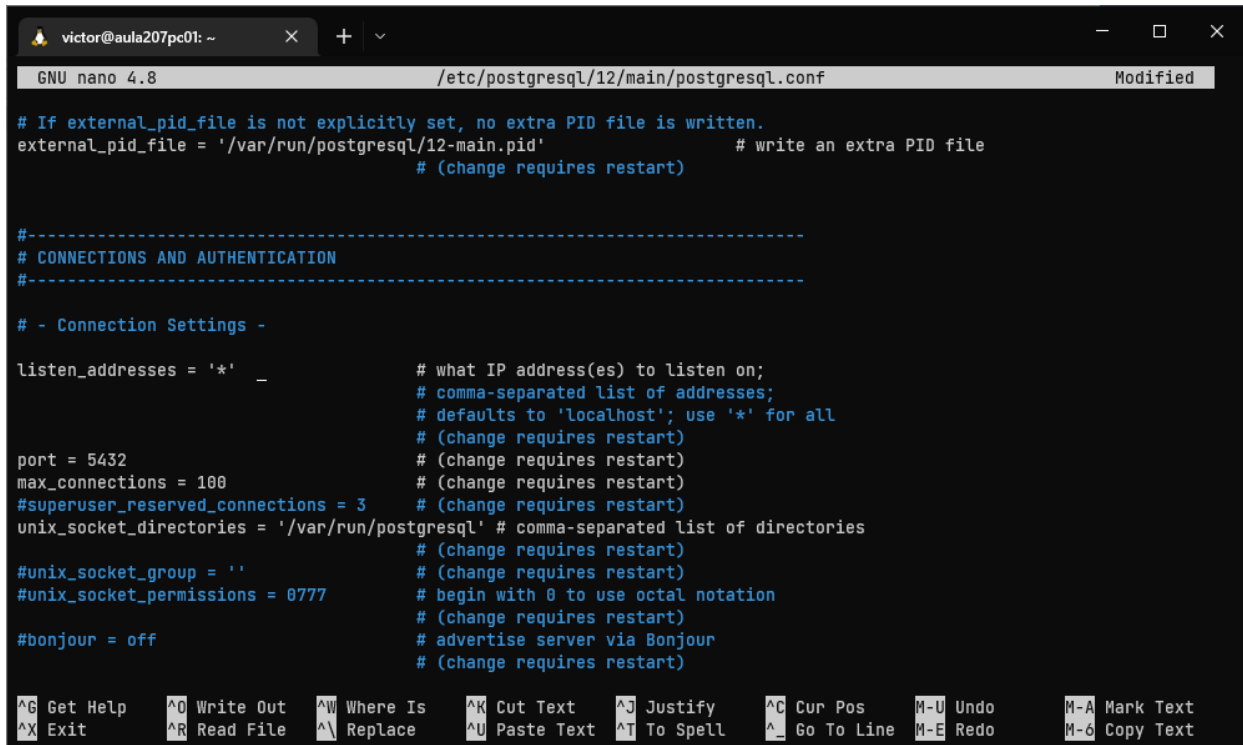
Iniciamos el servidor Postgres en Ubuntu y creamos el usuario odoo (contraseña odoo).

A terminal window with a dark background. The title bar shows 'victor@aula207pc01: ~' with window control buttons. The terminal text is as follows:

```
victor@aula207pc01:~$ sudo service postgresql start
* Starting PostgreSQL 12 database server
victor@aula207pc01:~$ sudo service postgresql enable
Usage: /etc/init.d/postgresql {start|stop|restart|reload|force-reload|status} [version ..]
victor@aula207pc01:~$ _
```

Modificamos los permisos de Postgres para poder conectarse de forma remota.

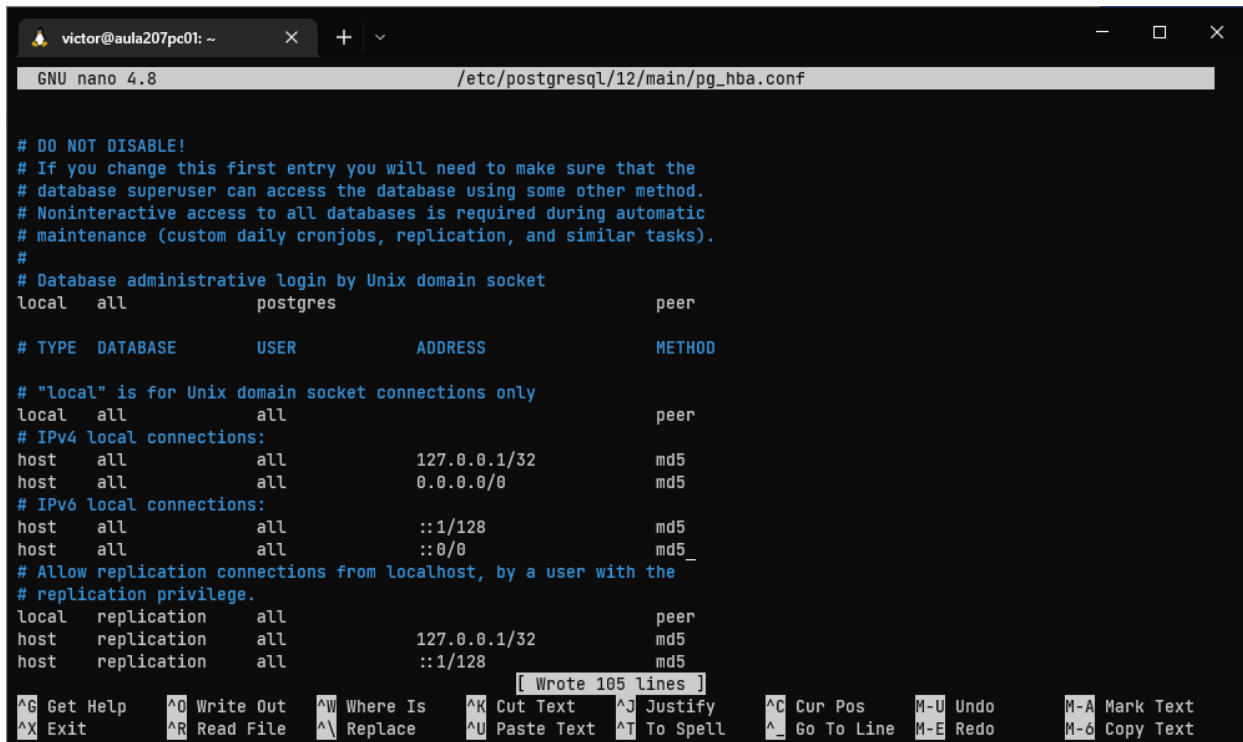
```
sudo nano /etc/postgresql/12/main/postgresql.conf  
listen_addresses = '*'
```



```
GNU nano 4.8 /etc/postgresql/12/main/postgresql.conf Modified  
  
# If external_pid_file is not explicitly set, no extra PID file is written.  
external_pid_file = '/var/run/postgresql/12-main.pid'           # write an extra PID file  
                                                                # (change requires restart)  
  
#-----  
# CONNECTIONS AND AUTHENTICATION  
#-----  
  
# - Connection Settings -  
  
listen_addresses = '*'                                          # 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)  
  
^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo      M-A Mark Text  
^X Exit      ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line  M-E Redo      M-^ Copy Text
```

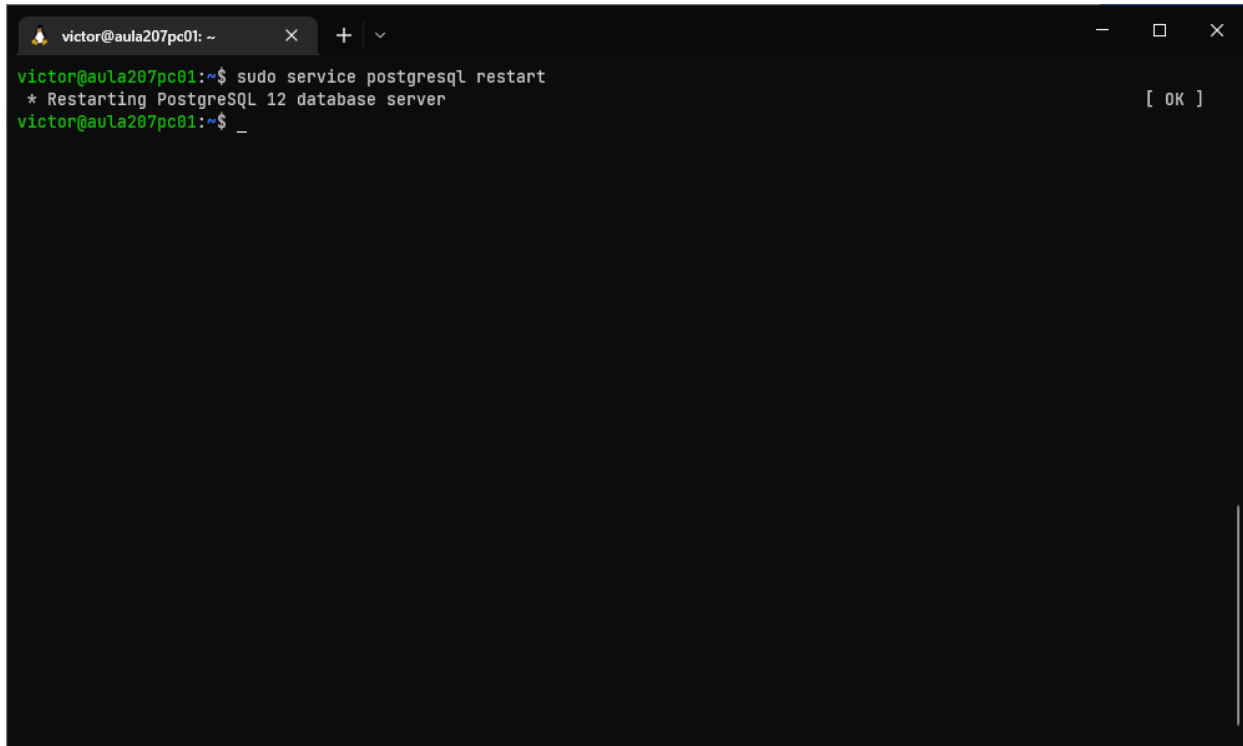
Añadimos la IP 0.0.0.0/0 tanto en IPv4 como en IPv6.

```
sudo nano /etc/postgresql/12/main/pg_hba.conf
```



```
GNU nano 4.8 /etc/postgresql/12/main/pg_hba.conf  
  
# DO NOT DISABLE!  
# If you change this first entry you will need to make sure that the  
# database superuser can access the database using some other method.  
# Noninteractive access to all databases is required during automatic  
# maintenance (custom daily cronjobs, replication, and similar tasks).  
#  
# Database administrative login by Unix domain socket  
local all postgres peer  
  
# TYPE DATABASE USER ADDRESS METHOD  
  
# "local" is for Unix domain socket connections only  
local all all peer  
# IPv4 local connections:  
host all all 127.0.0.1/32 md5  
host all all 0.0.0.0/0 md5  
# IPv6 local connections:  
host all all ::1/128 md5  
host all all ::0/0 md5  
# Allow replication connections from localhost, by a user with the  
# replication privilege.  
local replication all peer  
host replication all 127.0.0.1/32 md5  
host replication all ::1/128 md5  
[ Wrote 105 lines ]  
  
^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo      M-A Mark Text  
^X Exit      ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line  M-E Redo      M-^ Copy Text
```

Reiniciamos el servicio.

A terminal window with a dark background and light green text. The window title bar shows 'victor@aula207pc01: ~' and standard window controls. The terminal content shows a user prompt 'victor@aula207pc01:~\$' followed by the command 'sudo service postgresql restart'. The output is '* Restarting PostgreSQL 12 database server' followed by a '[OK]' status message. The prompt returns to 'victor@aula207pc01:~\$' with a cursor.

```
victor@aula207pc01:~$ sudo service postgresql restart
* Restarting PostgreSQL 12 database server [ OK ]
victor@aula207pc01:~$ _
```

Nos conectamos desde PGAdmin.

Create - Server

General

Connection

SSL

SSH Tunnel

Advanced

Host name/address

172.19.231.29

Port

5432

Maintenance database

postgres

Username

odoo

Password

••••

Save password?

☒

Role

Service

i

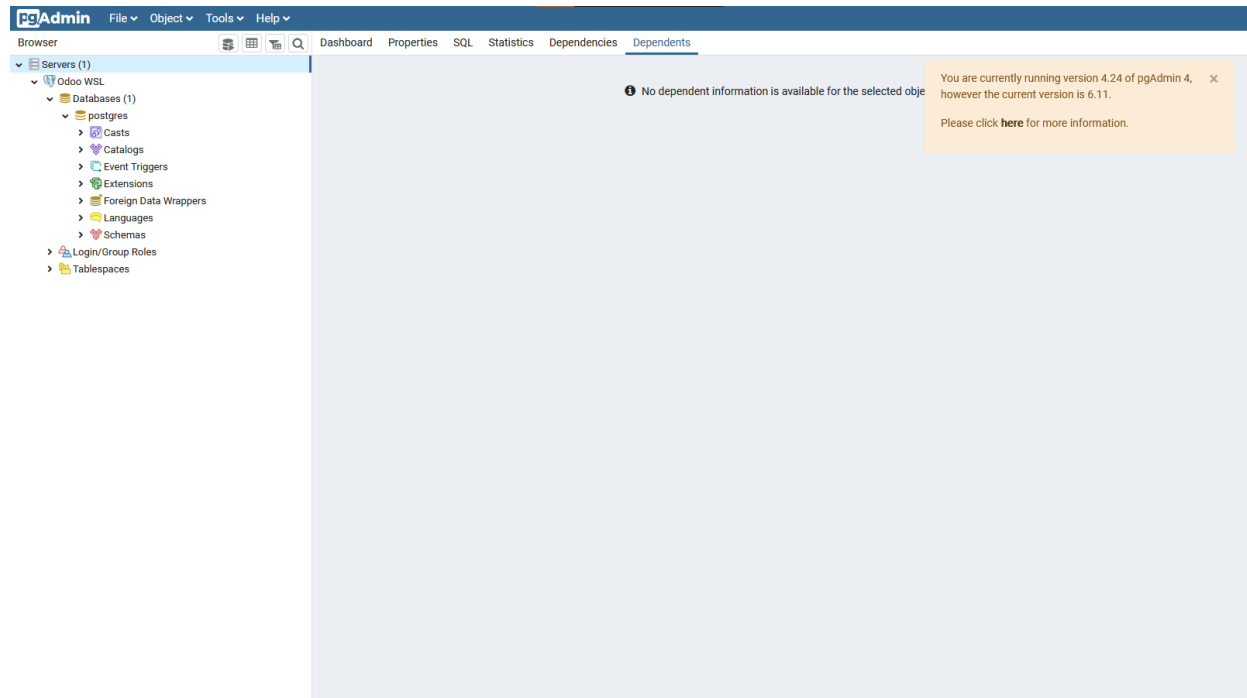
?

✕ Cancel

↺ Reset

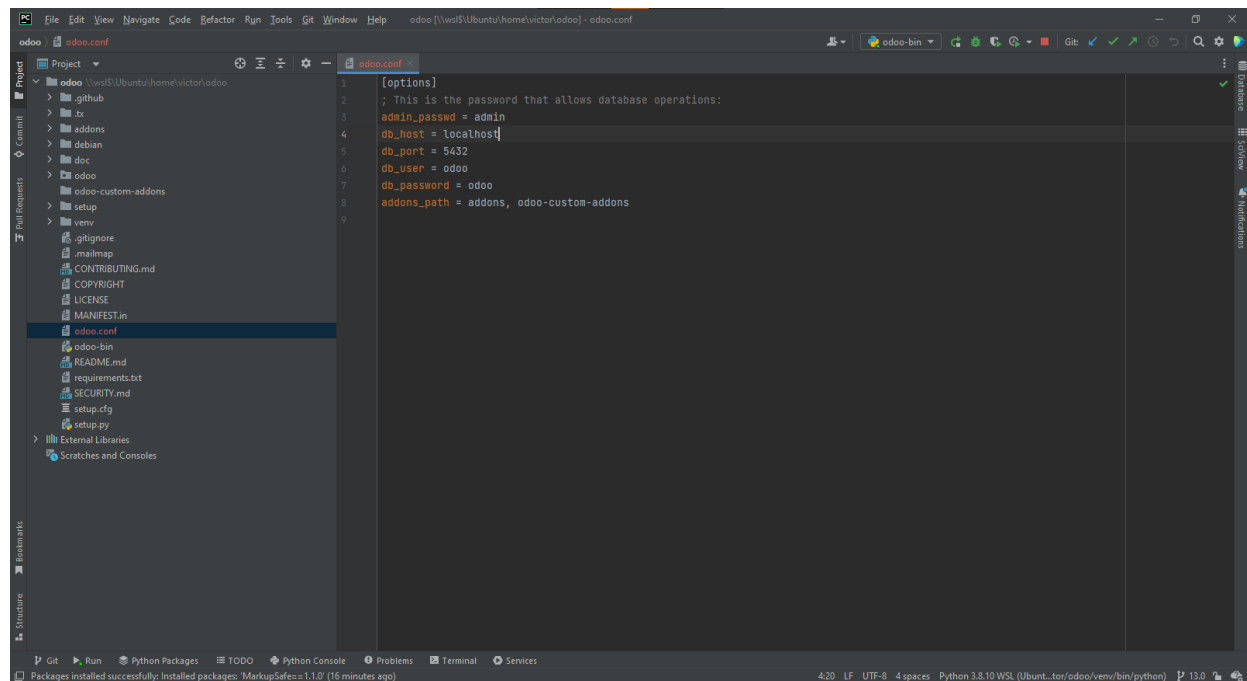
💾 Save

Se conecta correctamente.

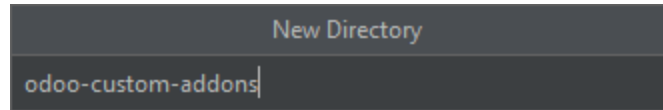


5) Configurar la base de datos en el proyecto

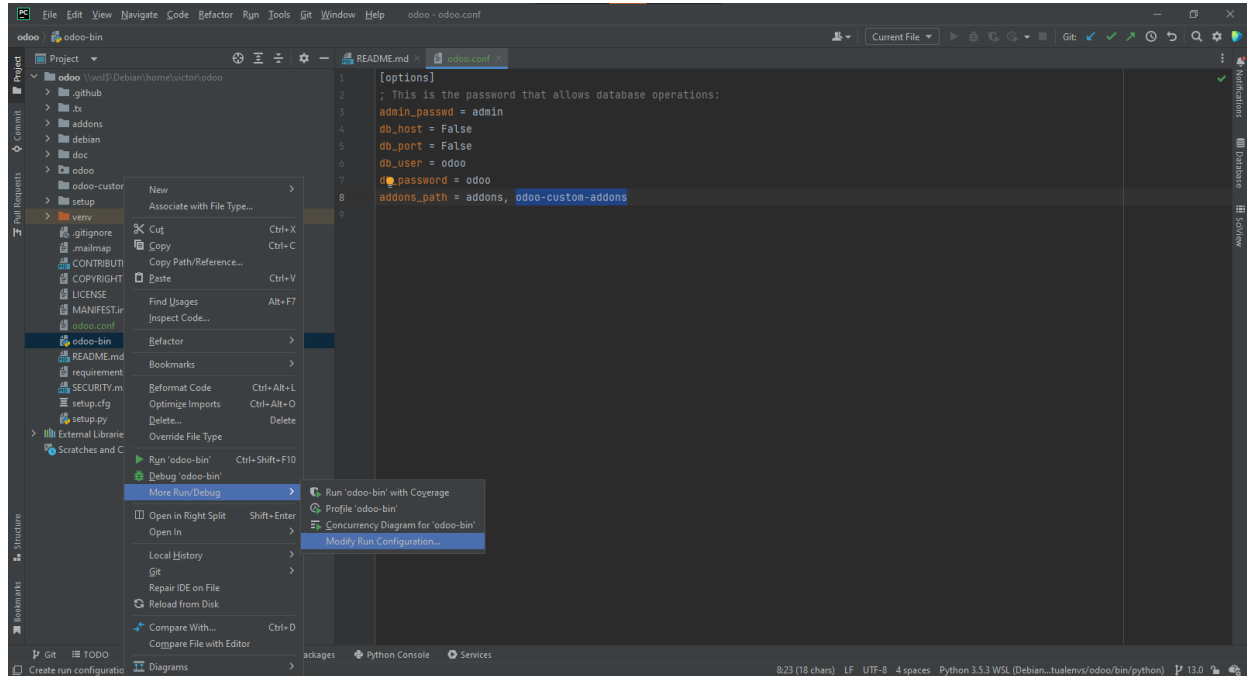
Creamos odoo.conf en el directorio raíz y configuramos los parámetros.



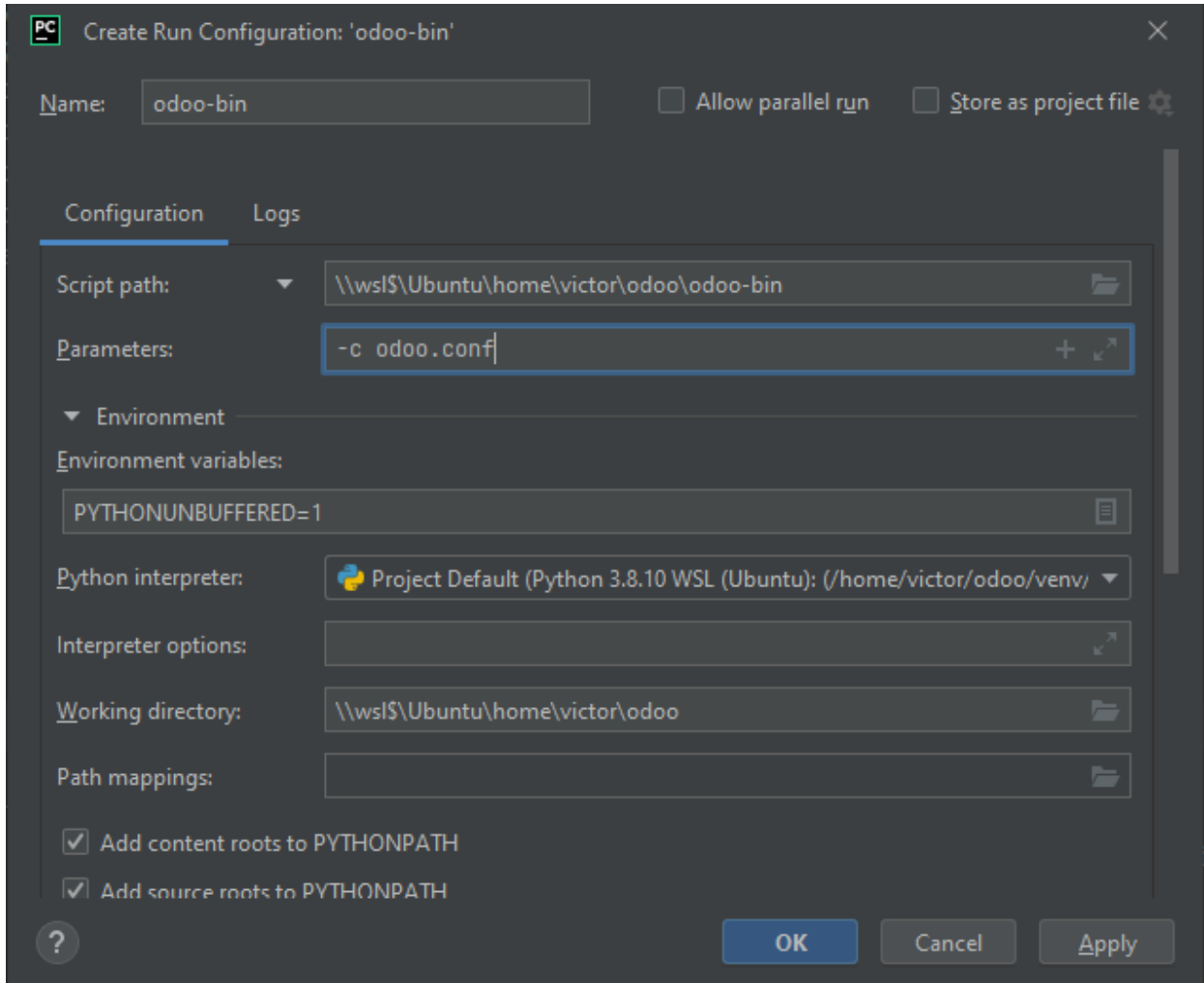
Creamos el directorio para las extensiones personalizadas.



Modificamos los parámetros de ejecución.

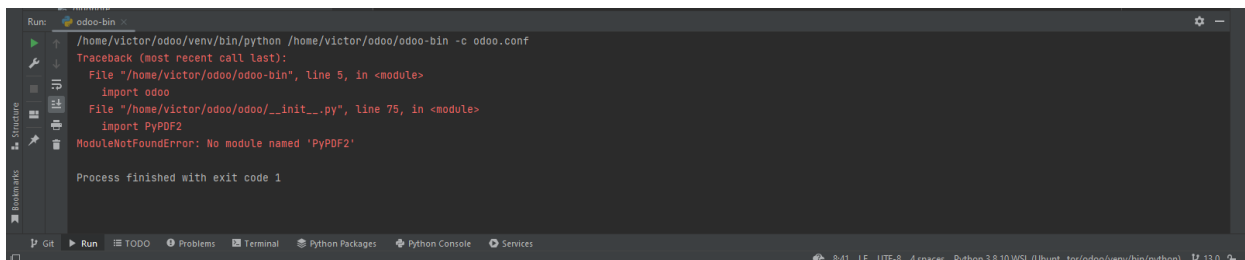


Añadimos los parámetros.



6) Errores

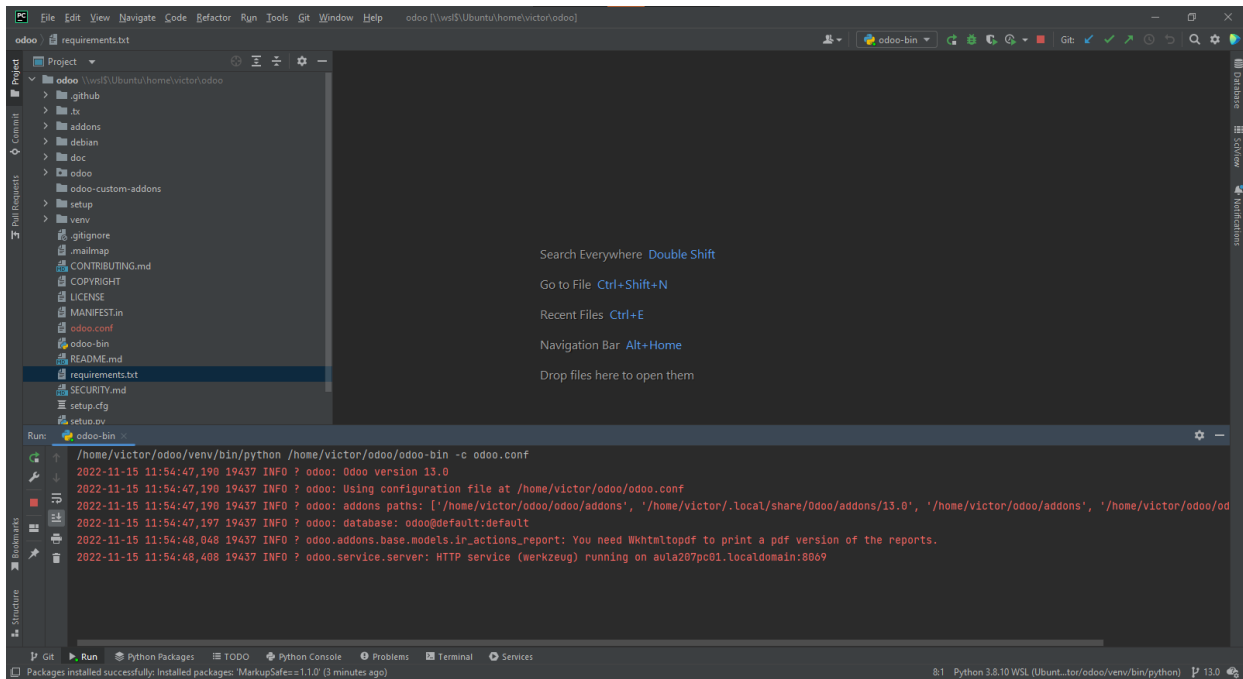
Me da un error porque me falta una dependencia.



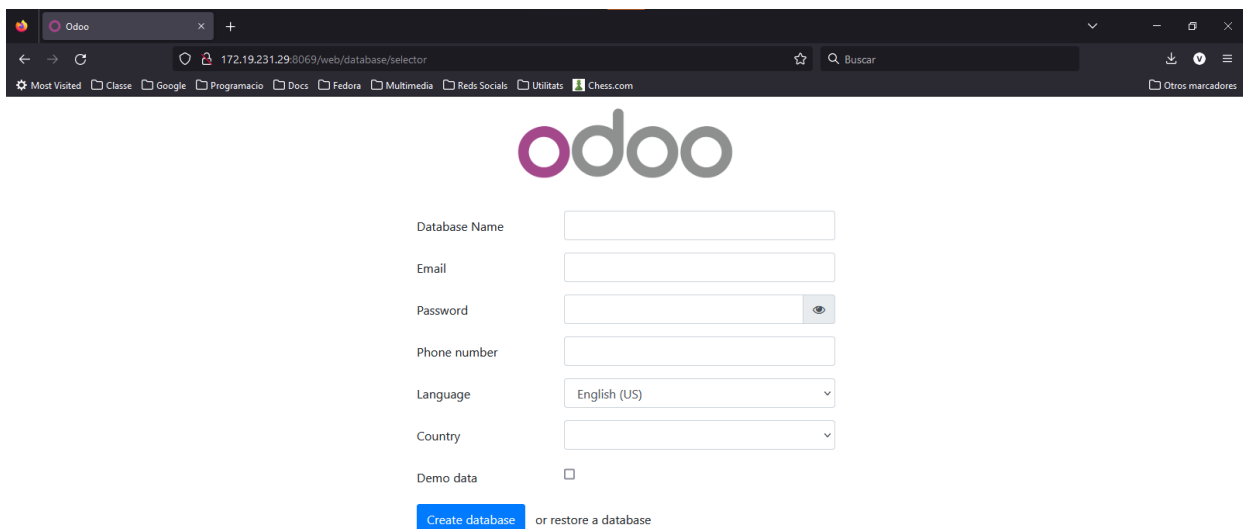
He encontrado la solución [aquí](#).

7) Lanzar programa

Damos click en el botón de ejecutar.






Accedemos a través del navegador con IP:puerto.

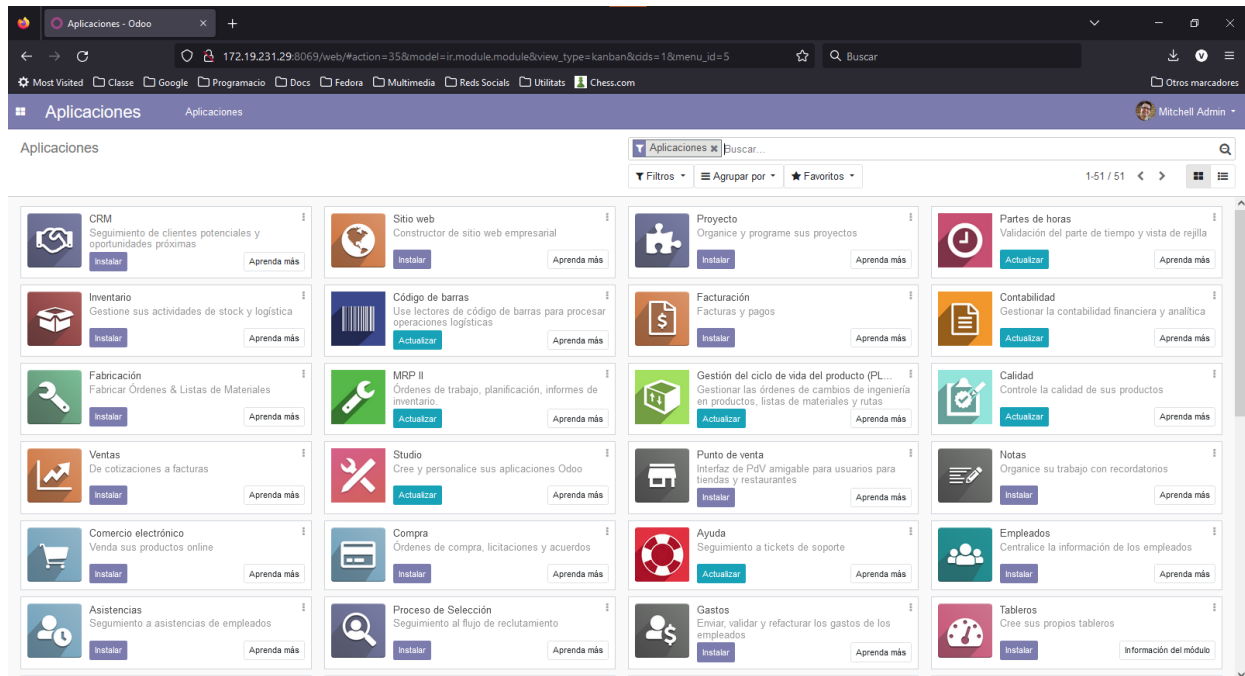


Rellenamos con los datos (contraseña odoo).



Database Name	<input type="text" value="odoo"/>
Email	<input type="text" value="admin@odoo.com"/>
Password	<input type="password" value="••••"/> 
Phone number	<input type="text" value="622674465"/>
Language	<input type="text" value="Spanish / Español"/> 
Country	<input type="text" value="Spain"/> 
Demo data	<input checked="" type="checkbox"/>
<input type="button" value="Create database"/> or restore a database	

Finalmente estamos en Odoo.



8) Conclusión

De esta forma podemos utilizar el ERP de forma virtualizada, pero sin tener que crear una máquina que utiliza muchos recursos en VirtualBox.

Además podemos utilizar el Pycharm en nuestro PC y se conecta de forma remota.

En mi opinión es una forma más cómoda de trabajar con un entorno virtualizado.