

SISTEMA BALANCEADO ODOO + POSTGRESQL

MANUAL DE USUARIO

PABLO HERRERO SÁNCHEZ

SGE – 2º DAM

UD2 – ACTIVIDAD 2

Contenido

1. Preparación del entorno	1
1.1. Creación Servidor 1 & 2.....	1
2. Instalación Odoo sobre Servidor 1	6
2.1. Dependencias y paquetes	6
2.2. Instalación y configuración Odoo.....	9
2.3. Configurar Odoo como servicio.....	14
3. Instalación PostgreSQL sobre Servidor 2	15
4. Comunicando los servidores	18
4.1. Diciéndole a Odoo dónde está PostgreSQL.....	18
4.2. Prueba de funcionamiento.....	19
5. Conclusión.....	20

A lo largo de este documento explicaremos cómo crear un sistema balanceado con **Odoo** y una base de datos **PostgreSQL**, con el que conseguiremos un mayor rendimiento del sistema.

Para facilitar el seguimiento de este manual, además de las capturas de pantalla, incluyo todas las sentencias usadas, para facilitar al usuario el posible copia-pegar de las mismas.

1. Preparación del entorno

Para el propósito de este manual, prepararemos **dos máquinas virtuales con VirtualBox**, cada una de ellas hará la función de servidor, siendo **Servidor 1** donde se alojará Odoo y **Servidor 2** donde correrá la base de datos PostgreSQL.

1.1. Creación Servidor 1 & 2

Como ambos servidores corren **Ubuntu Server 20.04.01 LTS**, la instalación del SO en ambas máquinas es exactamente igual, cambiando tan sólo el nombre que le doy al equipo. A continuación se muestra la creación de la máquina para Servidor 1, siguiendo la de Servidor 2 el mismo proceso.

1. Preparación de las **máquinas virtuales**:

?

×

←

Crear máquina virtual

Nombre y sistema operativo

Seleccione un nombre descriptivo y una carpeta destino para la nueva máquina virtual y seleccione el tipo de sistema operativo que tiene intención de instalar en ella. El nombre que seleccione será usado por VirtualBox para identificar esta máquina.

Nombre:

Servidor1

Carpeta de máquina:

C:\Users\pablo\VirtualBox VMs

Tipo:

Linux

64

Versión:

Ubuntu (64-bit)

Modo experto

Next

Cancelar

← Crear máquina virtual

Tamaño de memoria

Seleccione la cantidad de memoria (RAM) en megabytes a ser reservada para la máquina virtual.

El tamaño de memoria recomendado es **1024 MB**.



Next

Cancelar

← Crear de disco duro virtual

Ubicación del archivo y tamaño

Escriba el nombre del archivo de unidad de disco duro virtual en el campo debajo o haga clic en el icono de carpeta para seleccionar una carpeta diferente donde crear el archivo.

C:\Users\pablo\VirtualBox VMs\Servidor1\Servidor1.vdi



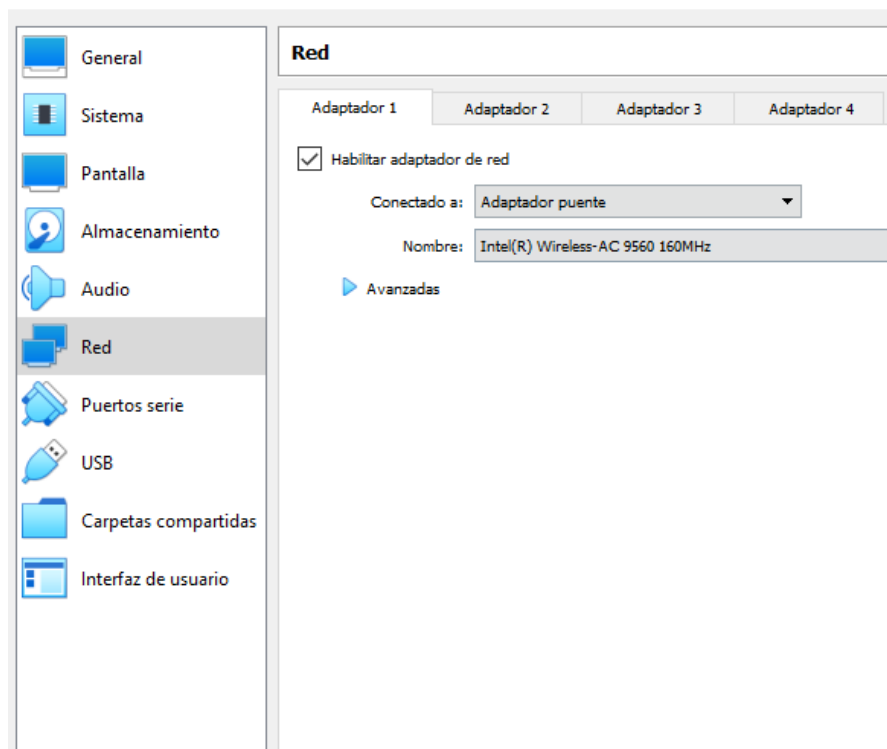
Seleccione el tamaño de disco duro virtual en megabytes. Este tamaño es el límite para el archivo de datos que una máquina virtual podrá almacenar en el disco duro.



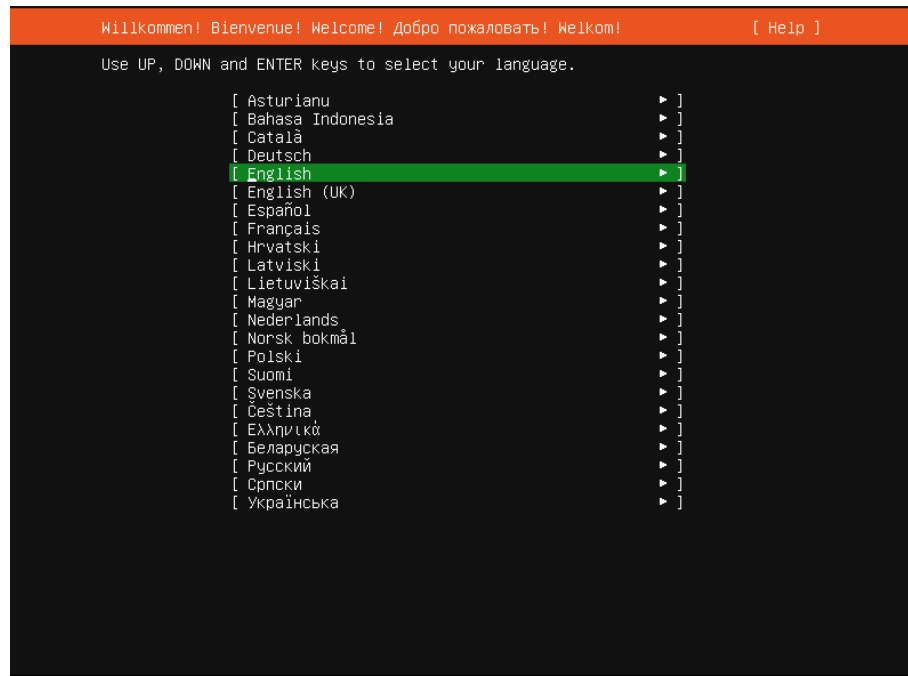
Crear

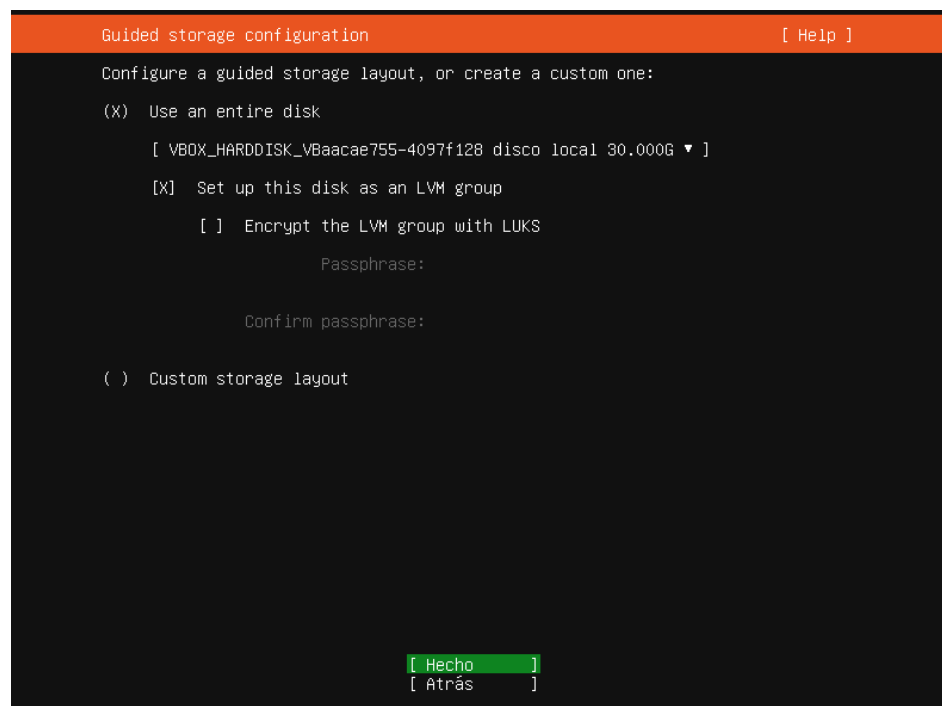
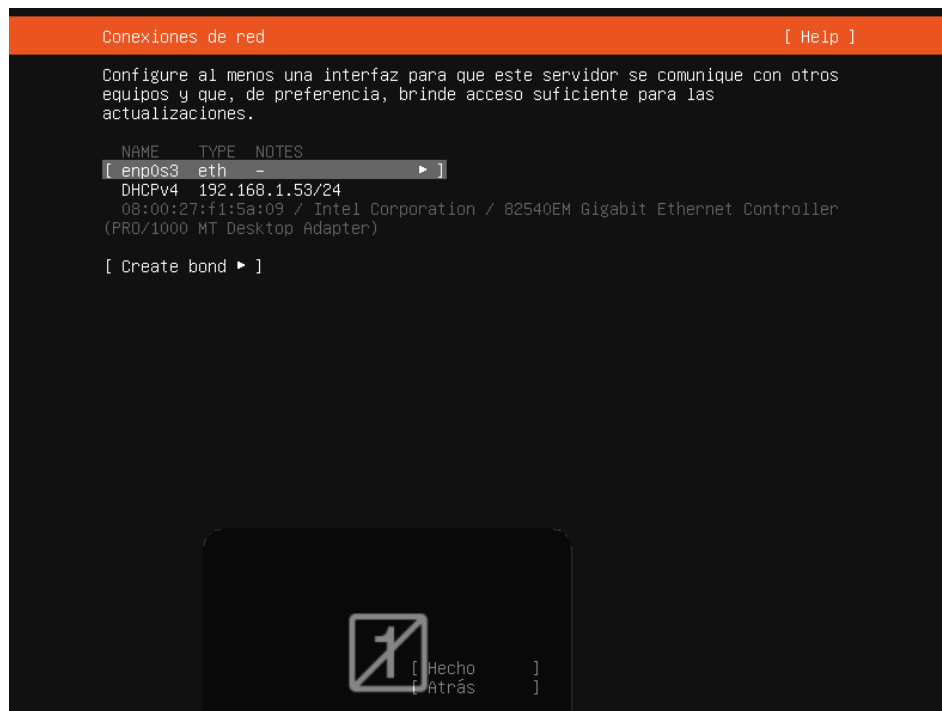
Cancelar

🔧 Servidor1 - Configuración



2. Instalación de **Ubuntu Server 20.04.01 LTS** sobre la máquina:





Configuración de perfil[Help]

Proporcione el nombre de usuario y la contraseña que utilizará para acceder al sistema. Puede configurar el acceso SSH en la pantalla siguiente, pero aun se necesita una contraseña para sudo.

Su nombre:

El nombre del servidor:
El nombre que utiliza al comunicarse con otros equipos.

Elija un nombre de usuario:

Elija una contraseña:

Confirme la contraseña:

1

[Hecho]

Configuración de SSH[Help]

You can choose to install the OpenSSH server package to enable secure remote access to your server.

☒ Instalar servidor OpenSSH

Importar identidad SSH:
Puede importar sus claves SSH desde GitHub o Launchpad.

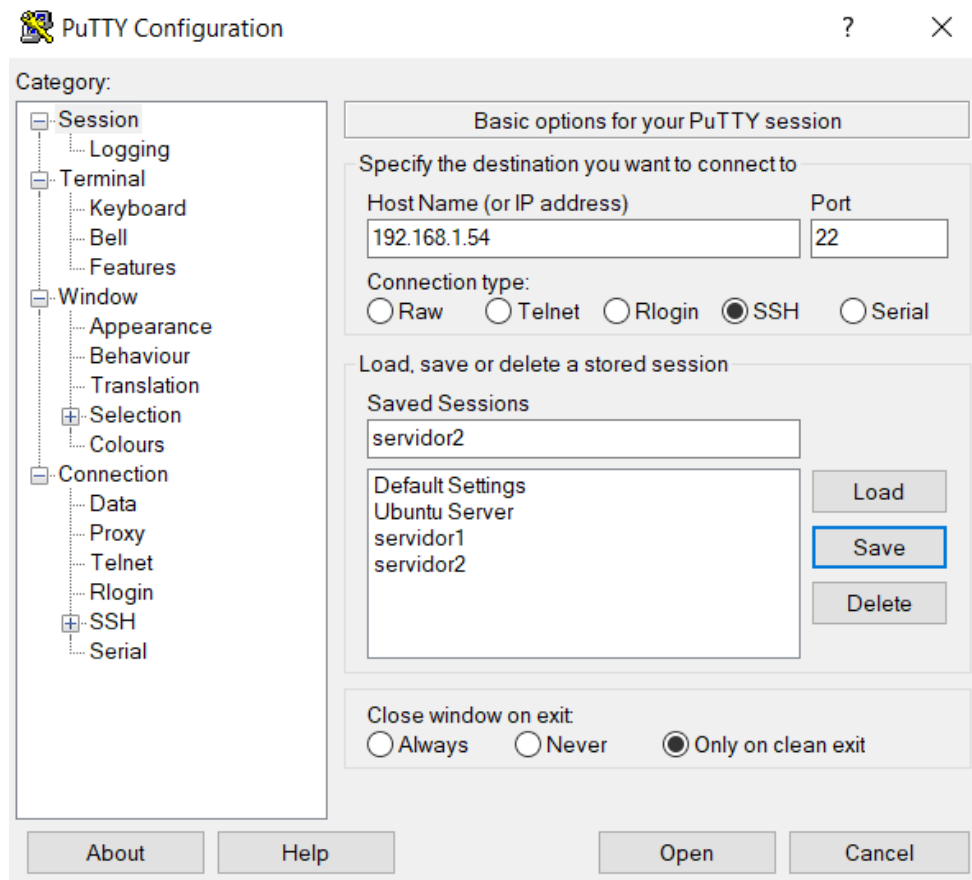
Importar nombre de usuario:

☒ Permitir autenticación con contraseña por SSH

1

[Hecho]
[Atrás]

3. Conecto ambas máquinas a **PuTTY** para facilitar el resto de instalaciones:



2. Instalación Odoo sobre Servidor 1

Para la configuración de **Odoo** en el Servidor 1 realizaremos una instalación manual del mismo:

2.1. Dependencias y paquetes

1. Instalación de **dependencias** necesarias para Odoo, usando el comando ***sudo apt-get***. Si al descargar alguno de los paquetes nos devolviera algún error por falta de dependencias, se puede corregir con el comando ***sudo apt-get --fix-broken install***:

- `sudo apt-get install git python3 python3-pip libxml2-dev libxslt-dev libevent-dev libsass2-dev libldap2-dev`

```
pherrero@servidor1:~$ sudo apt-get install git python3 python3-pip libxml2-dev 1
libxslt-dev libevent-dev libsass2-
```

- `sudo apt install xfonts-base xfonts-75dpi -y`

```
pherrero@servidor1:~$ sudo apt install xfonts-base xfonts-75dpi -y
```

- `sudo apt-get install -y software-properties-common`

```
pherrero@servidor1:~$ sudo apt-get install -y software-properties-common
```


- `sudo apt-add-repository -y "deb http://security.ubuntu.com/ubuntu bionic-security main"`

```
pherrero@servidor1:~$ sudo apt-add-repository -y "deb http://security.ubuntu.com
/ubuntu bionic-security main"
```

- `sudo apt-get -yq update`

```
pherrero@servidor1:~$ sudo apt-get -yq update
```

- `sudo apt-get install -y libxrender1 libfontconfig1 libx11-dev libjpeg62 libxtst6 fontconfig xfonts-75dpi xfonts-base libjpeg-turbo8`

```
pherrero@servidor1:~$ sudo apt-get install -y libxrender1 libfontconfig1 libx11-
dev libjpeg62 libxtst6 fontconfig
```

2. Instalación de **wkhtmltopdf**:

- `sudo wget https://github.com/wkhtmltopdf/packaging/releases/download/0.12.6-1/wkhtmltox_0.12.6-1.focal_amd64.deb` [OJO, al copiar la sentencia completa en PuTTY da error "URL NOT FOUND". Aconsejo escribir la sentencia completa en el terminal para que no dé dicho error]

```
pherrero@servidor1:~$ sudo wget https://github.com/wkhtmltopdf/packaging/release
s/download/0.12.6-1/wkhtmltox_0.12.6-1.focal_amd64.deb
```

- `sudo dpkg -i wkhtmltox_0.12.6-1.focal_amd64.deb`

```
pherrero@servidor1:~$ sudo dpkg -i wkhtmltox_0.12.6-1.focal_amd64.deb
```

3. Instalación de **npm**:

- `sudo apt-get -yq install npm`

```
pherrero@servidor1:~$ sudo apt-get -yq install npm
```

Al instalar este paquete es posible que nos dé un error, solucionable con los comandos `sudo apt-get -f install` + `sudo apt-get update` + `sudo apt-get clean`.

4. Instalación de **Node.js**:

- `sudo ln -s /usr/bin/nodejs /usr/bin/node`

```
pherrero@servidor1:~$ sudo ln -s /usr/bin/nodejs /usr/bin/node
```

5. Instalación de **less-plugin-clean-css**:

- `sudo npm install -g less less-plugin-clean-css`

```
pherrero@servidor1:~$ sudo npm install -g less less-plugin-clean-css
```

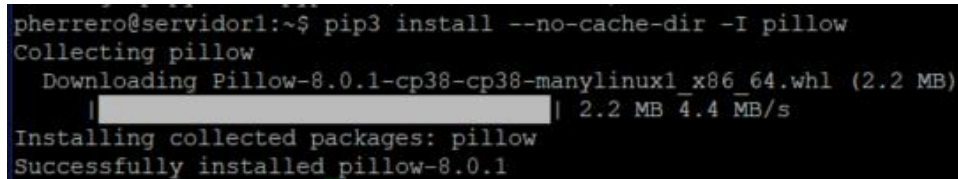
6. Instalación de dependencias de **Python y pip3**:

Aunque se pueden copiar y pegar la sentencia que indico a continuación, esto puede dar errores en la instalación, por lo que aconsejo, pese al tiempo que pueda llevar, instalar cada paquete individualmente.

- `sudo apt-get install python3 python3-pip fontconfig libxml2 libxml2-dev libxslt1.1 libxslt1-dev zlib1g liblcms2-2 libtiff5 tk tcl libpq5 libldap-2.4-2 libsasl2-2 libx11-6 libxext6 libxrender1 git libsasl2-dev libldap2-dev python3-passlib python3-babel python3-werkzeug python3-lxml python3-decorator python3-dateutil python3-psycopg2 python3-pil python3-psutil python3-jinja2 python3-reportlab python3-html2text python3-docutils python3-suds python3-pypdf2`

7. Instalación de **Pillow**:

- `pip3 install --no-cache-dir -I pillow`



```
pherrero@servidor1:~$ pip3 install --no-cache-dir -I pillow
Collecting pillow
  Downloading Pillow-8.0.1-cp38-cp38-manylinux1_x86_64.whl (2.2 MB)
    |#####| 2.2 MB 4.4 MB/s
Installing collected packages: pillow
Successfully installed pillow-8.0.1
```

8. Llegado a este punto, es muy recomendable realizar una **copia de seguridad** del Servidor 1 desde VirtualBox para tener este punto de restauración en el futuro:



Exportar servicio virtualizado

Preferencias de servicio virtualizado

Seleccione un formato al que exportar el servicio virtualizado.

El **Open Virtualization Format** soporta solo las extensiones **ovf** o **ova**. Si usa la extensión **ovf**, varios archivos serán escritos por separado. Si usa la extensión **ova**, todos los archivos serán combinados en un archivo Open Virtualization Format.

El formato **Oracle Cloud Infrastructure** solo soporta exportar a servidores remotos cloud. El disco virtual principal de cada máquina seleccionada será subido al servidor remoto.

Formato: Open Virtualization Format 1.0

Seleccione un nombre de archivo al que exportar el servicio. Aparte de eso puede especificar una cierta cantidad de opciones que afectan al tamaño y contenido del archivo resultante.

Archivo: C:\Users\pablo\Documents\Servidor1_ConPaquetes\ova

Política de direcciones MAC: Incluir solo las direcciones MAC de adaptador de red NAT

Adicionalmente: ☒ Escribir archivo de manifiesto

☐ Incluir archivos de imagen ISO

2.2. Instalación y configuración Odoo

1. Crear un **usuario del sistema odoo** con los valores por defecto vacíos:

- `sudo adduser odoo`

```
pherrero@servidor1:~$ sudo adduser odoo
[sudo] password for pherrero:
Adding user `odoo' ...
Adding new group `odoo' (1001) ...
Adding new user `odoo' (1001) with group `odoo' ...
Creating home directory `/home/odoo' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for odoo
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
```

2. Descargar el **código fuente de Odoo** de su repositorio de Git, almacenándolo en el usuario creado:

- `sudo apt-get install git`
- `su - odoo git clone https://github.com/Odoo/odoo.git --depth 1 --branch 13.0 --single-branch odoo`

```
pherrero@servidor1:~$ sudo apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.25.1-1ubuntu3).
0 upgraded, 0 newly installed, 0 to remove and 69 not upgraded.
pherrero@servidor1:~$ su - odoo
Password:
odoo@servidor1:~$ git clone https://github.com/Odoo/odoo.git --depth 1 --branch 13.0 --single-branch odoo
Cloning into 'odoo'...
remote: Enumerating objects: 28582, done.
remote: Counting objects: 100% (28582/28582), done.
remote: Compressing objects: 100% (23575/23575), done.
remote: Total 28582 (delta 8264), reused 12137 (delta 4117), pack-reused 0
Receiving objects: 100% (28582/28582), 122.16 MiB | 9.19 MiB/s, done.
Resolving deltas: 100% (8264/8264), done.
Updating files: 100% (25622/25622), done.
odoo@servidor1:~$
```

3. Volver al directorio de Odoo y salir del usuario Odoo:

```
odoo@servidor1:~/odoo/addons$ cd ..
odoo@servidor1:~/odoo$ cd ..
odoo@servidor1:~$ pwd
/home/odoo
odoo@servidor1:~$ exit
logout
pherrero@servidor1:~$
```

4. Instalar los requerimientos de Odoo en el archivo **requirements.txt**. Se almacenará en /home/odoo/odoo/:

- `sudo pip3 install -r /home/odoo/odoo/requirements.txt`

```
pherrero@servidor1:~$ sudo pip3 install -r /home/odoo/odoo/requirements.txt
pherrero@servidor1:/home/odoo/odoo$ ls
CONTRIBUTING.md  MANIFEST.in  addons  odoo  setup
COPYRIGHT        README.md    debian  odoo-bin  setup.cfg
LICENSE          SECURITY.md  doc     requirements.txt  setup.py
```

5. Incluir al usuario **odoo** en **sudoers**.

El usuario odoo necesita acceder a requirements.txt. Se le debe incluir en el archivo *sudoers* mediante el comando *sudo visudo* desde el usuario sudo. En este archivo es donde se configuran en un entorno Linux los permisos de los usuarios.

- `sudo visudo`

```
pherrero@servidor1:/home/odoo/odoo$ sudo visudo
```

- Añadir `odoo ALL=(ALL) NOPASSWD:ALL`

```
GNU nano 4.8 /etc/sudoers.tmp Modified
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults      env_reset
Defaults      mail_badpass
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/usr/local/games:/usr/games"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#include_dir /etc/sudoers.d

# Users added to sudoers
odoo ALL=(ALL) NOPASSWD:ALL

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^_ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

Tras esta operación, logarse con el usuario odoo y comprobar el acceso a requirements.txt:

- *Nano requirements.txt*

```
pherrero@servidor1:/home/odoo/odoo$ su odoo
Password:
odoo@servidor1:~/odoo$ nano requirements.txt
```

6. Crear **odoo-server.conf**, archivo de configuración de Odoo, en el directorio del usuario *odoo*. Para ello, situarse en el directorio del usuario y crear el archivo:

- *Nano odoo-server.conf*

```
/home/odoo/odoo
odoo@servidor1:~/odoo$ cd ..
odoo@servidor1:~$ pwd
/home/odoo
odoo@servidor1:~$ cd
odoo@servidor1:~$ nano odoo-server.conf
```

- Añadir este contenido al archivo, modificando la contraseña de la BBDD al gusto:

```
GNU nano 4.8 odoo-server.conf Modified
[options]
#Password de administrador de Odoo
admin_passwd = admin
#Puerto en el que funciona Odoo por defecto
xmlrpc = True
xmlrpc_port = 8069
#Host local, a cambiar si la BBDD se instala en otro equipo
db_host = 127.0.0.1
#Puerto de la BBDD por defecto
db_port = 5432
#Usuario de la BBDD
db_user = odoo
#Contraseña de la BBDD
db_password = xxxxx
addons_path = /home/odoo/odoo/addons
```

7. Al intentar **iniciar Odoo** es muy posible que nos devuelva este **error**:

```
odoo@servidor1:~$ /home/odoo/odoo/odoo-bin -c /home/odoo/odoo-server.conf
Traceback (most recent call last):
  File "/home/odoo/odoo/odoo-bin", line 5, in <module>
    import odoo
  File "/home/odoo/odoo/odoo/__init__.py", line 113, in <module>
    from . import modules
  File "/home/odoo/odoo/odoo/modules/__init__.py", line 8, in <module>
    from . import db, graph, loading, migration, module, registry
  File "/home/odoo/odoo/odoo/modules/graph.py", line 10, in <module>
    import odoo.tools as tools
  File "/home/odoo/odoo/odoo/tools/__init__.py", line 11, in <module>
    from .translate import *
  File "/home/odoo/odoo/odoo/tools/translate.py", line 11, in <module>
    import polib
ModuleNotFoundError: No module named 'polib'
```

- Para solucionarlo, modificamos el archivo *requirements.txt* con los datos que nos devuelven los comandos *pip3 list* para *Pillow* y *psycpg2*

```
keyring 18.0.1
language-selector 0.1
launchpadlib 1.10.13
lazr.restfulclient 0.14.2
lazr.uri 1.0.3
lxml 4.5.0
MarkupSafe 1.1.0
more-itertools 4.2.0
netifaces 0.10.4
oauthlib 3.1.0
olefile 0.46
passlib 1.7.2
Pillow 8.0.1
pip 20.0.2
psutil 5.5.1
psycpg2 2.8.4
pyasn1 0.4.2
pyasn1-modules 0.2.1
Pygments 2.3.1
PyGObject 3.36.0
PyHamcrest 1.9.0
pyinotify 0.9.6
PyJWT 1.7.1
pymacaroons 0.13.0
GNU nano 4.8 requirements.txt
MarkupSafe==1.1.0
mock==2.0.0
num2words==0.5.6
ofxparse==0.19
passlib==1.7.1
#Pillow==5.4.1 ; python_version < '3.7' or sys_platform != 'win32'
#Pillow==6.1.0 ; sys_platform == 'win32' and python_version >= '3.7'
Pillow==8.0.1
polib==1.1.0
psutil==5.6.6
#psycpg2==2.7.7; sys_platform != 'win32' and python_version < '3.8'
#psycpg2==2.8.3; sys_platform == 'win32' or python_version >= '3.8'
psycpg2==2.8.4
pydot==1.4.1
python-ldap==3.1.0; sys_platform != 'win32'
pyparsing==2.2.0
PyPDF2==1.26.0
pyserial==3.4
python-dateutil==2.7.3
pytz==2019.1
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

- Volvemos a instalar los requerimientos con
sudo pip3 install -r /home/odoo/odoo/requirements.txt

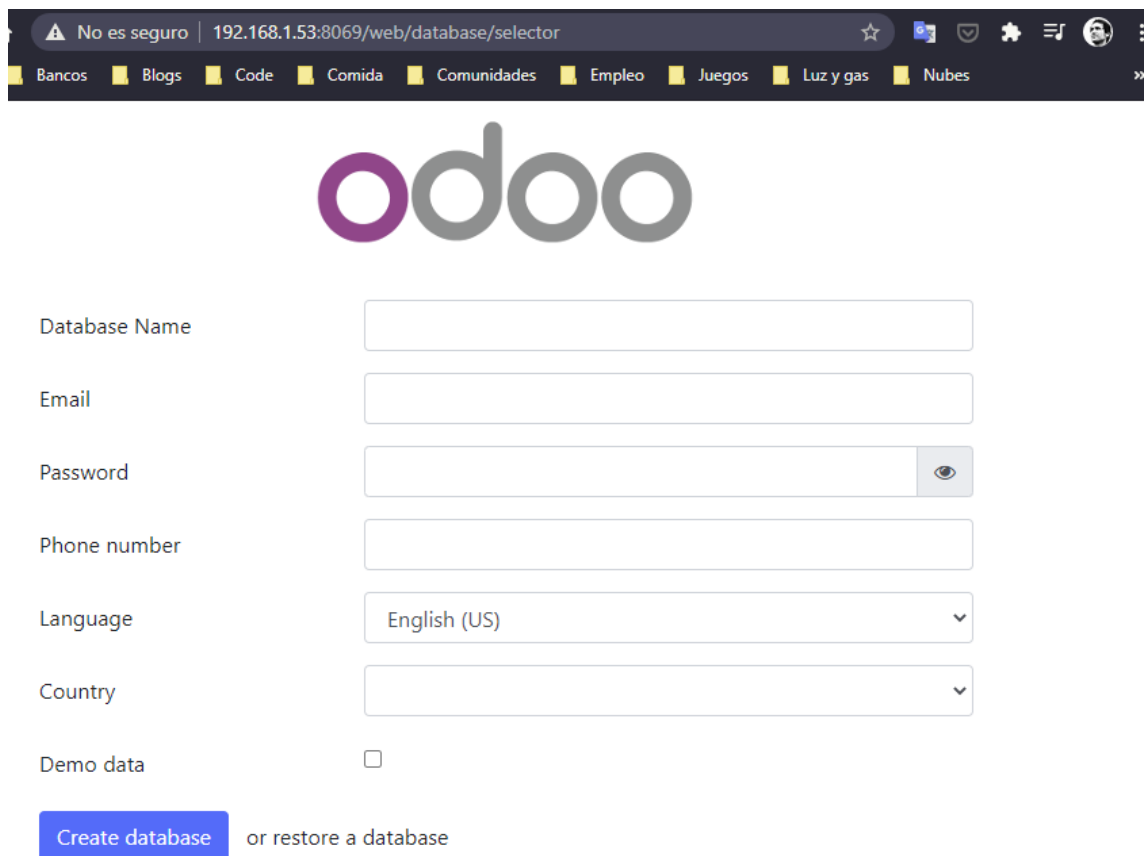
```
odoo@servidor1:~/odoo$ nano requirements.txt
odoo@servidor1:~/odoo$ cd ..
odoo@servidor1:~$ sudo pip3 install -r /home/odoo/odoo/requirements.txt
```

8. **Lanzamos Odoo manualmente**, estando logados como usuario odoo, y comprobamos que corre perfectamente:

- `/home/odoo/odoo/odoo-bin -c /home/odoo/odoo-server.conf`

```
odoo@servidor1:~$ /home/odoo/odoo/odoo-bin -c /home/odoo/odoo-server.conf
2020-11-17 09:01:06,324 1538 INFO ? odoo: Odoo version 13.0
2020-11-17 09:01:06,324 1538 INFO ? odoo: Using configuration file at /home/odoo/odoo-server.conf
2020-11-17 09:01:06,324 1538 INFO ? odoo: addons paths: ['/home/odoo/odoo/odoo/addons', '/home/odoo/.local/share/Odoo/addons/13.0', '/home/odoo/odoo/addons']
2020-11-17 09:01:06,325 1538 INFO ? odoo: database: odoo@127.0.0.1:5432
2020-11-17 09:01:06,562 1538 INFO ? odoo.addons.base.models.ir_actions_report: Will use the Wkhtmltopdf binary at /usr/local/bin/wkhtmltopdf
2020-11-17 09:01:06,800 1538 INFO ? odoo.service.server: HTTP service (werkzeug) running on servidor1:8069
```

9. Para **abrir Odoo en el navegador** del pc madre, en vez de abrir Odoo con la dirección habitual, <http://ip:8069/>, indicamos la ip del Servidor 1, en mi caso <http://192.168.1.53:8069/>




No es seguro | 192.168.1.53:8069/web/database/selector

Bancos | Blogs | Code | Comida | Comunidades | Empleo | Juegos | Luz y gas | Nubes

odoo

Database Name

Email

Password 

Phone number

Language

Country

Demo data ☐

Create database or restore a database

2.3. Configurar Odoo como servicio

A continuación detallamos cómo configurar Odoo como servicio, conseguimos que se ejecute automáticamente cuando se inicia el servidor sin necesidad siquiera de tener que logarnos en la terminal y permitiéndonos, si así lo quisiéramos, seguir utilizando la consola.

1. Creamos y editamos el fichero **odoo13.service** en el directorio **/etc/systemd/system/**:

- `Sudo nano /etc/systemd/system/odoo13.service`

```
pherrero@servidor1:~$ sudo nano /etc/systemd/system/odoo13.service
```

- **Añadimos** los siguientes parámetros:

[options]

`admin_passwd = admin`

`xmlrpc = True`

`xmlrpc_port = 8069`

`db_host = 127.0.0.1`

`db_port = 5432`

`db_user = odoo` [usuario de la base de datos que crearemos en el siguiente apartado]

`db_password = xxxxx` [contraseña del usuario de la base de datos]

`addons_path = /home/odoo/odoo/addons`

```
GNU nano 4.8 odoo13.service
[Unit]
Description=Odoo
[Service]
Type=simple
User=odoo
ExecStart=/home/odoo/odoo/odoo-bin -c /home/odoo/odoo-server.conf
[Install]
WantedBy=default.target

[ Read 8 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line
```

2. Le indicamos al SO que queremos que el servicio de Odoo que acabamos de crear se **arranque al inicio del sistema**, habilitando dicho servicio:

- `sudo systemctl enable odoo13.service`

```
pherrero@servidor1:~$ sudo nano /etc/systemd/system/odoo13.service
[sudo] password for pherrero:
pherrero@servidor1:~$ sudo systemctl start odoo13.service
pherrero@servidor1:~$ sudo systemctl stop odoo13.service
pherrero@servidor1:~$ sudo systemctl enable odoo13.service
Created symlink /etc/systemd/system/default.target.wants/odoo13.service → /etc/systemd/system/odoo13.service.
```


3. Reiniciamos el sistema y comprobamos que Odoo esté activo

- `sudo reboot`
- `ps aux | grep odoo`

```
odoo@servidor1:/home/pherrero$ ps aux | grep odoo
odoo      596  0.0  3.9 237748 80960 ?        Ssl   07:16   0:03 python3 /home
/odoo/odoo/odoo-bin -c /home/odoo/odoo-server.conf
root      3442  0.0  0.2   8208  4556 pts/0    S     09:12   0:00 sudo su odoo
root      3444  0.0  0.1   7128  3848 pts/0    S     09:12   0:00 su odoo
odoo      3445  0.0  0.2   7028  5008 pts/0    S     09:12   0:00 bash
odoo      3463  0.0  0.1   7632  3340 pts/0    R+    09:13   0:00 ps aux
odoo      3464  0.0  0.1   5192  2476 pts/0    S+    09:13   0:00 grep --color=
auto odoo
```

4. Es importante recalcar los **comandos** con los que podemos **manejar el servicio de Odoo**:

- Iniciar: `sudo systemctl start odoo13.service`
- Estado: `sudo systemctl status odoo13.service`
- Parar: `sudo systemctl stop odoo13.service`
- Reiniciar: `sudo systemctl restart odoo13.service`

5. Llegado este punto, recomiendo hacer una **copia de seguridad** de la máquina virtual.

3. Instalación PostgreSQL sobre Servidor 2

En este apartado procederemos a realizar la instalación de la base de datos PostgreSQL junto con su cliente para terminal en el Servidor 2.

1. Instalamos la base de datos PostgreSQL:

- `sudo apt-get install postgresql`

```
pherrero@servidor2:~$ sudo apt-get install postgresql
```

2. Asignamos una **contraseña al usuario postgres**:

- `sudo passwd postgres`

```
pherrero@servidor2:~$ sudo passwd postgres
New password:
Retype new password:
passwd: password updated successfully
pherrero@servidor2:~$
```

3. Nos **logamos** como postgres y **comprobamos el funcionamiento** de la base de datos:

- *su postgres*

```
pherrero@servidor2:~$ su postgres
Password:
postgres@servidor2:/home/pherrero$ psql
psql (12.4 (Ubuntu 12.4-0ubuntu0.20.04.1))
Type "help" for help.

postgres=# \l

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

4. Logados como el usuario postgres, **creamos el usuario odoo** para la base de datos:

- *createuser -P -s -e odoo*

```
pherrero@servidor2:~$ sudo su postgres
postgres@servidor2:/home/pherrero$ createuser -P -s -e odoo
Enter password for new role:
Enter it again:
SELECT pg_catalog.set_config('search_path', '', false);
CREATE ROLE odoo PASSWORD 'md5c557e035efe7c54eb9a364ba489be985' SUPERUSER CREATE
DB CREATEROLE INHERIT LOGIN;
```

5. **Configuramos** el archivo **postgresql.conf**. Contiene los parámetros de configuración que afectan a PostgreSQL:

- Acceder a */etc/postgresql/12/main*
- Editar con *sudo nano postgresql.conf*
- Descomentar y modificar la línea *listen_addresses*, quedando de esta forma:
listen_addresses = ''*

```
GNU nano 4.8 postgresql.conf Modified
# (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
                        # (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 direc
# (change requires restart)
#unix_socket_group = '' # (change requires restart)
#unix_socket_permissions = 0777 # begin with 0 to use octal notation

[Cancelled]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

De esta forma la base de datos escuche todas las IP. Tras realizar el cambio, reiniciamos con *sudo reboot*.

6. **Configuramos** el archivo **pg_hba.conf**:

- Nos situamos en la ruta */etc/postgresql/12/main*
- Editamos el archivo con *sudo nano pg_hba.conf*
- Añadimos la línea *host all all 0.0.0.0/0 md5* en el apartado *IPv4*. Tras realizar el cambio, reiniciamos con *sudo reboot*:

```
GNU nano 4.8                                pg_hba.conf                                Modified
# 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
# 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

[ Cancelled ]
^G Get Help  ^O Write Out ^W Where Is   ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line
```

7. **Configuramos** la **contraseña** de la base de datos.

- Nos logamos como usuario *postgres* y ejecutando el cliente *psql* aplicamos la sentencia *ALTER USER postgres PASSWORD 'Password_para_base_de_datos';* donde *password* lo sustituimos por la contraseña que queramos.

8. En este punto, aconsejo hacer una **copia de seguridad** de la máquina del Servidor 2.

4. Comunicando los servidores

En este último apartado, comunicaremos ambos servidores y crearemos una base de datos de prueba desde Odoo para comprobar el correcto funcionamiento del sistema

4.1. Diciéndole a Odoo dónde está PostgreSQL

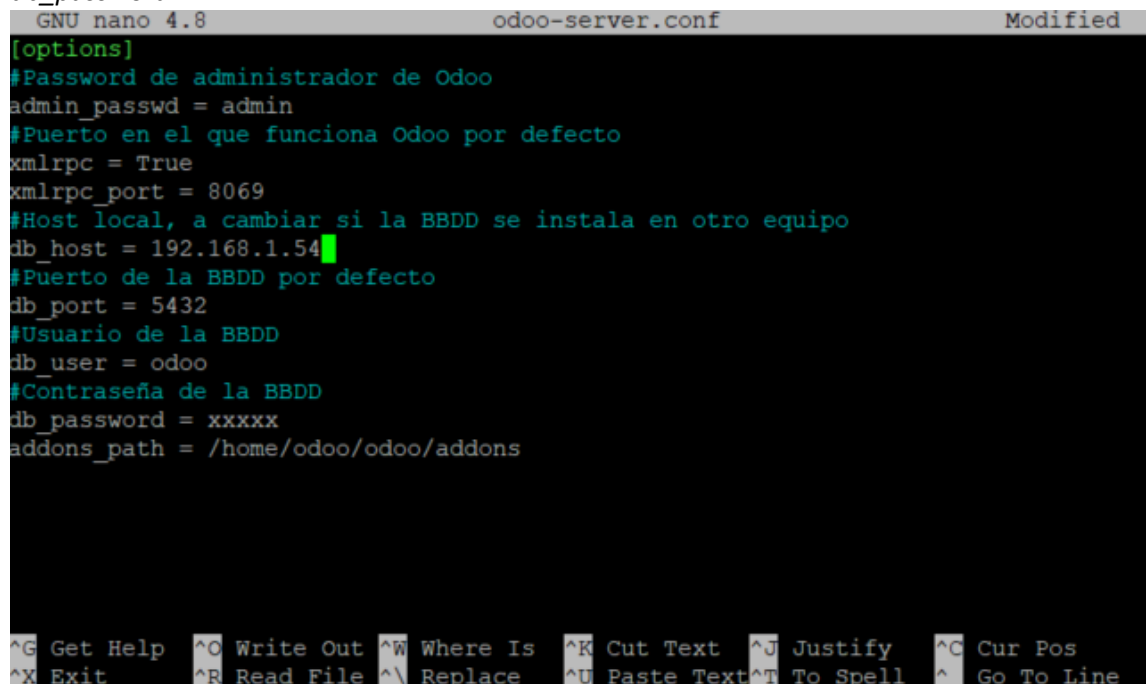
1. **Detenemos el servicio Odoo en el Servidor 1**, logándonos como usuario odoo:

- `sudo systemctl stop odoo13.service`

2. **Averiguamos la IP del Servidor 2** con la sentencia `ifconfig`.

3. **Modificamos el archivo `odoo-server.conf` en el Servidor 1:**

- Logándonos como usuario odoo, nos situamos en el directorio `/home/odoo`.
- Editamos con `sudo nano odoo-server.conf`, indicando la ip del Servidor 2 del punto anterior.
- Verificamos el usuario de la base de datos en `db_user` y ponemos su contraseña en `db_password`.



```
GNU nano 4.8 odoo-server.conf Modified
[options]
#Password de administrador de Odoo
admin_passwd = admin
#Puerto en el que funciona Odoo por defecto
xmlrpc = True
xmlrpc_port = 8069
#Host local, a cambiar si la BBDD se instala en otro equipo
db_host = 192.168.1.54
#Puerto de la BBDD por defecto
db_port = 5432
#Usuario de la BBDD
db_user = odoo
#Contraseña de la BBDD
db_password = xxxxxx
addons_path = /home/odoo/odoo/addons


^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```


4. **Iniciamos el servicio Odoo en el Servidor 1:**

- `sudo systemctl start odoo13.service`

4.2. Prueba de funcionamiento

1. Iniciamos las dos máquinas virtuales (no hay necesidad de logarnos).
2. Accedemos a Odoo desde el navegador e introducimos los datos de la prueba:



Database Name	<input type="text"/>
Email	<input type="text"/>
Password	<input type="password"/> 
Phone number	<input type="text"/>
Language	<input type="text" value="English (US)"/> ▼
Country	<input type="text"/> ▼
Demo data	<input type="checkbox"/>

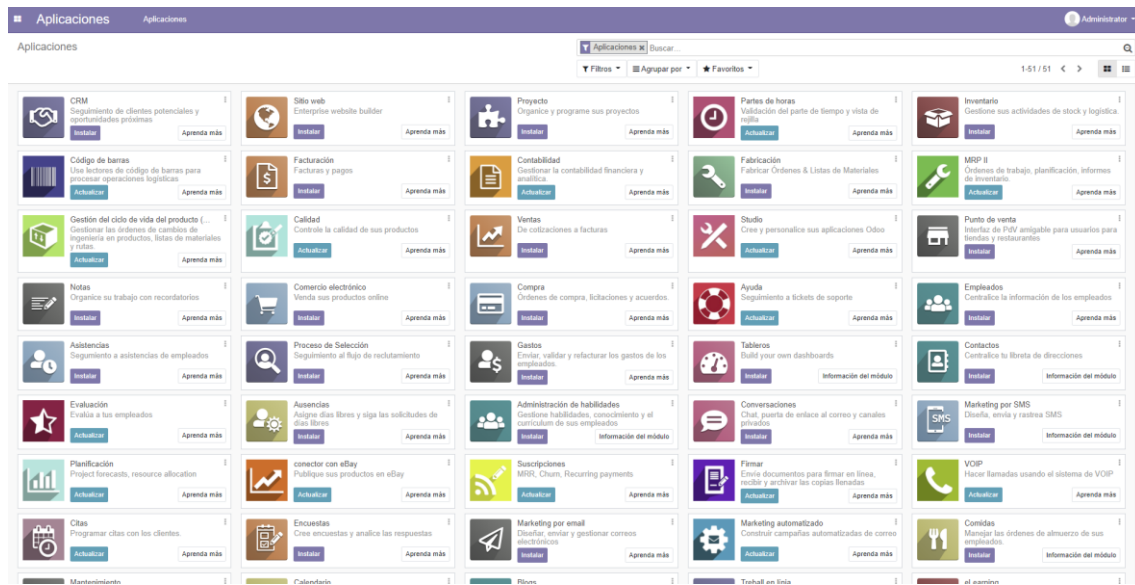
[Create database](#) or restore a database



Database Name	<input type="text" value="pruebaFuncionamientoOdoo"/>
Email	<input type="text" value="pablohs84@educastur.es"/>
Password	<input type="password" value="....."/> 
Phone number	<input type="text" value="666666666"/>
Language	<input type="text" value="Spanish / Español"/> ▼
Country	<input type="text" value="Spain"/> ▼
Demo data	<input type="checkbox"/>

[Create database](#) or restore a database

3. Accedemos al panel de control de Odoo, verificando que el sistema funciona correctamente:



5. Conclusión

Con los puntos indicados, ya tenemos nuestro sistema balanceado de Odoo + PostgreSQL listo para funcionar, dándonos un mayor rendimiento general en todas las operaciones que realicemos con él.