

# **Sistemas de Gestión Empresarial (SGE)**

## **Trabajo: Actividad 8. UD2. Instalación de Odoo en Contenedores Docker**

---

**Alumno:** Blagovest Doukov Dimitrova

**Curso:** 2º DAM

**Profesor:** ANTONIO CECILIA ANTON

**Fecha de entrega:** 24/11/2025

ESTA ES LA ENTREGA DEFINITIVA

## Odo 16 en Docker: - Parte 1: Contenedor Odoo en producción

1. Creamos el contenedor de PostgreSQL con el siguiente comando en la terminal indicando todo lo necesario para crear correctamente el contenedor como el directorio en el cual lo queremos almacenar, el nombre del contenedor...



2. Creamos el contenedor de odoo una vez arrancado el postgre pero en los apuntes hay un fallo y es que pone -v pero no especifica ningún volumen ni en los apuntes especifica nada de él, decidido quitar la opción de -v.

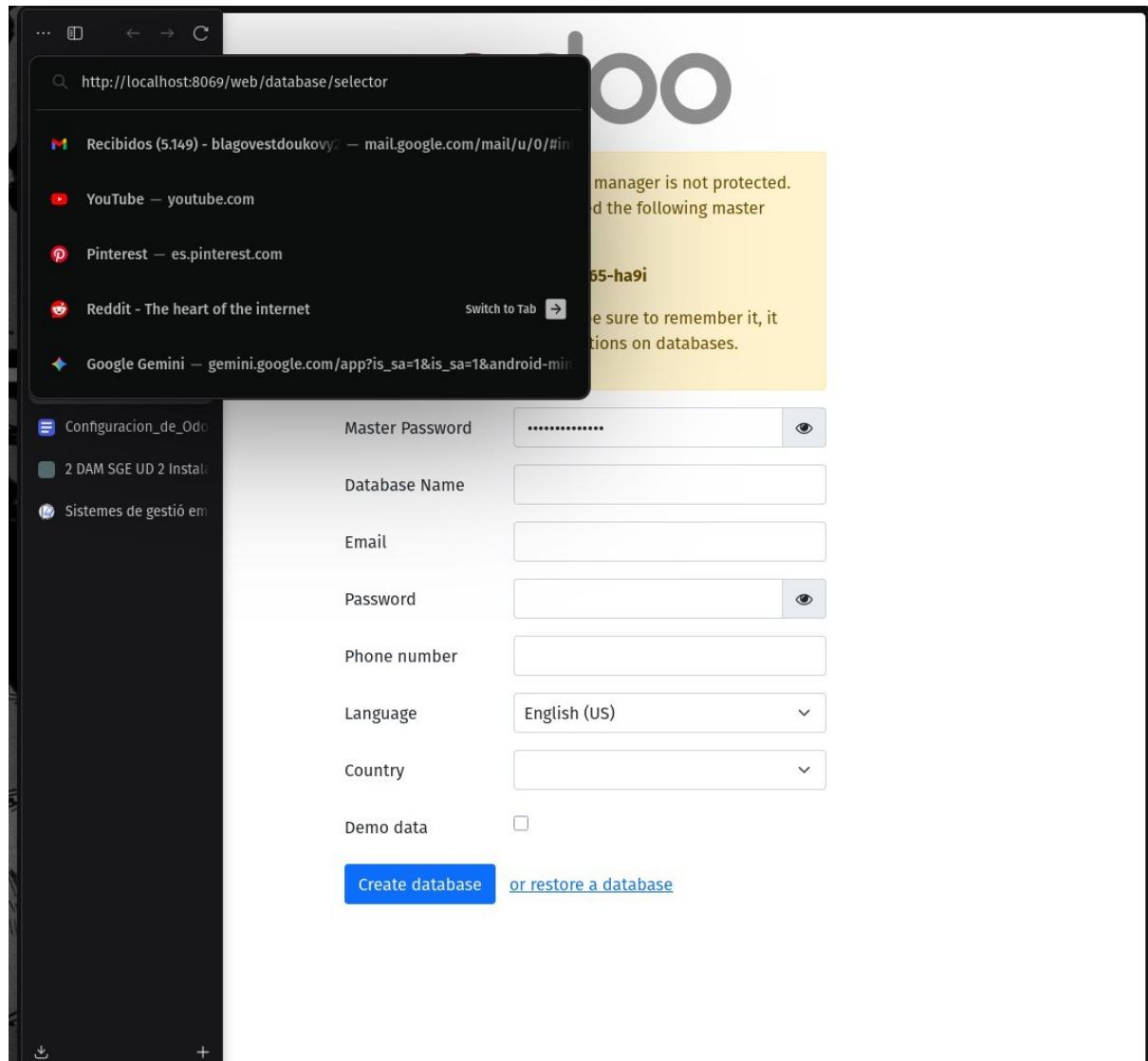
Mal:

```
dan ~ 16:05 > docker run -d -v -p 8069:8069 --name odooprod --user="root" --link db:db odoo:16
```

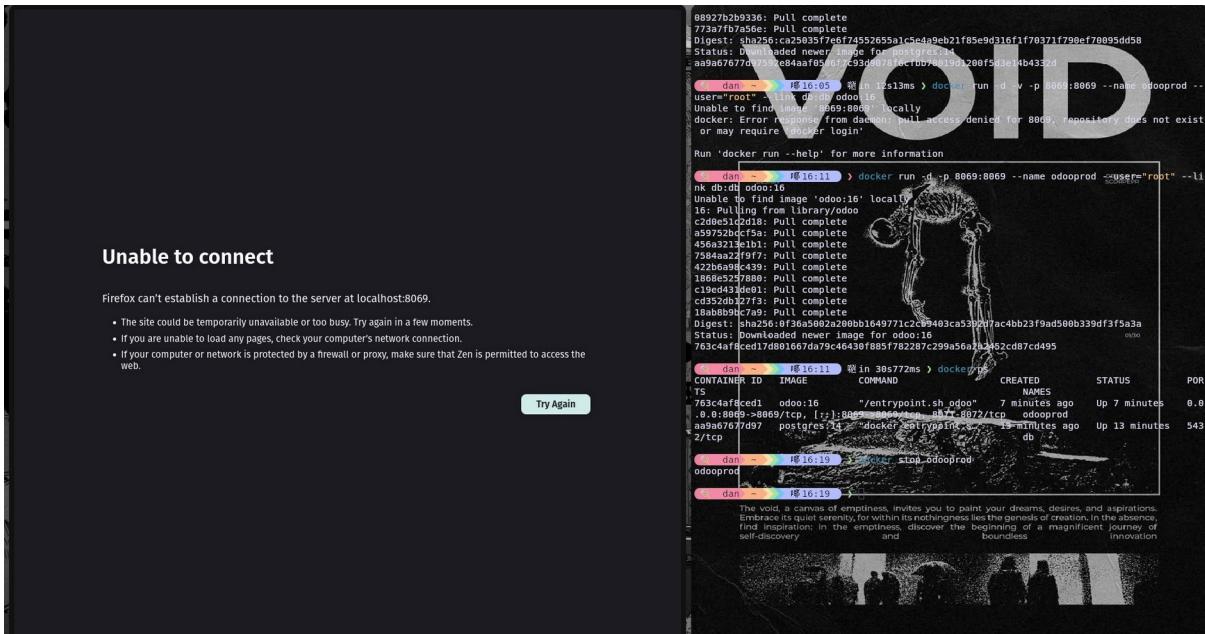
Bien:

```
dan ~ 16:11 > docker run -d -p 8069:8069 --name odooprod --user="root" --link db:db odoo:16
Unable to find image 'odoo:16' locally
16: Pulling from library/odoo
c2d0e51c2d18: Pull complete
a59752bdcf5a: Pull complete
456a3213e1b1: Pull complete
7584aa22f9f7: Pull complete
422b6a98c439: Pull complete
1868e5257880: Pull complete
c19ed431de01: Pull complete
cd352db127f3: Pull complete
18ab8b9bc7a9: Pull complete
Digest: sha256:0f36a5002a200bb1649771c2cb9403ca5392d7ac4bb23f9ad500b339df3f5a3a
Status: Downloaded newer image for odoo:16
763c4af8ced17d801667da79c46430f885f782287c299a56a202452cd87cd495
```

Y si luego vamos a la pagina con localhost y el puerto que hemos abierto podremos comprobar si lo hemos hecho bien y se abre correctamente el servicio:

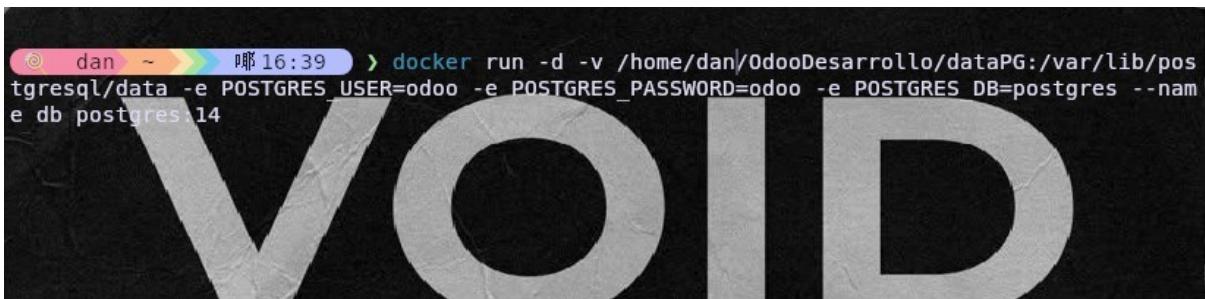


Como forma adicional de comprobar si va podremos parar el contenedor y recargar la página y así si lo hemos hecho bien no tendría que cargar la página tal que así:



## Odoo 15 en Docker: - Parte 2: Contenedor Odoo en desarrollo

Lanzamos Odoo en un contenedor preparado junto a postgresql (un tema importante que no he comentado en el comando anteriormente hecho es que tenemos que reemplazar en la ruta /usuario/ por nuestro usuario de ordenador en mi caso /dan/)

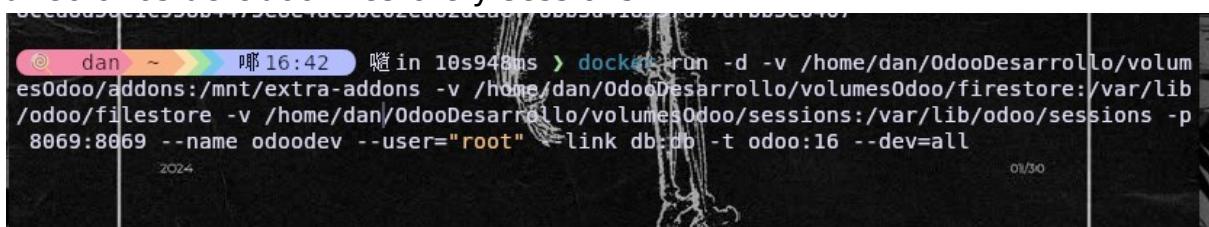




```
dan ~ 16:39 > docker run -d -v /home/dan/OdooDesarrollo/dataPG:/var/lib/postgresql/data -e POSTGRES_USER=odoo -e POSTGRES_PASSWORD=odoo -e POSTGRES_DB=postgres --name db postgres:14
Unable to find image 'postgres:14' locally
14: Pulling from library/postgres
0e4bc2bd6656: Pull complete
80a8c0ac4601: Pull complete
08927b2b9336: Pull complete
a9cbee5ac34d: Pull complete
5920f036e231: Pull complete
83dbba3aaab3: Pull complete
0d67bc96accb: Pull complete
ca269a9e240c: Pull complete
7c576d448336: Pull complete
3a909eca6304: Pull complete
da2a031a8efd: Pull complete
117e98dc7655: Pull complete
d0c34fc6d7b7: Pull complete
773a7fb7a56e: Pull complete
Digest: sha256:ca25035f7e6f74552655a1c5e4a9eb21f85e9d316f1f70371f790ef70095dd58
Status: Downloaded newer image for postgres:14
8cccd8d56c1c958b4475e8e4dc9bc62ed62dcae1f8bb3d418557d77dfbb3e6407
```

Ahora seguimos creando el contenedor de Odoo añadiendo extras como “-v”

/home/usuario/OdooDesarrollo/addons:/mnt/extra-addons” para que cargue los modulos del directorio después de los dos puntos como se lo indicamos de esta manera, en “...:var/lib/odoo/sessions” montaremos los contenedores y los volumenes aqui para tener persistencia de los directorios de Odoo firestore y sessions.



```
dan ~ 16:42 > docker run -d -v /home/dan/OdooDesarrollo/volumesOdoo/addons:/mnt/extra-addons -v /home/dan/OdooDesarrollo/volumesOdoo/firestore:/var/lib/odoo/filestore -v /home/dan/OdooDesarrollo/volumesOdoo/sessions:/var/lib/odoo/sessions -p 8069:8069 --name odoodev --user="root" -link db:db -t odoo:16 --dev=all
```



```
dan ~ 16:42 > docker run -d -v /home/dan/OdooDesarrollo/volumesOdoo/addons:/mnt/extra-addons -v /home/dan/OdooDesarrollo/volumesOdoo/firestore:/var/lib/odoo/filestore -v /home/dan/OdooDesarrollo/volumesOdoo/sessions:/var/lib/odoo/sessions -p 8069:8069 --name odoodev --user="root" -link db:db -t odoo:16 --dev=all
Unable to find image 'odoo:16' locally
16: Pulling from library/odoo
c2d0e51c2d18: Pull complete
a59752bccf5a: Pull complete
456a3213e1b1: Pull complete
422b6a98c439: Pull complete
18ab8b9bc7a9: Pull complete
7584aa22f9f7: Pull complete
c19ed431de01: Pull complete
cd352db127f3: Pull complete
1868e5257880: Pull complete
Digest: sha256:0f36a5002a200bb1649771c2cb9403ca5392d7ac4bb23f9ad500b339df3f5a3a
Status: Downloaded newer image for odoo:16
8338f20aca1a287c3fe614f178a4bc0a8a1f6a75982ecba5f730c95d46f8a697c
```

Es recomendable para ahorrarnos dolores de cabeza o rodeos darle permisos a la carpeta addons con el comando `sudo chmod -R 777 /home/usuario/volumenesOdoo/addons`, deberemos adaptarlo a nuestros directorios por ejemplo en mi caso sería: "`sudo chmod -R 777 /home/dan/OdooDesarrollo/volumesOdoo/addons/`". El primer comando de la imagen es el incorrecto y no adaptado a nuestro sistema de archivos, el segundo es el correcto en mi caso.

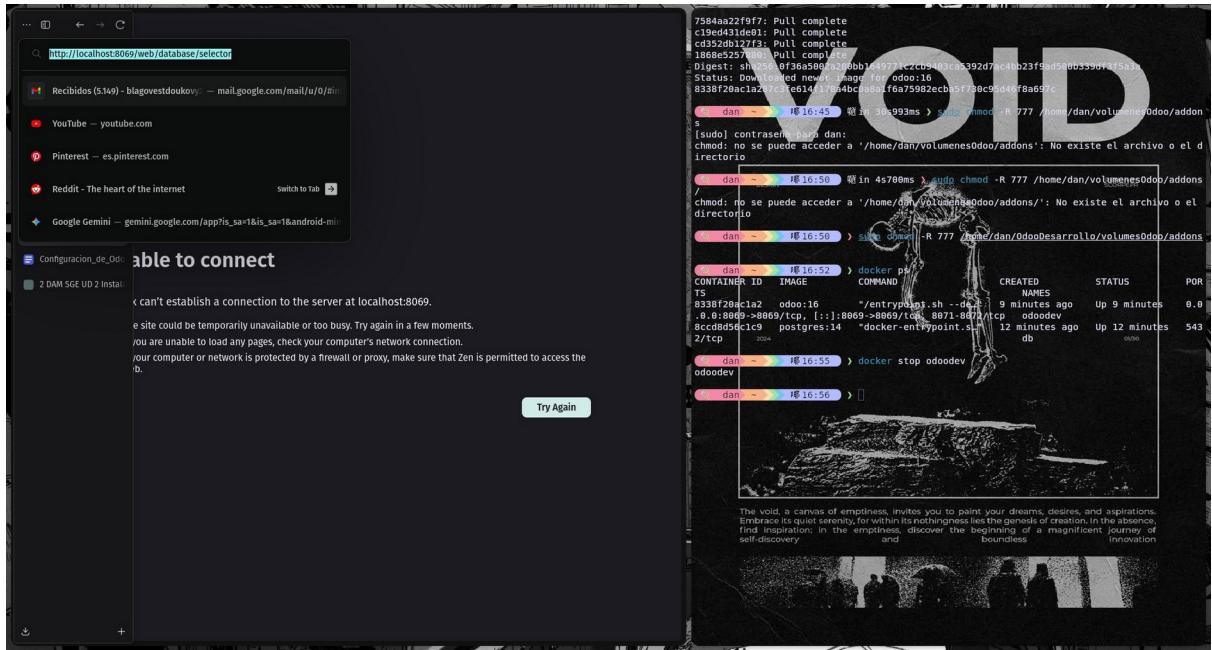
```

dan ~ 16:50 > sudo chmod -R 777 /home/dan/volumenesOdoo/addons
chmod: no se puede acceder a '/home/dan/volumenesOdoo/addons/': No existe el archivo o el directorio

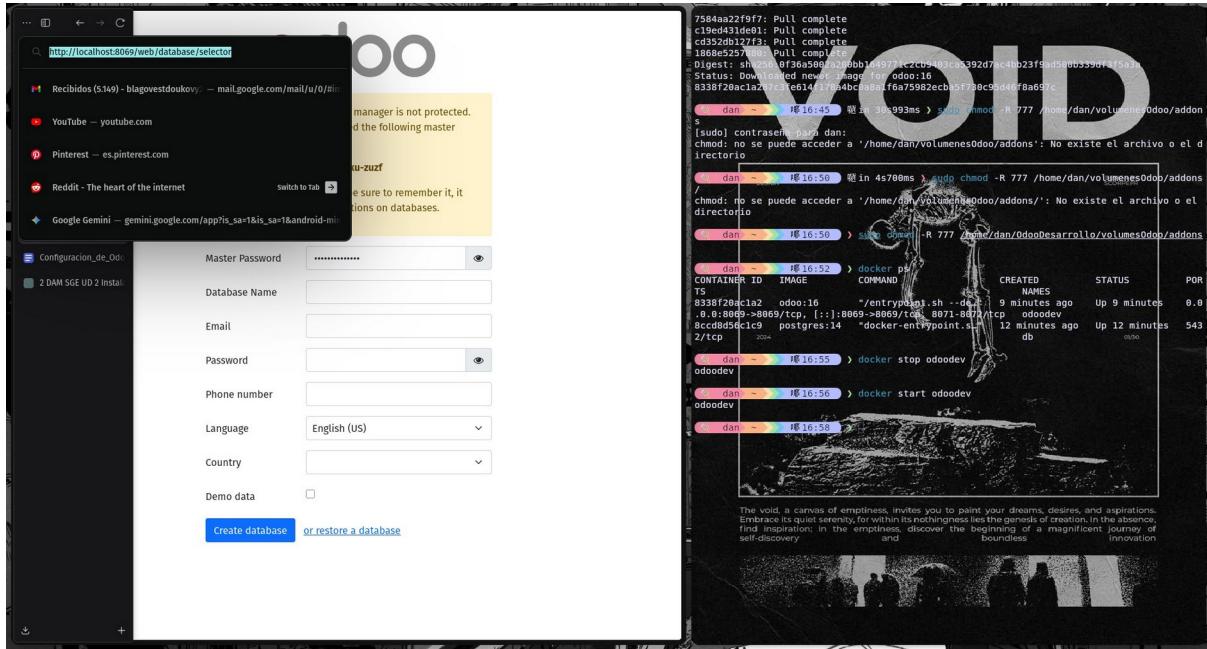
dan ~ 16:50 > sudo chmod -R 777 /home/dan/OdooDesarrollo/volumesOdoo/addons

```

Finalmente hacemos la comprobación de que funciona correctamente con un docker stop odoodev y recargando la página para ver si se para que en este caso ha sido exitoso



y luego hacemos un docker start odoodev para ver si se inicia de nuevo correctamente que si que lo hace



## Odoo 16 en Docker: - Parte 3: Docker Compose para Odoo

El primer paso para docker compose es la creación del archivo docker-compose.yml (tambien puede ser .yaml, es lo mismo) y poner en el contenido las especificaciones que queremos que tengan los contenedores, las imagenes... en este caso creamos db (con la información de la base de datos como un volumen para almacenar, contraseña, usuario y nombre de la base de datos y la imagen de la que queremos que haga pull que es en este caso postgres:14 (versión 14)) y web (con 3 volúmenes, especificando los puertos 8069:8069 aun que sea el predeterminado que pone odoo, la imagen que queremos que haga un pull que es odoo:16(versión 16)).

Así quedaría el archivo compose:

```

43  version: '3.3'
42
41  services:
40    #Definimos el servicio Web, en este caso Odoo
39    web:
38      #Indicamos que imagen de Docker Hub utilizaremos
37      image: odoo:16
36      #Indicamos que depende de "db", por lo cual debe ser procesada primero "db"
35      depends_on:
34        - db
33
32      # Port Mapping: indicamos que el puerto 8069 del contenedor se mapeara con el mismo puerto en el anfitrión
31      # Permitiendo acceder a Odoo mediante http://localhost:8069
30      ports:
29        - 8069:8069
28
27      # Mapeamos el directorio de los contenedores (como por ejemplo "/mnt/extra-addons" )
26      # en un directorio local (como por ejemplo en un directorio "./volumesOdoo/addons")
25      # situado en el lugar donde ejecutemos "Docker compose"
24      volumes:
23        - ./volumesOdoo/addons:/mnt/extra-addons
22        - ./volumesOdoo/odoo/filestore:/var/lib/odoo/filestore
21        - ./volumesOdoo/odoo/sessions:/var/lib/odoo/sessions
20
19      # Definimos variables de entorno de Odoo
18      environment:
17        - HOST=db
16        - USER=odoo
15        - PASSWORD=odoo
14      # Indica que pasa ese parametro al arrancar el servicio Odoo
13      command: --dev=all
12
11  #Definimos el servicio de la base de datos
10  db:
9    image: postgres:14
8    # Definimos variables de entorno de PostgreSQL
7    environment:
6      - POSTGRES_PASSWORD=odoo
5      - POSTGRES_USER=odoo
4      - POSTGRES_DB=postgres
3      # Mapeamos el directorio del contenedor "var/lib/postgresql/data" en un directorio "./volumesOdoo/dataPostgreSQL"
2      # situado en el lugar donde ejecutemos "Docker compose"
1      volumes:
44        - ./volumesOdoo/dataPostgreSQL:/var/lib/postgresql/data

```

Una vez hemos especificado todo lo necesario en el documento ahí le hemos dado las instrucciones para construir los contenedores y solo nos hace falta escribir por terminal “docker compose up -d” (la opción “up” se usa para poner en marcha los contenedores y si es la primera vez que lo hacemos los crea y “-d” la utilizamos para que la terminal no se bloquee) para ponerlo en marcha.

Primero veremos al ejecutar el comando y veremos una interfaz cli del progreso de creacion con compose:

```

15      - PASSWORD=odoo
14      # Indica que pasa ese parametro al arrancar el servicio Odoo
13      command: --dev=all

1: docker compose up -d
dan ~/Escritorio/composeEmpresa 15:56 > docker compose up -d
WARN[0000] /home/dan/Escritorio/composeEmpresa/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 13/24
.. web [██████] Pulling
.. db [██████] Pulling
4.6s
4.6s

```

Y terminado esto podemos ver que ha terminado la creación y el start de los contenedores:

```

1: dan@pop-os:~/Escritorio/composeEmpresa
dan ~/Escritorio/composeEmpresa 15:56 docker compose up -d
WARN[0000] /home/dan/Escritorio/composeEmpresa/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 24/24
  ✓ web Pulled
  ✓ db Pulled
254.1s
128.8s

[+] Running 3/3
  ✓ Network composeempresa_default Created
  ✓ Container composeempresa-db-1 Started
  ✓ Container composeempresa-web-1 Started
0.0s
0.3s
0.4s

```

Las primeras no funciona solo con esto por que no tiene permisos y no se los hemos concedido así que no hace satisfactoriamente la creación y puesta en marcha y es recomendable hacerlo con “sudo chmod -R 777 ./addons” que en mi caso es “sudo chmod -R 777 ./volumesOdoo”.

```

dan ~/Escritorio/composeEmpresa 16:09 sudo chmod -R 777 ./volumesOdoo
dan ~/Escritorio/composeEmpresa 16:09
TERMINAL $ term:/..bin/zsh
i < 跳10 Bot 33:55 16:09

```

Y ahora si que funcionará correctamente(si no nos saldría una página con error como la primera imagen):

## Internal Server Error

The server encountered an internal error and was unable to complete your request. Either the server is overloaded or there is an error in the application.

