Kevin Gómez Valderas 2ºDAM

**INSTALAR ODOO DOCKERS**

**1. Utiliza una máquina virtual con Ubuntu Server.**

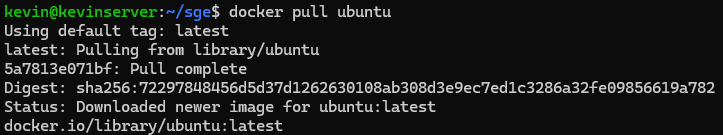
**2. Instala el paquete docker-compose. Puedes añadir tu usuario al grupo docker para hacer los comandos docker sin necesidad de sudo**

| Actualizar el sistema |
| --- |
| sudo apt update && sudo apt upgrade -y |
| Instalar dependencias necesarias |
| sudo apt install apt-transport-https ca-certificates curl software-properties-common -y |
| Agregar la clave GPG oficial de Docker |
| curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg |
| Agregar el repositorio de Docker |
| echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null |
| Actualizar e instalar Docker |
| sudo apt update  sudo apt install docker-ce docker-ce-cli containerd.io -y |
| Verificar la instalación de Docker |
| sudo docker --version |
| Instalar Docker Compose y establecer permisos |
| sudo curl -L "https://github.com/docker/compose/releases/download/v2.23.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose |
| sudo chmod +x /usr/local/bin/docker-compose |
| Verificar la instalación de Docker Compose |
| docker-compose --version |

| Añadir usuario al grupo docker |
| --- |
| Permite ejecutar comandos docker sin necesidad de usar sudo |
| sudo usermod -aG docker $USER |
| newgrp docker |

**3. Descarga una imagen de un docker de ubuntu con docker pull**

docker pull ubuntu



**1. Lista las imagenes que tienes disponibles en docker (solo debes tener la de ubuntu)**

docker images



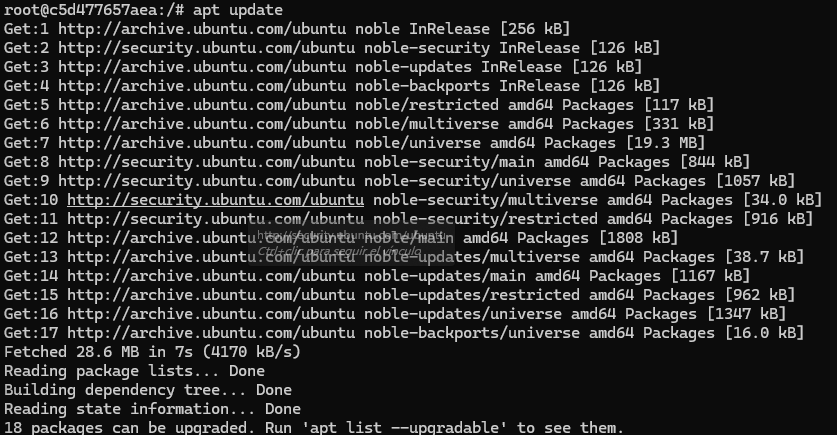
**2. Pon en marcha el contenedor con docker run (dale de nombre ubuntu-tunombre) de manera iterativa con -it**

docker run -it --name ubuntu-kevin ubuntu

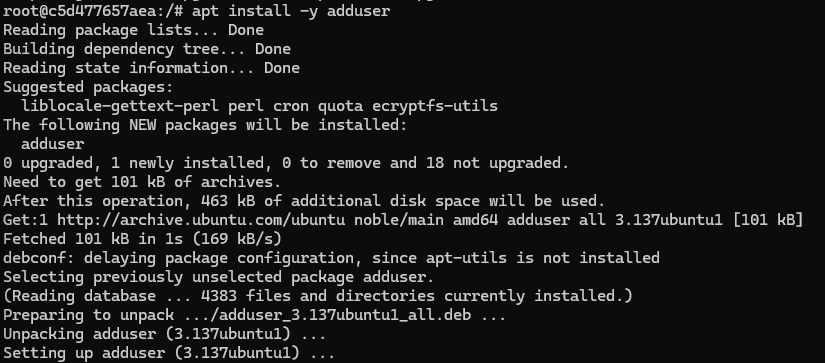


**3. Crea un usuario con tu nombre dentro del contenedor**

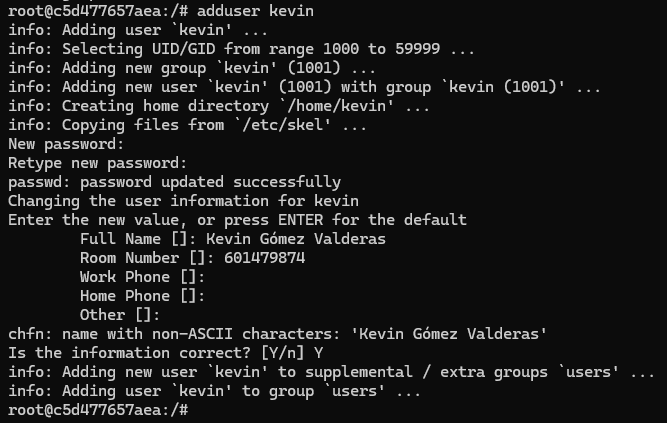
apt update



apt install -y adduser



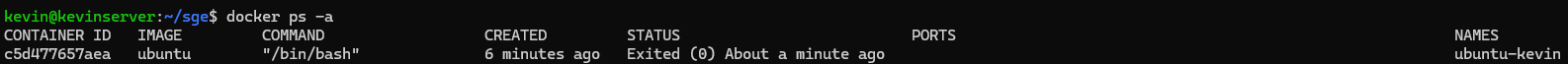
adduser kevin



**4. Comprueba con docker ps los contenedores que hay en marcha. Y con docker ps -a los que están corriendo y los que no.**

docker ps

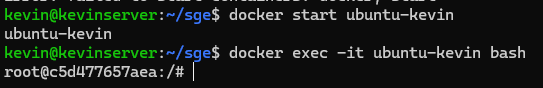
docker ps -a



**5. Pon en marcha el contendor ubuntu-tunombre con el comando docker start**

docker start ubuntu-kevin

docker exec -it ubuntu-kevin bash

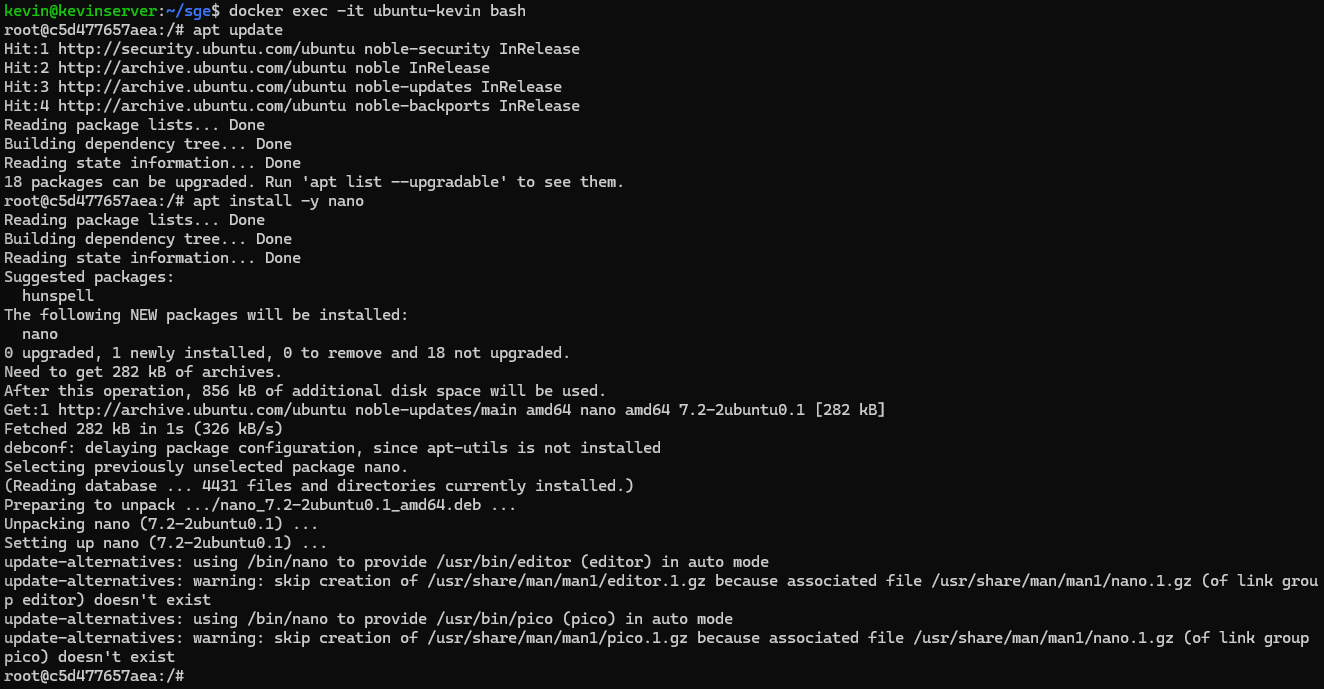


**6. Ejecuta en este contenedor el comando /bin/bash. Instala el editor nano y edita un fichero de texto expresando todo lo que disfrutas en el módulo SGE.**

docker exec -it ubuntu-kevin /bin/bash

apt update

apt install -y nano



nano sge\_enjoyment.txt



**7. Crea un directorio llamado dirdockcomp en tu sistema (en la maquina virtual estamos hablando).**

mkdir dirdockcomp



**8. Crea un nuevo contenedor llamado ubuntu-tunombre2 donde tengas un bind volume (opcion -v) en el que asocies el directorio dirdockcomp con el directorio /mnt/dirdcomp del contenedor.**

docker run -it --name ubuntu-kevin2 -v ~/sge/dirdockcomp:/mnt/dirdcomp ubuntu



**9. Ejecuta /bin/bash y crea un nuevo fichero en el directorio /mnt/dirdocker. Comprueba que aparece en el dirdocker de tu sistema.**

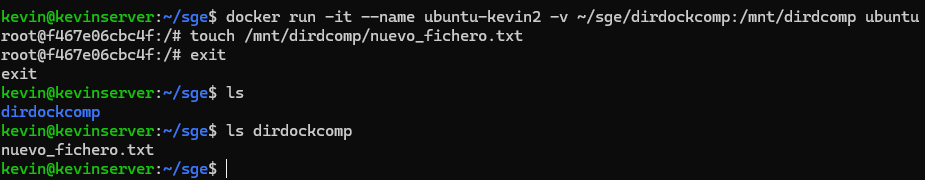
docker start ubuntu-kevin2

docker exec -it ubuntu-kevin2 /bin/bash

touch /mnt/dirdcomp/nuevo\_fichero.txt

exit

ls dirdockcomp

****

**10. Busca la diferencia entre volúmenes docker y bind mounts. ¿Qué has utilizado en los puntos anteriores?**

|  | **VOLÚMENES DOCKER** | **BIND MOUNTS** |
| --- | --- | --- |
| **GESTIÓN** | Los volúmenes son gestionados por Docker y se almacenan en un directorio específico en el host, generalmente dentro de /var/lib/docker/volumes/. | Los bind mounts enlazan un archivo o directorio específico del host directamente con el contenedor. El archivo o directorio debe existir en el host. |
| **PORTABILIDAD** | Son más portables y fáciles de respaldar o migrar, ya que Docker se encarga de su administración. | Son menos portables, ya que dependen de la estructura de archivos del host. |
| **RENDIMIENTO** | Suelen tener un mejor rendimiento en comparación con los bind mounts, especialmente en sistemas con muchos archivos. | Pueden ser más lentos que los volúmenes, especialmente si el host tiene un sistema de archivos lento. |
| **USO** | Ideal para almacenar datos persistentes, como bases de datos o archivos de configuración que deben sobrevivir a la eliminación de contenedores. | Útiles para desarrollo, donde se necesita sincronizar código fuente entre el host y el contenedor, o para acceder a archivos específicos del host. |

En los puntos anteriores, utilize bind mounts al vincular el directorio ~/sge/dirdockcomp del host con /mnt/dirdcomp en el contenedor.

**4. En la siguiente guia puedes ver como instalar odoo en docker. La primera parte es de como instalar docker con los repositorios originales. Empieza en la parte de comprobar que docker está instalado**

**(https://www.cybrosys.com/blog/how-to-install-odoo-16-in-docker).**

**1. Crea los ficheros de configuración en el directorio docker\_TUNOMBRE/odoo17**

| Crear carpeta para odoo docker |
| --- |
| mkdir sge/odoo-docker  cd sge/odoo-docker |
| Crear el archivo docker-compose.yml |
| nano docker-compose.yml  version: '3.1'  services:  web:  image: odoo:17.0  container\_name: odoo17  depends\_on:  - db  ports:  - "8069:8069"  volumes:  - odoo-data:/var/lib/odoo  - ./config:/etc/odoo  - ./addons:/mnt/extra-addons  environment:  - HOST=db  - USER=db\_user  - PASSWORD=db\_passwd  db:  image: postgres:13  container\_name: postgres13  environment:  - POSTGRES\_DB=postgres  - POSTGRES\_USER=db\_user  - POSTGRES\_PASSWORD=db\_passwd  volumes:  - postgres-data:/var/lib/postgresql/data  volumes:  odoo-data:  postgres-data: |
| Iniciar los contenedores en segundo plano |
| sudo docker-compose up -d |
| Ya se puede acceder a odoo web desde http://192.168.56.101:8069 |
| Se configura odoo.conf automáticamente aunque es recomendable configurar odoo.conf manualmente |
| Configurar odoo.conf |
| Se habrá creado la carpeta config y en ella creamos el archivo odoo.conf |
| sudo nano config/odoo.conf |
| [options]  db\_host = db  db\_port = 5432  db\_user = db\_user  db\_password = db\_passwd |

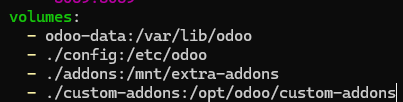
**2. Fijate en el fichero de configuración como indica los volumenes docker y los bind mounts. Queremos que custom-addons (dentro de tu directorio de configuración docker de la máquina virtual ) lo monte en directorio /opt/odoo/custom-addons del contenedor docker.**

mkdir custom-addons

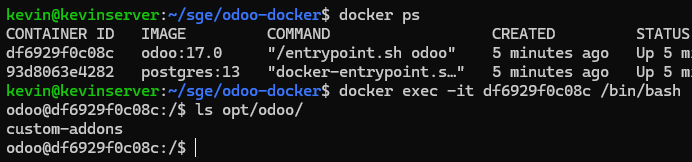


nano docker-compose.yml

- ./custom-addons:/opt/odoo/custom-addons

****

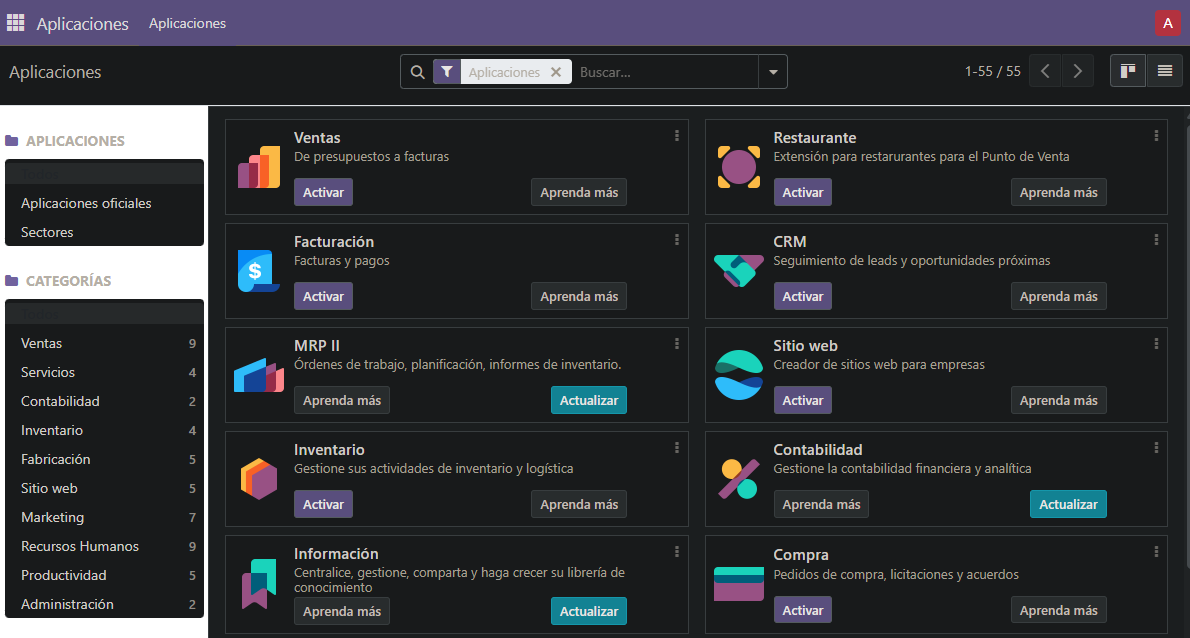
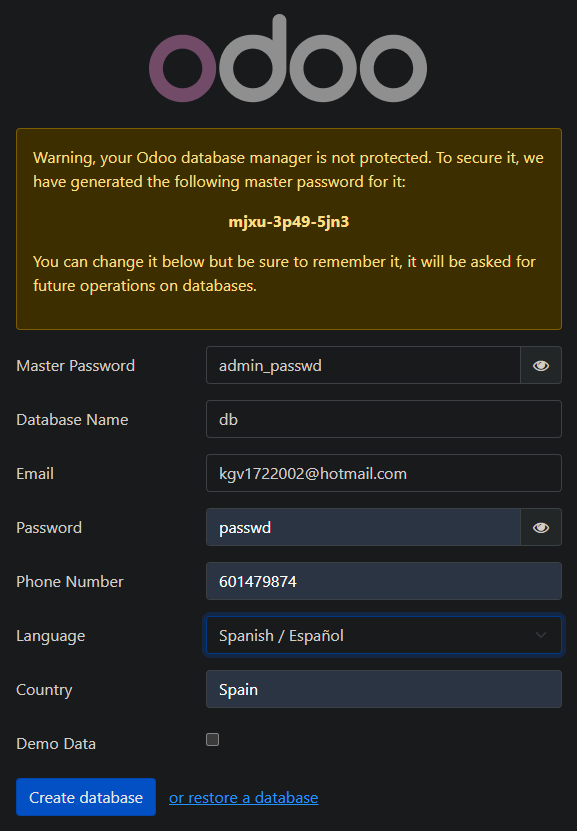
**5. Ejecuta /bin/bash para acceder al contendor de odoo y comprueba que el contenido del directorio /opt/odoo corresponde al directorio asociado en la máquina virtual.**

****

**COMPROBANDO LA INSTALACIÓN DE ODOO**

**6. Accede a Odoo desde tu ordenador y realiza la misma configuración inicial (crear empresa, cambiar logo,…) que hiciste en la práctica anterior (instalar odoo desde paquetes).**

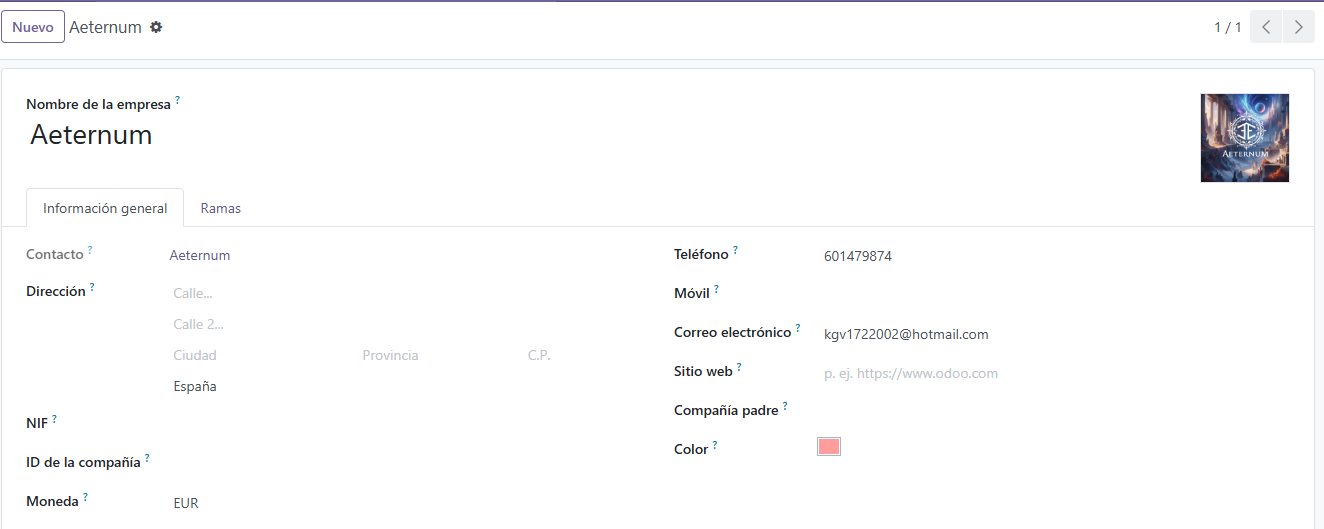
sudo docker-compose up -d



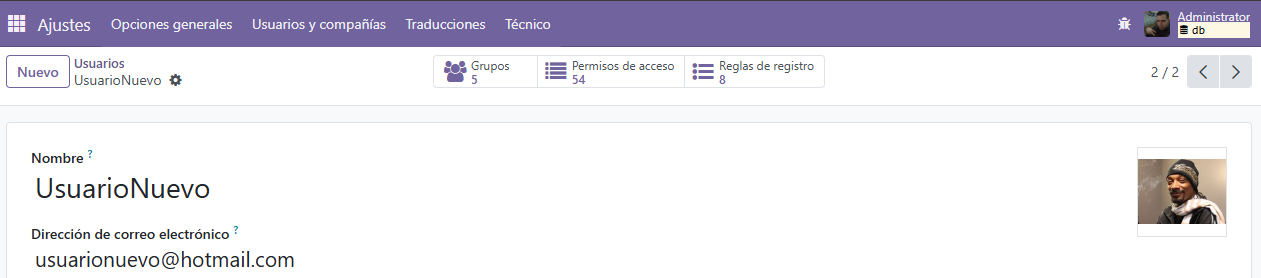
**6. Una vez instalado y que ya puedas acceder desde el navegador vamos a repetir aproximadamente lo que hicimos con Odoo Saas.**

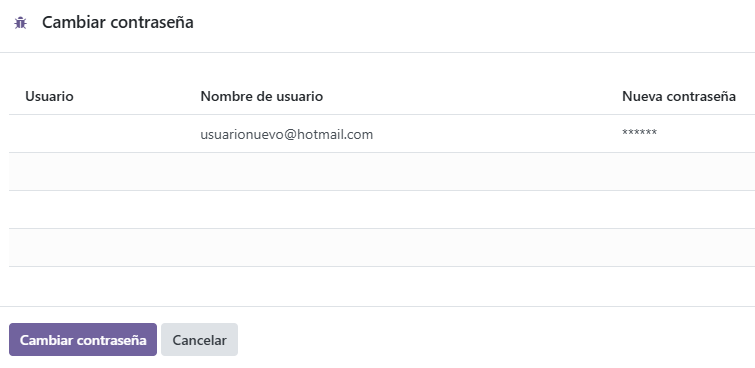
**a) Crea la primera empresa (el correo no hace falta que sea real).**

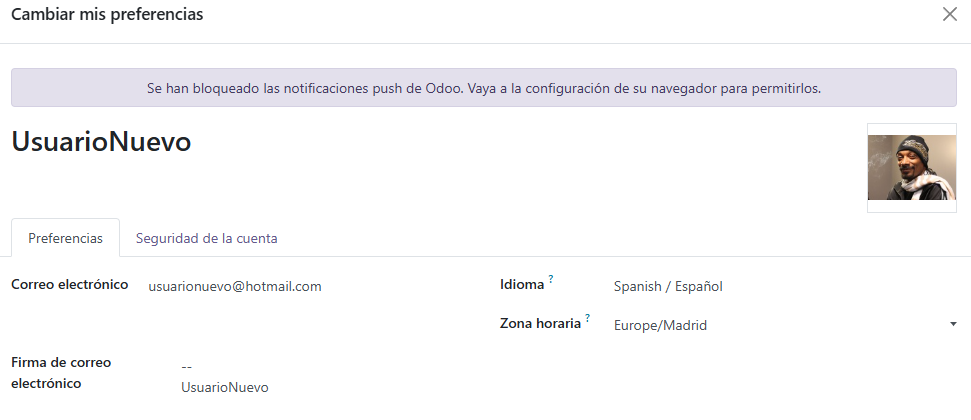
**b) Pon un logo a tu empresa**

****

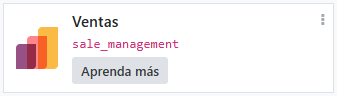
**c) Crea un usuario con un correo inventado. Cambiale la contraseña tu mismo y prueba que puedes conectarte con este usuario**

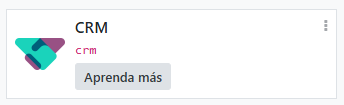
****

****

****

**d) Activa el módulo de ventas y CRM.**

****

****

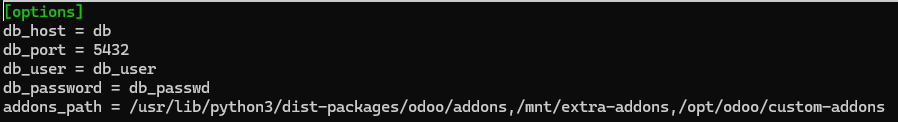
**7. Repite la instalación de los módulos de la comunidad (l10n\_es\_partner y l10n\_es\_toponyms. ) que ya realizaste en la práctica anterior**

1. Modificar odoo.conf

sudo nano odoo.conf

Indicar las rutas de la ubicación de los módulos:

addons\_path = /usr/lib/python3/dist-packages/odoo/addons,/mnt/extra-addons,/opt/odoo/custom-addons

****

En addons pondré los módulos ya creados y custom-addons los míos.

En addons clonamos el repositorio con l10n-spain.git

2. Importar módulos

sudo git clone -b 17.0 https://github.com/OCA/l10n-spain.git

De l10n-spain.git extraemos l10n\_es\_partner y l10n\_es\_toponyms

sudo cp -r l10n\_es\_partner ~/sge/odoo-docker/addons/

sudo cp -r l10n\_es\_toponyms ~/sge/odoo-docker/addons/



3. Importar dependencias

Dependencias de l10n\_es\_toponyms:

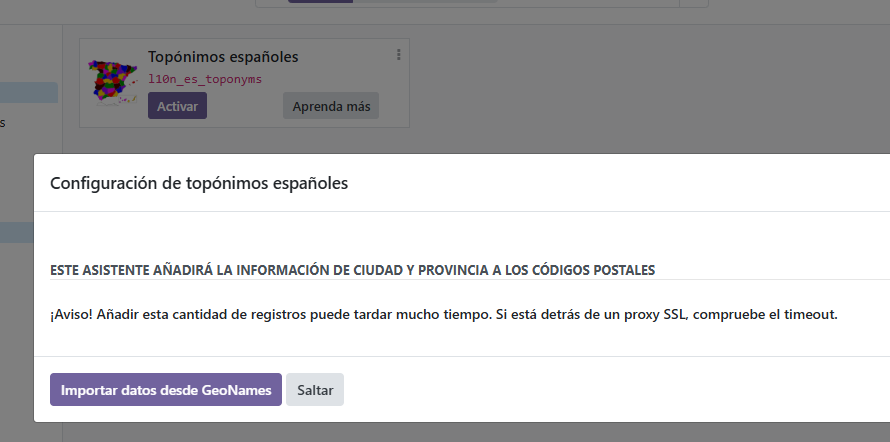
sudo git clone -b 17.0 <https://github.com/OCA/partner-contact.git>

cd partner-contact

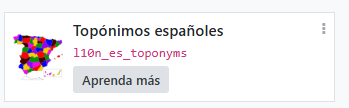
sudo cp -r base\_location ~/sge/odoo-docker/addons/

sudo cp -r base\_location\_geonames\_import ~/sge/odoo-docker/addons/

Actualizar lista de aplicaciones



Importar datos desde GeoNames, si se va la conexión darle a saltar:



Dependencias de l10n\_es\_partner:

sudo git clone -b 17.0 <https://github.com/OCA/community-data-files.git>

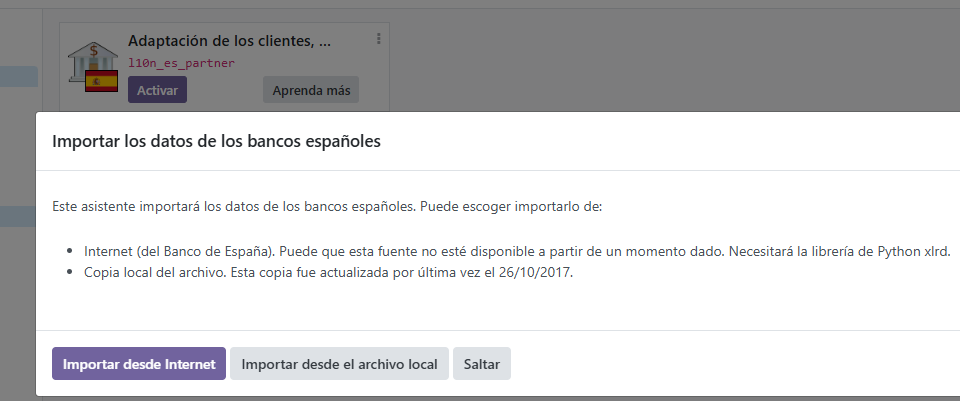
cd community-data-files

sudo cp -r base\_bank\_from\_iban ~/sge/odoo-docker/addons/

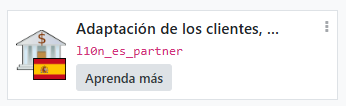
docker exec -it odoo17 bash

pip install schwifty==2024.4.0

Actualizar lista de aplicaciones



Pulsar importar desde internet



4. Actualización de aplicaciones

Actualizar ambas aplicaciones para asegurar la versión más reciente.

