

ODOO: PREPARAMOS EL SISTEMA PARA DESARROLLAR EN ODOO

Comprobamos que está Odoo en marcha:

service odoo status

Comprobamos:

Ip:8069

Creamos una BDD de prueba:



The form for creating a new database. It includes fields for Database Name, Email, Password, Phone number, Language, Country, and Demo data. A blue button at the bottom left says "Create database".

Database Name	pruebas
Email	admin
Password 
Phone number	
Language	Spanish / Español
Country	Spain
Demo data	<input checked="" type="checkbox"/>
Create database or restore a database	

- Instalamos los módulos: CRM, Ventas, Compras, Facturación.

- Con

service odoo status

```
Ubuntu Server 20.04 LTS Limpia [Running]
The list of available updates is more than a week old.
To check for new updates run: sudo apt update

last login: Tue Dec  8 16:44:38 UTC 2020 on ttym1
$ videos y mucho más m:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.0.105  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fe59:f015  prefixlen 64  scopeid 0x20<link>
          ether 08:00:27:59:f0:15  txqueuelen 1000  (Ethernet)
            RX packets 64002  bytes 96691350 (96.6 MB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 22593  bytes 1528572 (1.5 MB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
      inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
          loop  txqueuelen 1000  (Local Loopback)
            RX packets 152  bytes 19773 (19.7 KB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 152  bytes 19773 (19.7 KB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

alumno@alumnodam:~$ service odoo status
● odoo.service - LSB: Start odoo daemon at boot time
  Loaded: loaded (/etc/init.d/odoo; generated)
  Active: active (running) since Sat 2021-01-30 15:41:07 UTC; 24min ago
    Docs: man:systemd-sysv-generator(8)
    Tasks: 4 (limit: 4621)
   Memory: 233.5M
   CGroup: /system.slice/odoo.service
           └─677 python3 /usr/bin/odoo --config /etc/odoo/odoo.conf --logfile /var/log/odoo/odoo->

Jan 30 15:41:07 alumno@alumnodam systemd[1]: Starting LSB: Start odoo daemon at boot time...
Jan 30 15:41:07 alumno@alumnodam odoo[627]: Starting odoo: ok
Jan 30 15:41:07 alumno@alumnodam systemd[1]: Started LSB: Start odoo daemon at boot time.
Lines 1-12/12 (END)
```

Podemos ver lo último que ha ejecutado: con python3, ha ejecutado el programa odoo que está en la ruta /usr/bin/odoo con esa configuración y volcando el fichero log en la ruta indicada.

Vamos a ver el fichero de los: **cat /var/log/odoo/odoo-server.log**

```
Ubuntu Server 20.04 LTS Limpia [Running]
eb/dataset/search_read HTTP/1.1" 200 - 7 0.002 0.030
2021-01-30 16:02:13,600 677 INFO pruebas odoo.addons.base.models.ir_cron: Starting job `Post process payment transactions`.
2021-01-30 16:02:13,613 677 INFO pruebas odoo.addons.base.models.ir_cron: Job `Post process payment transactions` done.
2021-01-30 16:02:13,614 677 INFO pruebas odoo.addons.base.models.ir_cron: Starting job `Mail: Email Queue Manager`.
2021-01-30 16:02:13,621 677 INFO pruebas odoo.addons.base.models.ir_cron: Job `Mail: Email Queue Manager` done.
2021-01-30 16:02:13,621 677 INFO pruebas odoo.addons.base.models.ir_cron: Starting job `Notification :: Delete Notifications older than 6 Month`.
2021-01-30 16:02:13,627 677 INFO pruebas odoo.addons.base.models.ir_cron: Job `Notification: Delete Notifications older than 6 Month` done.
2021-01-30 16:02:13,627 677 INFO pruebas odoo.addons.base.models.ir_cron: Starting job `Users: Notify About Unregistered Users`.
2021-01-30 16:02:13,632 677 INFO pruebas odoo.addons.base.models.ir_cron: Job `Users: Notify About Unregistered Users` done.
2021-01-30 16:02:13,633 677 INFO pruebas odoo.addons.base.models.ir_cron: Starting job `Partner Auto complete : Sync with remote DB`.
2021-01-30 16:02:13,637 677 INFO pruebas odoo.addons.base.models.ir_cron: Job `Partner Autocomplete : Sync with remote DB` done.
2021-01-30 16:02:13,637 677 INFO pruebas odoo.addons.base.models.ir_cron: Starting job `Snailmail: process letters queue`.
2021-01-30 16:02:13,642 677 INFO pruebas odoo.addons.base.models.ir_cron: Job `Snailmail: process letters queue` done.
2021-01-30 16:02:13,642 677 INFO pruebas odoo.addons.base.models.ir_cron: Starting job `SMS: SMS Queue Manager`.
2021-01-30 16:02:13,647 677 INFO pruebas odoo.addons.base.models.ir_cron: Job `SMS: SMS Queue Manager` done.
2021-01-30 16:02:13,647 677 INFO pruebas odoo.addons.base.models.ir_cron: Starting job `Digest Emails`.
2021-01-30 16:02:13,652 677 INFO pruebas odoo.addons.base.models.ir_cron: Job `Digest Emails` done.
2021-01-30 16:02:13,652 677 INFO pruebas odoo.addons.base.models.ir_cron: Starting job `Mail: Notify channel moderators`.
2021-01-30 16:02:13,656 677 INFO pruebas odoo.addons.base.models.ir_cron: Job `Mail: Notify channel moderators` done.
alumno@alumnodam:~$
```

Con **tail -f /var/log/odoo/odoo-server.log** podemos ver lo último que está pasando.

Si queremos buscar por ejemplo Warnings, podemos hacer: **cat /var/log/odoo/odoo-server.log | grep 'WARN'**

En el fichero de configuración podemos ver entre otras cosas: usuario, usuario de bdd y contraseña (que por defecto no tenemos contraseña de BDD).

```
[options]
addons_path = /opt/odoo/odoo/addons
admin_passwd = admin
csv_internal_sep = ,
data_dir = /opt/odoo/.local/share/Odoo
db_host = False
db_maxconn = 64
db_name = False
db_password = False
db_port = False
db_sslmode = prefer
db_template = template0
db_user = False
dbfilter =
demo = {}
email_from = False
geoip_database = /usr/share/GeoIP/GeoLite2-City.mmdb
http_enable = True
http_interface =
http_port = 8069
import_partial =
limit_memory_hard = 2684354560
limit_memory_soft = 2147483648
limit_request = 8192
limit_time_cpu = 60
limit_time_real = 120
limit_time_real_cron = -1
list_db = True
log_db = False
log_db_level = warning
log_handler = :INFO
log_level = info
logfile = /var/log/odoo/odoo-server.log
longpolling_port = 8072
max_cron_threads = 2
osv_memory_age_limit = 1.0
--More--(65%)
```

Tenemos un servicio web y lo que vemos en el navegador es un cliente web que es una aplicación en JS muy completa. Si abrimos las herramientas de desarrollo en el navegador, en la pestaña de red y refrescamos:

The screenshot shows the Odoo application store interface with several modules listed:

- CRM: Seguimiento de clientes potenciales y oportunidades próximas. Status: Instalado.
- Sitio web: Enterprise website builder. Status: Instalar.
- Proyecto: Organice y programe sus proyectos. Status: Instalar.
- Partes de horas: Validación del parte de tiempo y vista de rejilla. Status: Actualizar.
- Inventory: Gestione sus actividades de stock y logística. Status: Instalar.
- Código de barras: Use lectores de código de barras para procesar operaciones logísticas. Status: Actualizar.
- Facturación: Facturas y pagos. Status: Aprenda más.
- Contabilidad: Gestión de la contabilidad financiera y analítica. Status: Instalar.
- Fabricación: Fabricar órdenes & listas de materiales. Status: Aprenda más.

Below the store, the developer tools Network tab displays a timeline of requests:

Name	Status	Type	Initiator	Size	Time	Waterfall
web/	200	document	Other	4.2 kB	27 ms	
web.assets_common.css	200	stylesheet	(index)	(memory cache)	0 ms	
web.assets_backend.css	200	stylesheet	(index)	(memory cache)	0 ms	
web.assets_common.js	200	script	(index)	(memory cache)	0 ms	
web.assets_backend.js	200	script	(index)	(memory cache)	0 ms	
48a66c6539b7872f148753f6d12fc40a7d2db1	200	xhr	(index):32	(disk cache)	4 ms	
767d554a3c1463ff649edca19dc8b8c2fc6cf50?mod=base...acco...	200	xhr	web.assets_common.js:1325	(disk cache)	3 ms	
fontawesome-webfont.woff2?v=4.7.0	200	font	:8069/web/static/lib/fontawesome...	(memory cache)	0 ms	
favicon.ico	200	vnd.microsoft.icon	Other	1.5 kB	3 ms	
favicon/	200	x-icon	Other	1.5 kB	10 ms	
2c0c8c87969569b8a25bfd5de4b3b71ac08367d?mod=base...gr...	200	xhr	web.assets_common.js:1325	(disk cache)	4 ms	
es_ES	200	script	web.assets_common.js:3835	(memory cache)	0 ms	

Timeline details: 73 requests, 73.3 kB transferred, 4.3 MB resources, Finish: 50.44 s, DOMContentLoaded: 310 ms, Load: 310 ms.

Lo primero que demanda es un web/ que envía una página html que básicamente inicia el cliente web:

The screenshot shows the developer tools Network tab with the Response tab selected for the initial request:

```

19<script type="text/javascript">
20    odoo.session_info.uid = 2, "is_system": true, "is_admin": true, "user_context": {"lang": "es_ES", "tz": "Europe/Brussels", "uid": 2}
21    odoo.reloadMenus = function () {
22        return new Promise(function(resolve, reject) {
23            function onLoaded() {
24                resolve(oReq.response);
25            }
26            var oReq = new XMLHttpRequest();
27            oReq.open("GET", "/web/webclient/load_menus");
28            oReq.addEventListener("load", onLoaded);
29            oReq.addEventListener("error", reject);
30            oReq.open("GET", '/web/webclient/load_menus/' + odoo.session_info.cache_hashes.load_menus);
31            oReq.send();
32        });
33    }
34    odoo.loadMenusPromise = odoo.reloadMenus();
35</script>
36<link type="text/css" rel="stylesheet" href="/web/content/630-80cef80/web.assets_common.css"/>
37
38

```

Timeline details: 82 requests, 76.9 kB transferred.

Luego comienza a descargar css y js del servidor.

Utiliza paradigmas de programación moderna y configura un SPA.

El servicio Odoo se amplía mediante módulos que están en la ruta indicada en el .conf: /opt/odoo/odoo/addons

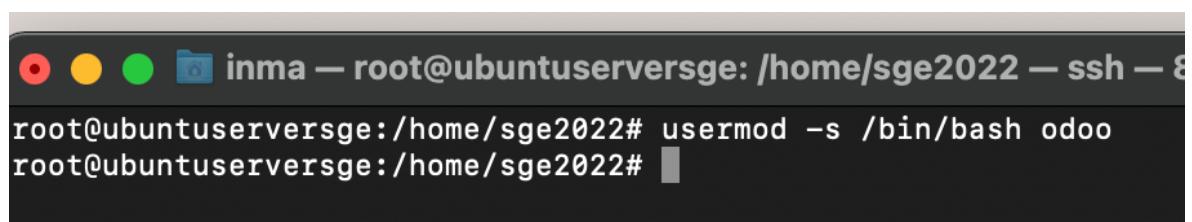
Tenemos un directorio para cada módulo. El módulo base y el web serían unos de ellos.

Si quiero hacer un módulo para Odoo, bastaría con crear un directorio y hacerlo ahí directamente pero hay un problema, ahí están los módulos oficiales y es mejor tener los módulos provisionales o los módulos en desarrollo en un directorio diferente.

Vamos a crear los módulos con el usuario **odoo**, dicho usuario se instala cuando se instala odoo, como puedes ver en **/etc/passwd**.

```
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
el_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuid:x:107:112::/run/uuid:/usr/sbin/nologin
tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin
landscape:x:109:115::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
sshd:x:111:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
alumno:x:1000:1000:Alumno DAM2:/home/alumno:/bin/bash
lxde:x:998:100::/var/snap/lxde/common/lxde:/bin/false
odoo:x:112:117:odoo,,,:/opt/odoo:/bin/bash
postgres:x:113:119:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
alumno@alumnodam:/$
```

- Le establecemos una contraseña a dicho usuario: En modo root hacemos **passwd odoo** y establecemos la nueva contraseña. Entramos como usuario odoo: **su odoo**
- Para poder acceder nos queda una cosa más:



```
inma — root@ubuntuserversge: /home/sge2022 — ssh — 8
root@ubuntuserversge:/home/sge2022# usermod -s /bin/bash odoo
root@ubuntuserversge:/home/sge2022#
```

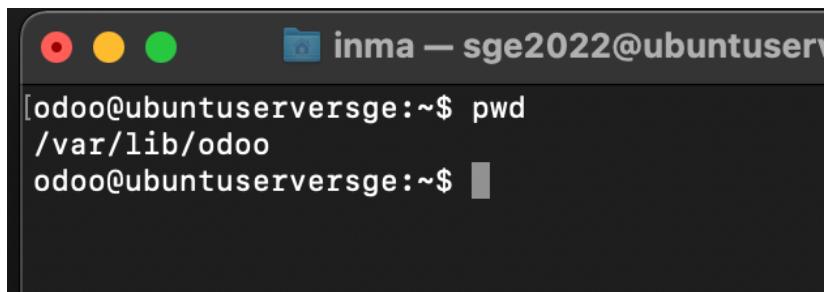
- Vamos a parar el servicio oficial de Odoo como root:

```
[root@ubuntuserversge:/home/sge2022# systemctl stop odoo
root@ubuntuserversge:/home/sge2022#
```

- vamos a comprobar que podemos entrar ya con ese usuario.

```
root@ubuntuserversge:/home/sge2022# exit  
exit  
sge2022@ubuntuserversge:~$ su odoo  
Password:  
odoo@ubuntuserversge:/home/sge2022$
```

- Nos situamos en su directorio home: /[/var/lib/odoo](#)



```
[odoo@ubuntuserversge:~$ pwd  
/var/lib/odoo  
odoo@ubuntuserversge:~$
```

- Creamos un directorio **modules**.
- Tenemos que crear el primer módulo porque no podemos decirle a Odoo que ese directorio es de módulos si no tiene ningún módulo. Vamos a crear un módulo plantilla ejemplo **school**.

```
[odoo@alumnodam:~/modules$ odoo scaffold school  
odoo@alumnodam:~/modules$
```

```
[odoo@alumnodam:~/modules$ ls  
school  
[odoo@alumnodam:~/modules$ cd school  
[odoo@alumnodam:~/modules/school$ ls  
__init__.py __manifest__.py controllers demo models security views  
odoo@alumnodam:~/modules/school$
```

Dentro del directorio **school** se establece la base para poder comenzar a trabajar.

En el fichero de configuración podemos ver:

```
[odoo@ubuntusersges:~/modules/school$ cat /etc/odoo/odoo.conf
[options]
addons_path = /usr/lib/python3/dist-packages/odoo/addons
admin_passwd = $pbkdf2-sha512$25000$d06p1TrnvLe21nqPcU5pjQ$0R4sT2.qqI0nf1JNpa/gH
qsAe6yTarMo9Q5hEiPG9R.lvFQe/RxeJLKF9aJy25psP70YGrAXB2c5ez027Jdog
csv_internal_sep = ,
data_dir = /var/lib/odoo/.local/share/Odoo
db_host = False
db_maxconn = 64
db_name = False
db_password = False
db_port = False
db_sslmode = prefer
db_template = template0
db_user = odoo
dbfilter =
demo = {}
email_from = False
geoip_database = /usr/share/GeoIP/GeoLite2-City.mmdb
http_enable = True
http_interface =
http_port = 8069
import_partial =
limit_memory_hard = 2684354560
```

Addons_path es el path por defecto que carga los paquetes instalados o para instalar en Odoo. Vamos a arrancar nuestro servicio poniendo el path que nos indica por defecto y el directorio **modules** que acabamos de crear, del siguiente modo:

- Iniciamos el servicio indicando que tire de los dos path para los módulos mediante:

```
odoo@ubuntusersges:~/modules/school$ odoo --addons-path="/var/lib/odoo/modules,/usr/lib/python3/dist-packages/odoo addons" --save
2022-02-01 20:08:18,401 3099 INFO ? odoo: Odoo version 14.0-20220201
2022-02-01 20:08:18,401 3099 INFO ? odoo: Using configuration file at /var/lib/odoo/.odoorc
2022-02-01 20:08:18,401 3099 INFO ? odoo: addons paths: ['/usr/lib/python3/dist-packages/odoo/addons', '/var/lib/odoo/.local/share/Odoo/addons/14.0', '/var/lib/odoo/modules']
2022-02-01 20:08:18,402 3099 INFO ? odoo: database: default@default:default
Warn: Can't find .pfb for face 'Times-Roman'
2022-02-01 20:08:18,406 3099 INFO ? odoo.addons.base.models.ir_actions_report: Will use the Wkhtmltopdf binary at /usr/local/bin/wkhtmltopdf
2022-02-01 20:08:12,408 3099 INFO ? odoo.service.server: HTTP service (werkzeug) running on ubuntusersges:8069
```

- Tenemos que tener lo siguiente al hacer `ls -la` en el directorio home del usuario Odoo:

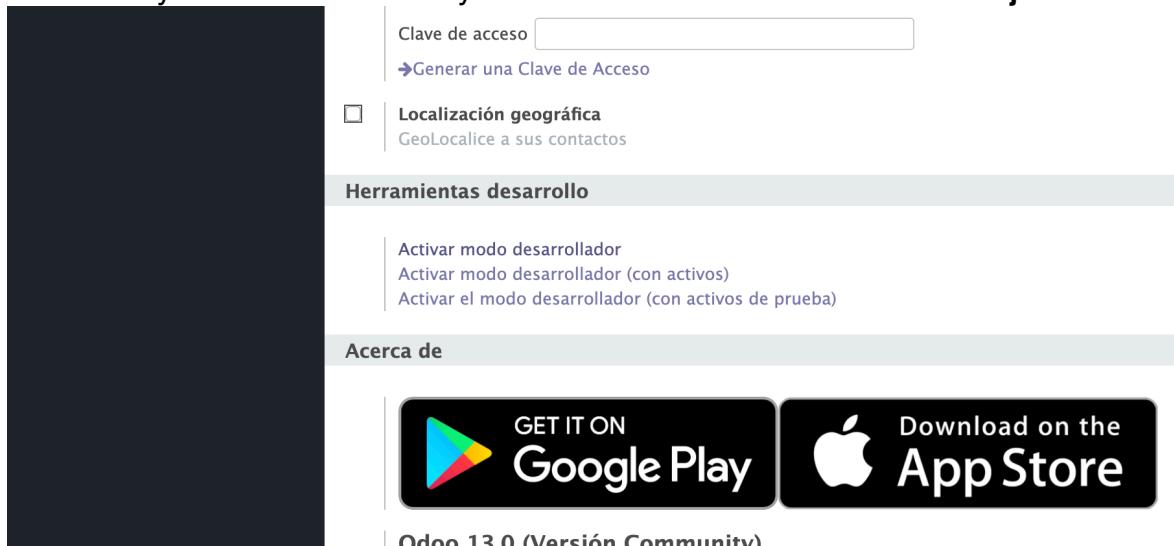
```
[odoo@alumnodam:~$ ls -la
total 36
drwxr-xr-x  5 odoo odoo 4096 Feb  2 11:20 .
drwxr-xr-x  3 root root 4096 Oct 19 18:02 ..
-rw-------  1 odoo odoo  814 Feb  2 10:55 .bash_history
drwx-----  2 odoo odoo 4096 Jan 30 16:58 .cache
drwxrwxr-x  3 odoo odoo 4096 Oct 19 18:20 .local
-rw-------  1 odoo odoo 1205 Feb  2 11:19 .odoorc
-rw-------  1 odoo odoo 5737 Feb  2 10:55 .viminfo
drwxr-xr-x 11 odoo odoo 4096 Feb  2 11:08 odoo
odoo@alumnodam:~$ ]
```

- El usuario odoo tiene su propio **.odoorc** que tiene la misma información que el **.conf**. Pero esto pasa sólo si iniciamos odoo con el usuario odoo desde el

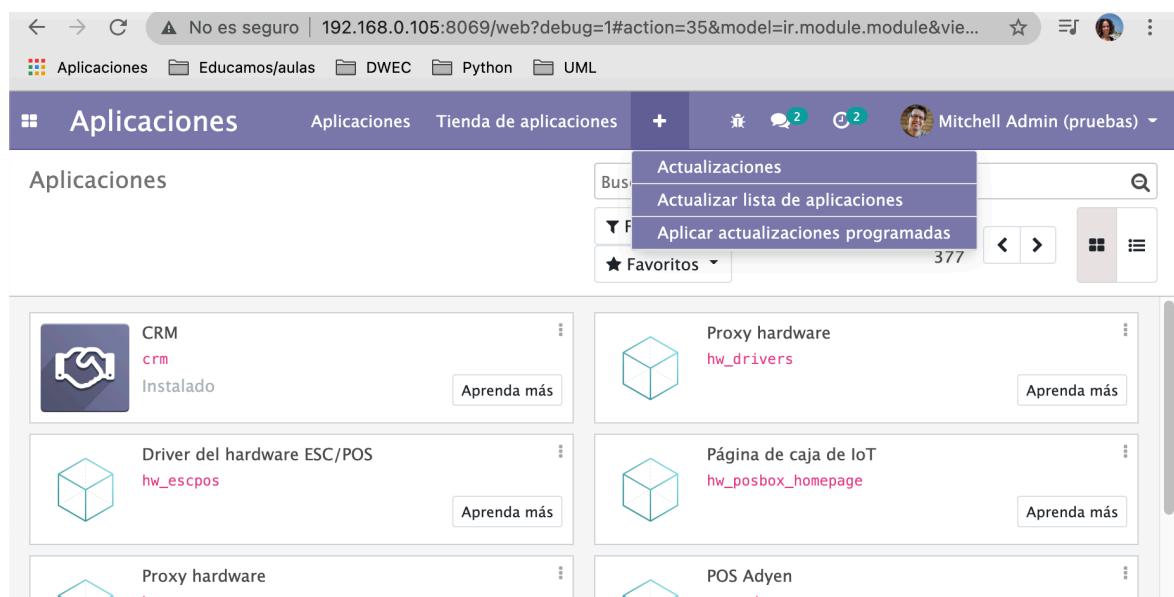
directorio personal de éste porque cogerá dicho fichero .odoorc que se ha creado con la instrucción anterior y tirará del addons-path que tiene incluido el directorio **modules** que acabamos de crear.

- También nos muestra en el terminal el log. Sería la manera de trabajar cuando estamos desarrollando.

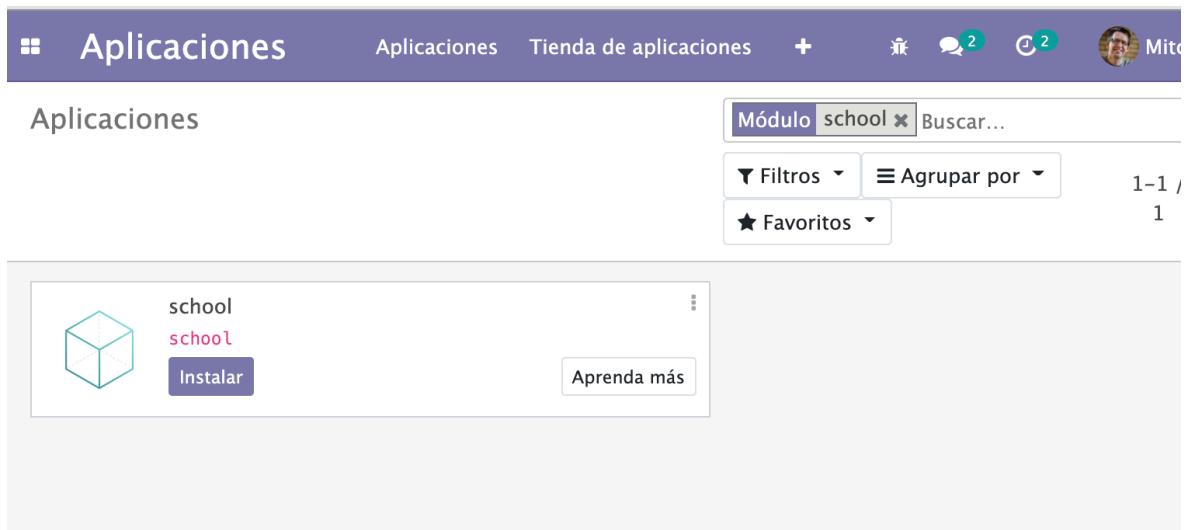
- Vamos a instalar el módulo school. Para ello tenemos que tener el servicio arrancado y nos vamos a **Odoo** y activamos el modo desarrollador en **Ajustes**.



- En el menú de aplicaciones, aparece arriba una opción que nos da la posibilidad de actualizar la lista de aplicaciones:



- Si buscamos nuestro módulo, ya aparece:



- Instalamos el módulo, pero aún no ocurre nada porque no hemos codificado nada para dicho módulo.
- Si modifco alguna cosa en el módulo, tendré que reiniciar el servicio **odoo** y después permitirá **actualizar** el módulo. Hay una forma más rápida de actualizar el módulo que es mediante:

```
inma — alumno@alumnodam: /etc/odoo — ssh — 93x
[odoo@alumnodam:/etc/odoo$ odoo -u school -d pruebas
2021-01-30 18:46:48,251 4402 INFO ? odoo: Odoo version 13.0
2021-01-30 18:46:48,252 4402 INFO ? odoo: Using configuration file at /opt/odoo/odoo.conf
2021-01-30 18:46:48,252 4402 INFO ? odoo: addons paths: ['/opt/odoo/odoo/.local/share/Odoo/addons/13.0', '/opt/odoo/modules', '/opt/odoo/odoo-bin']
2021-01-30 18:46:48,253 4402 INFO ? odoo: database: default@default:default
2021-01-30 18:46:48,352 4402 INFO ? odoo.addons.base.models.ir_actions_report: wkhtmltopdf binary at /usr/local/bin/wkhtmltopdf
2021-01-30 18:46:48,419 4402 INFO ? odoo.service.server: HTTP service (www) started on port 8069
2021-01-30 18:46:48,422 4402 INFO pruebas odoo.modules.loading: loading
2021-01-30 18:46:48,431 4402 INFO pruebas odoo.modules.loading: 1 module queries
2021-01-30 18:46:48,498 4402 INFO pruebas odoo.modules.loading: updating
2021-01-30 18:46:48,501 4402 INFO pruebas odoo.addons.base.models.ir_model: module.update_list on [] to user __system__ #1 via n/a
2021-01-30 18:46:48,802 4402 INFO pruebas odoo.modules.loading: loading
2021-01-30 18:46:49,030 4402 INFO pruebas odoo.modules.loading: 47 module queries
2021-01-30 18:46:49,030 4402 INFO odoo.service.server: HTTP service (www) stopped
```

Con **-u** indicamos el módulo y con **-d** indicamos la bdd sobre la cual se va a actualizar

Entorno de desarrollo

- Vamos a configurar un IDE que se conecte en remoto al servidor para programar. Debemos hacerlo porque tengamos una conexión de forma rápida.

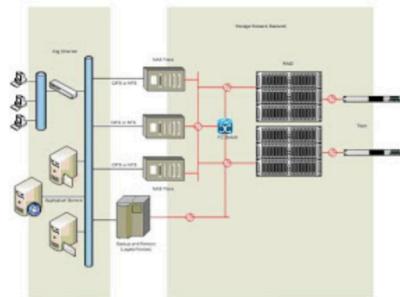
- Para poder utilizar VSCode sobre el directorio del módulo que estamos desarrollando deberíamos poder montar esa carpeta en nuestro sistema anfitrión como una unidad de red.
- Para ello hay distintas opciones, yo voy a mostrar aquí cómo hacerlo mediante nfs.



COMPARTIR DIRECTORIOS EN RED LOCAL CON NFS

NFS o **Network File System** es un *sistema de archivos distribuido* en una red local. Este sistema de archivos distribuido permite que distintos dispositivos a esa red local puedan acceder a ficheros remotos como si estuvieran en el propio dispositivo.

Mediante **NFS** permite compartir un directorio que se encuentra en un dispositivo de la red con otros ordenadores y dispositivos. El ordenador donde está el directorio es el *servidor*, mientras que los dispositivos que se conectan a este *servidor* son los clientes.



Precisamente, un sistema **NFS**, es perfecto para un **NAS** o **Networked Attached Storage** en un medio *Linux*. **NFS** es un protocolo nativo en *Linux*. Justo lo contrario que *Samba* que utiliza el protocolo *SMB* desarrollado por *Microsoft*. Tanto Mac OSX como Windows 7 tienen un buen soporte para NFS. Por todo ello NFS es ideal para compartir directorios en red local.

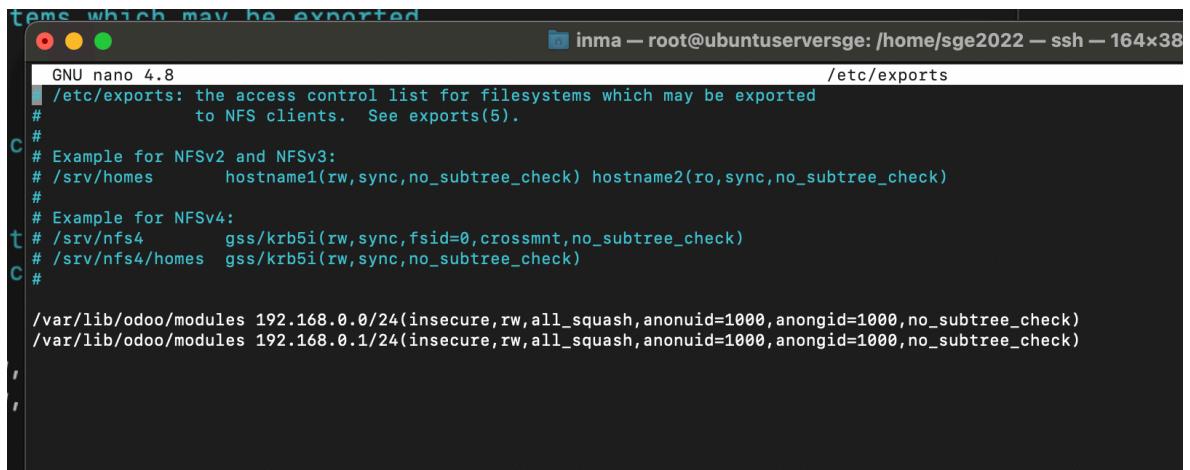
Como he comentado, NFS se compone de un servidor y un cliente. Así es necesario instalar las herramientas necesarias. Tanto en el ordenador donde se encuentra el directorio y que actuará como servidor, como en los otros ordenadores que actuarán como clientes.

CONFIGURACIÓN DEL SERVIDOR

- Instalamos las herramientas necesarias en Ubuntu Server.

```
sudo apt install nfs-kernel-server
```

- Definimos los directorios que queremos compartir. Para esto, editamos el fichero /etc/exports.



```
tente which may be exported
GNU nano 4.8
/etc/exports: the access control list for filesystems which may be exported
#          to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes      hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# # /srv/nfs4      gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# # /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/var/lib/odoo/modules 192.168.0.0/24(insecure,rw,all_squash,anonuid=1000,anongid=1000,no_subtree_check)
/var/lib/odoo/modules 192.168.0.1/24(insecure,rw,all_squash,anonuid=1000,anongid=1000,no_subtree_check)
'
'
```

- Con esto hemos compartido el directorio en cuestión con toda nuestra red. Indicamos que es de lectura y escritura, que no realice comprobación de subdirectorios (lo que en ocasiones mejora el rendimiento). En el caso de que no queramos compartirlo con todos los equipos de nuestra red, sino sólo con uno concreto se pondría su ip.

- Después de cualquier cambio sobre **/etc/exports** tenemos que ejecutar la siguiente orden para actualizar la tabla de NFS.

```
sudo exportfs -arv
```

- La opción **-a** exporta todos los directorios, la opción **-r** eliminará las entradas antiguas y **-v** nos mostrará el resultado de la ejecución.
- Una vez configurado el servidor es necesario reiniciar el servicio para que funcione correctamente.

```
sudo systemctl enable nfs-kernel-server
sudo systemctl start nfs-kernel-server
```

- Debemos asegurarnos que el directorio en el que queremos escribir tiene permisos, para ello en Ubuntu Server:

```
[root@ubuntuserversge:/home/sge2022# chmod -R 777 /var/lib/odoo/modules
root@ubuntuserversge:/home/sge2022# ]
```

- Reiniciamos también el servidor RPC para la monitorización NFS:

```
root@alumnodam:/home/alumno# service rpc-statd restart
root@alumnodam:/home/alumno# service rpc-statd status
● rpc-statd.service - NFS status monitor for NFSv2/3 locking.
  Loaded: loaded (/lib/systemd/system/rpc-statd.service; disabled; vendor preset: enabled)
  Active: active (running) since Fri 2021-02-05 11:02:32 UTC; 2s ago
    Process: 2342 ExecStart=/sbin/rpc.statd --no-notify $STATDARGS (code=exited, status=0/SUCCESS)
   Main PID: 2349 (rpc.statd)
     Tasks: 1 (limit: 4621)
    Memory: 48.8M
      CGroup: /system.slice/rpc-statd.service
              └─2349 /sbin/rpc.statd --no-notify

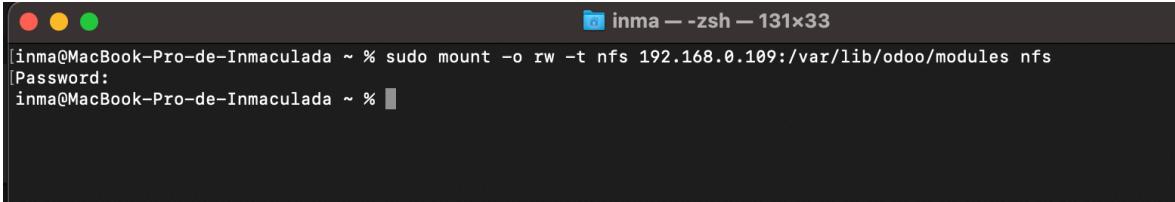
Feb 05 11:02:32 alumnodam systemd[1]: Starting NFS status monitor for NFSv2/3 locking....
Feb 05 11:02:32 alumnodam rpc.statd[2349]: Version 1.3.3 starting
Feb 05 11:02:32 alumnodam rpc.statd[2349]: Flags: TI-RPC
Feb 05 11:02:32 alumnodam systemd[1]: Started NFS status monitor for NFSv2/3 locking..
```

CONFIGURACIÓN DEL CLIENTE

- Dependerá de qué SO tenemos en el anfitrión. En este caso es MAC, para Windows podemos hacer lo siguiente:

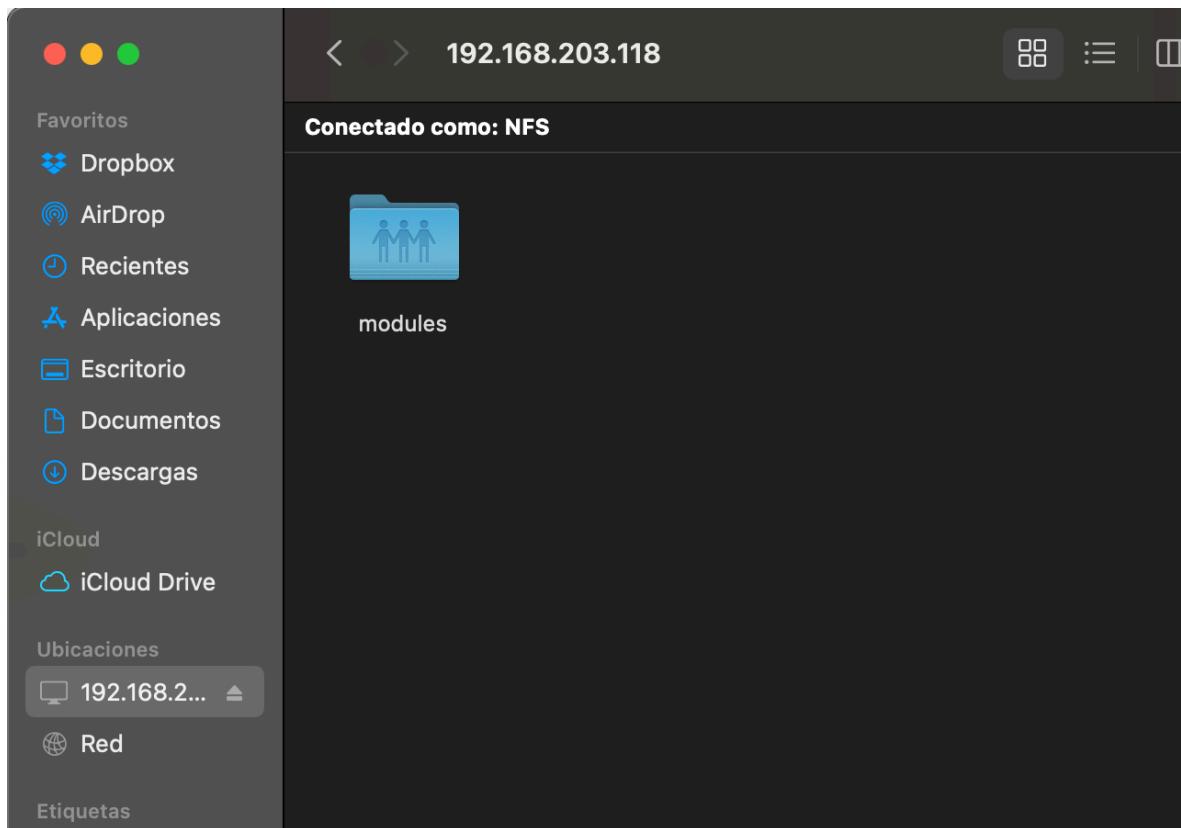
<https://imperioweb.net/compartir-archivos-con-nfs-en-servidor-ubuntu-y-cliente-windows-10>

- En Mac podemos hacerlo desde un terminal con:

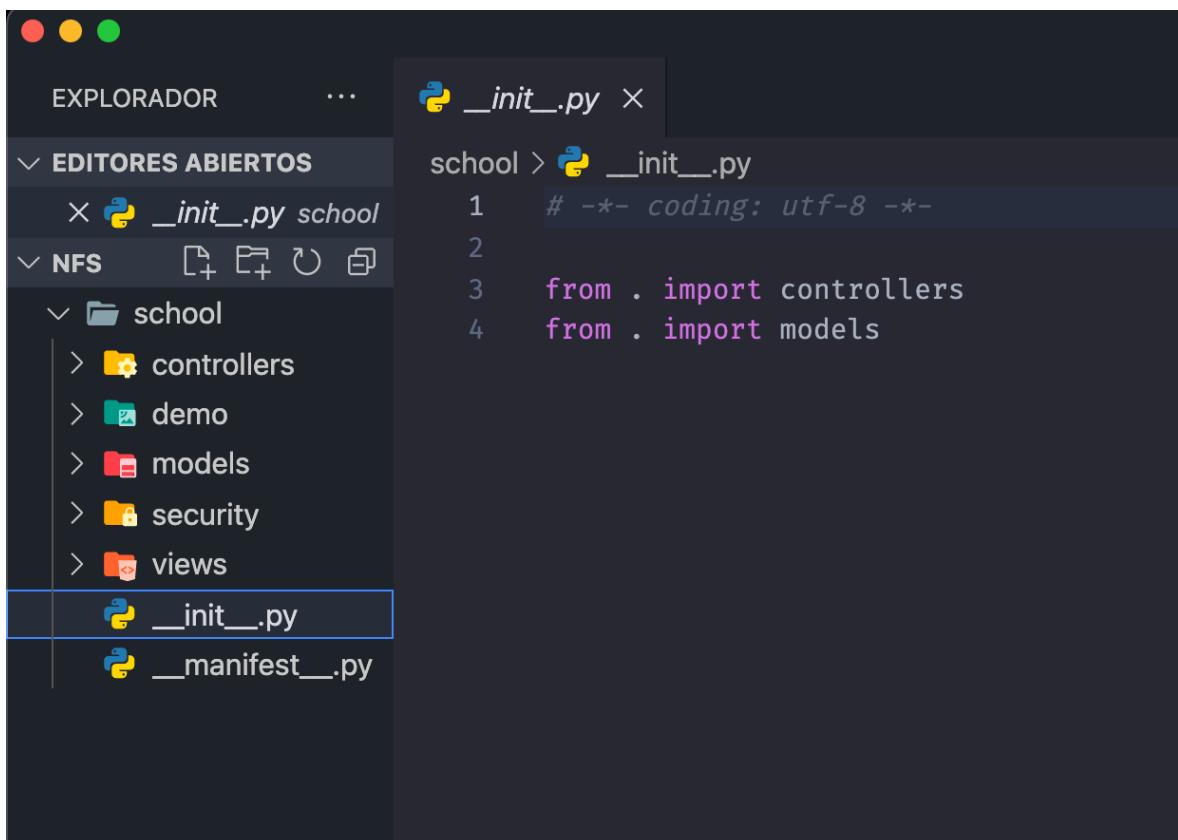


A screenshot of a macOS terminal window titled "inma — zsh — 131x33". The window shows the command "sudo mount -o rw -t nfs 192.168.0.109:/var/lib/odoo/modules nfs" being run. The password prompt "[Password:" is visible. The command is completed with a success message: "inma@MacBook-Pro-de-Inmaculada ~ %".

- Podemos ver nuestra carpeta montada:

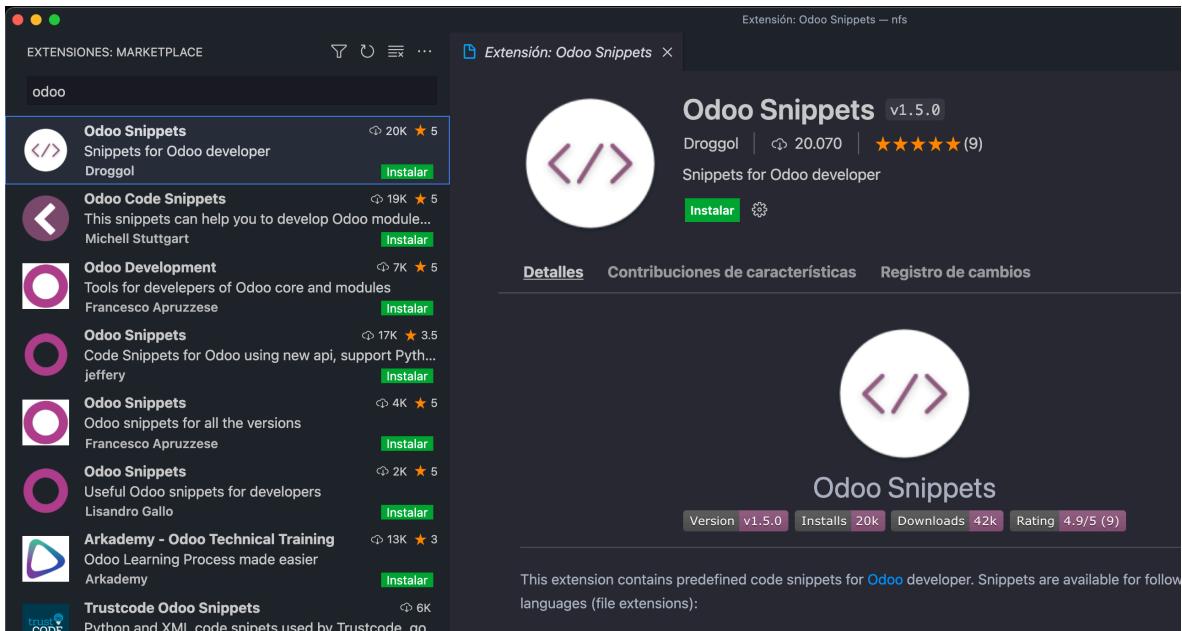


- Desde VSCode ya podemos acceder a ella:

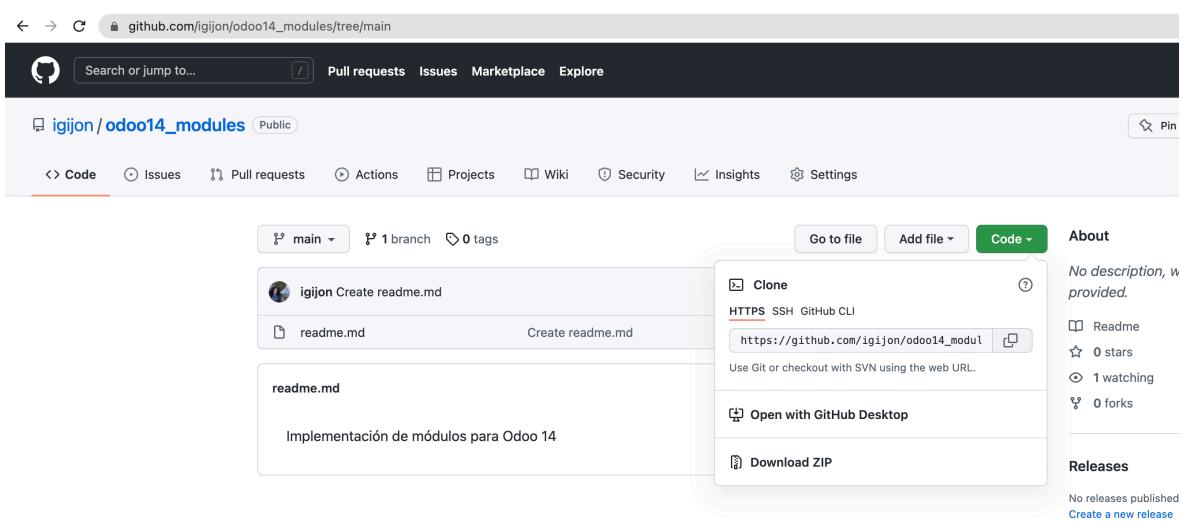


- Es recomendable abrir la terminal en VSCode. De este modo según vamos modificando podemos parar el servicio y volver a conectarlos. En esta terminal nos conectaremos por ssh.

- Podemos instalar como extensiones:



Vamos a añadir control de versiones con Github.



Copio mi enlace y hago lo siguiente:

```
[odoo@ubuntusersversge:~/modules$ git init
Initialized empty Git repository in /var/lib/odoo/modules/.git/
[odoo@ubuntusersversge:~/modules$ git remote add origin https://github.com/igijon/odoo14_modules.git
[odoo@ubuntusersversge:~/modules$ git pull origin main
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 1.36 KiB | 694.00 KiB/s, done.
From https://github.com/igijon/odoo14_modules
 * branch            main      -> FETCH_HEAD
 * [new branch]      main      -> origin/main
[odoo@ubuntusersversge:~/modules$ ls -la
total 20
drwxrwxrwx 4 odoo odoo 4096 feb  3 15:30 .
drwxr-xr-x 5 odoo odoo 4096 feb  2 20:02 ..
drwxrwxr-x 8 odoo odoo 4096 feb  3 15:30 .git
-rw-rw-r-- 1 odoo odoo 105 feb  3 15:30 README.md
drwxrwxrwx 7 odoo odoo 4096 feb  3 15:29 school
odoo@ubuntusersversge:~/modules$ ]
```

```
[odoo@ubuntusersversge:~/modules$ git add *
[odoo@ubuntusersversge:~/modules$ git commit -m "Mi primer módulo"
[master 156777f] Mi primer módulo
 10 files changed, 200 insertions(+)
 create mode 100755 school/__init__.py
 create mode 100755 school/__manifest__.py
 create mode 100755 school/controllers/__init__.py
 create mode 100755 school/controllers/controllers.py
 create mode 100755 school/demo/demo.xml
 create mode 100755 school/models/__init__.py
 create mode 100755 school/models/models.py
 create mode 100755 school/security/ir.model.access.csv
 create mode 100755 school/views/templates.xml
 create mode 100755 school/views/views.xml
odoo@ubuntusersversge:~/modules$ ]
```

Si nos pide configurar email y nombre, ejecutamos dichos comandos de este modo:

```
[odoo@ubuntusersversge:~/modules$ git config --global user.email "igijoninf@gmail.com"
[odoo@ubuntusersversge:~/modules$ git config --global user.name "Inma Gijón"]
```

Después volveríamos a hacer el commit correspondiente.

Por defecto, en Ubuntu, se crea una rama master, en GitHub, la rama principal por defecto es main.

Para saber en qué rama estoy hacemos:

```
[odoo@ubuntusersversge:~/modules$ git branch
* master
odoo@ubuntusersversge:~/modules$ ]
```

Vamos a movernos a la rama main y mergear lo que tenemos en master. De este modo ... lo tendremos todo actualizado en Github. Vamos a trabajar de momento en la rama main.

```

[odoo@ubuntusersversge:~/modules$ git checkout main
Branch 'main' set up to track remote branch 'main' from 'origin'.
Switched to a new branch 'main'
[odoo@ubuntusersversge:~/modules$ git merge master
Updating e8f3923..156777f
Fast-forward
 school/__init__.py |    4 +++
 school/__manifest__.py | 35 ++++++=====
 school/controllers/__init__.py |   3 +++
 school/controllers/controllers.py | 21 ++++++=====
 school/demo/demo.xml | 30 ++++++=====
 school/models/__init__.py |    3 +++
 school/models/models.py | 18 ++++++=====
 school/security/ir.model.access.csv |  2 ++
 school/views/templates.xml | 24 ++++++=====
 school/views/views.xml | 60 ++++++=====
10 files changed, 200 insertions(+)
create mode 100755 school/__init__.py
create mode 100755 school/__manifest__.py
create mode 100755 school/controllers/__init__.py
create mode 100755 school/controllers/controllers.py
create mode 100755 school/demo/demo.xml
create mode 100755 school/models/__init__.py
create mode 100755 school/models/models.py
create mode 100755 school/security/ir.model.access.csv
create mode 100755 school/views/templates.xml
create mode 100755 school/views/views.xml
[odoo@ubuntusersversge:~/modules$ git pull
Already up to date.

```

Ahora vamos a hacer un push para que en el repositorio remoto de Github tengamos todos los contenidos sincronizados. Nos va a pedir usuario y contraseña. Github no permite contraseñas y tenemos que generar un token de autenticación y pegarlo en su lugar

Entramos en Github, en nuestro perfil > Settings > Developer Settings > Personal Access Token > Generate new token

The screenshot shows the GitHub developer settings page. At the top, there is a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below that is a breadcrumb trail: Settings / Developer settings. On the left, there is a sidebar with three options: GitHub Apps, OAuth Apps, and Personal access tokens, with Personal access tokens being the active tab. The main content area is titled "Personal access tokens" and contains a "Generate new token" button. A note below the title says: "Need an API token for scripts or testing? [Generate a personal access token](#) for quick access to the [GitHub API](#)." Another note at the bottom states: "Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#)."



© 2022 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

The screenshot shows the GitHub developer settings page for creating a new personal access token. The token is named 'VSCode' and is intended for 'VSCode'. It has an expiration date of '60 days' (Sat, Apr 2 2022). The 'Select scopes' section lists various GitHub permissions. Most checkboxes are unchecked except for 'repo', 'workflow', 'write:packages', 'delete:packages', 'admin:org', and 'admin:public_key'. A note at the bottom states: 'Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.'

Le voy a dar acceso a todo de momento.

Genero el token y lo copio.

Lo vuelvo a hacer y cuando me pide la contraseña en el terminal, pego el token:

```
odoo@ubuntuserversge:~/modules$ git push
Username for 'https://github.com': igijon
Password for 'https://igijon@github.com':
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Compressing objects: 100% (15/15), done.
Writing objects: 100% (18/18), 2.94 KiB | 753.00 KiB/s, done.
Total 18 (delta 0), reused 0 (delta 0)
To https://github.com/igijon/odoo14_modules.git
  e8f3923..156777f main -> main
odoo@ubuntuserversge:~/modules$
```

En el repo debemos tener:

github.com/gijon/odoo14_modules

Search or jump to... Pull requests Issues Marketplace Explore

igijon / odoo14_modules Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

igijon mi primer módulo 92ab451 1 minute ago 2 commits

school mi primer módulo 1 minute ago

readme.md Create readme.md 24 minutes ago

readme.md

Implementación de módulos para Odoo 14

This screenshot shows a GitHub repository page for 'odoo14_modules' owned by 'gijon'. The repository is public. At the top, there are links for Pull requests, Issues, Marketplace, and Explore. Below that is a navigation bar with Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area shows a commit history for the 'main' branch. The first commit is from 'igijon' titled 'mi primer módulo' with hash '92ab451' made 1 minute ago. This is followed by a commit from 'school' titled 'mi primer módulo' made 1 minute ago. Below the commits is a file named 'readme.md' with the content 'Implementación de módulos para Odoo 14'.

Podemos utilizar VSCode para gestionar el repositorio desde el anfitrión:

The screenshot shows the VSCode interface with a dark theme. In the top left, there's a message: 'Mensaje (⌘Enter para confirmar en "main")'. The top right shows tabs for '_manifest__.py' and '_manifest__.py (árbol de trabajo)'. On the far right, there's a status bar with 'read'. The main area has a 'Cambiros' (Changes) sidebar on the left and a 'Graph' view on the right. The 'Graph' view displays a git commit history with nodes for 'main' and 'origin' branches. The commits shown are: 'Pruebas Github en remoto' (blue node), 'prueba commit inicial' (blue node), 'mi primer módulo' (blue node), 'primer módulo' (pink node), and 'Create readme.md' (blue node). The 'Description' column provides details for each commit.

← → ⌂ github.com/igijon/odoo14_modules

Search or jump to... Pull requests Issues Marketplace Explore

igijon / odoo14_modules Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

igijon Pruebas Github en remoto cec0bc3 1 minute ago 4 commits

 school Pruebas Github en remoto 1 minute ago

 readme.md Pruebas Github en remoto 1 minute ago

readme.md

Implementación de módulos para Odoo 14
Módulo school inicial para Sistemas de Gestión Empresarial 2022

About No description, website, or topics provided.

Readme 0 stars 1 watching 0 forks

Releases No releases published Create a new release

Packages No packages published Publish your first package

Languages

