

# **Tarea 2.4 – Instalación de Odoo ERP en Docker**

Nombre y apellidos: Cristina Sandoval  
Laborde

Curso: 2ºDAM

Asigantura: Sistemas de gestión  
empresarial

# Índice

1.Introducción .....	1
2.Pasos .....	1
Paso 1. Descarga e instalación de Docker .....	1
Paso 2. Instalación y comprobación de Docker Compose en Windows 11 .....	2
Paso 3. Instalación y configuración de Portainer CE en Windows 11 .....	6
Paso 4. Creación del contenedor de Odoo y PostgreSQL con Docker Compose.....	12
3.Problemas y soluciones .....	15
4. Conclusiones personales.....	15

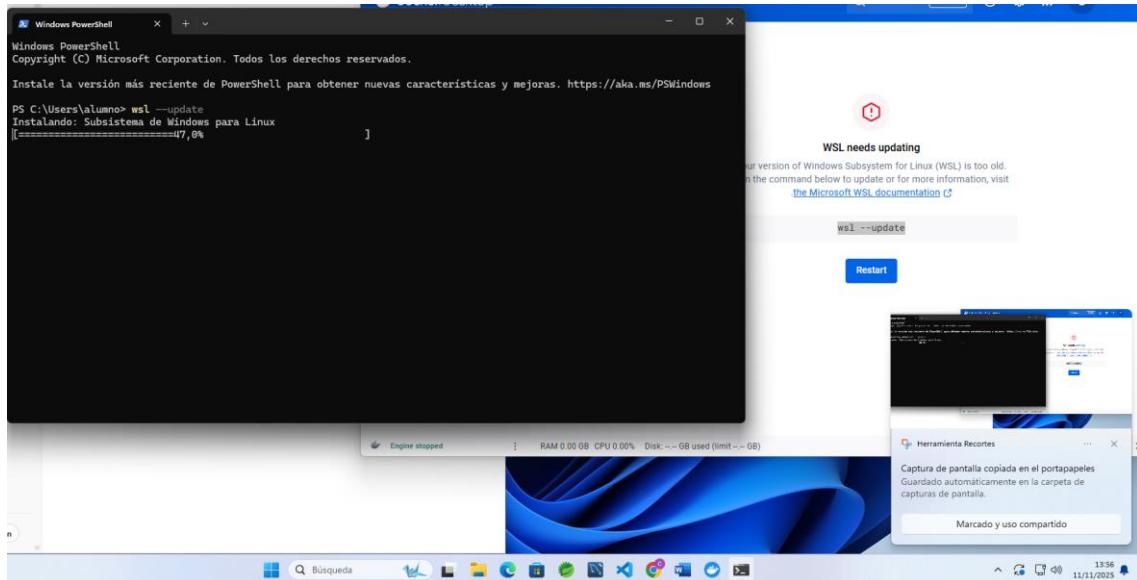
## 1. Introducción

Docker es una plataforma que permite crear y ejecutar aplicaciones dentro de contenedores, que son entornos aislados donde cada servicio tiene todo lo necesario para funcionar sin depender del sistema operativo principal. Estos contenedores se despliegan muy rápido, ocupan menos recursos que las máquinas virtuales y facilitan que una aplicación funcione igual en cualquier ordenador o servidor. En entornos empresariales, como el despliegue de Odoo, Docker resulta especialmente útil porque permite levantar varios servicios (Odoo + PostgreSQL) de forma coordinada, reproducible y sin complicadas instalaciones manuales. Gracias a esto, administrar aplicaciones es más sencillo, seguro y eficiente.

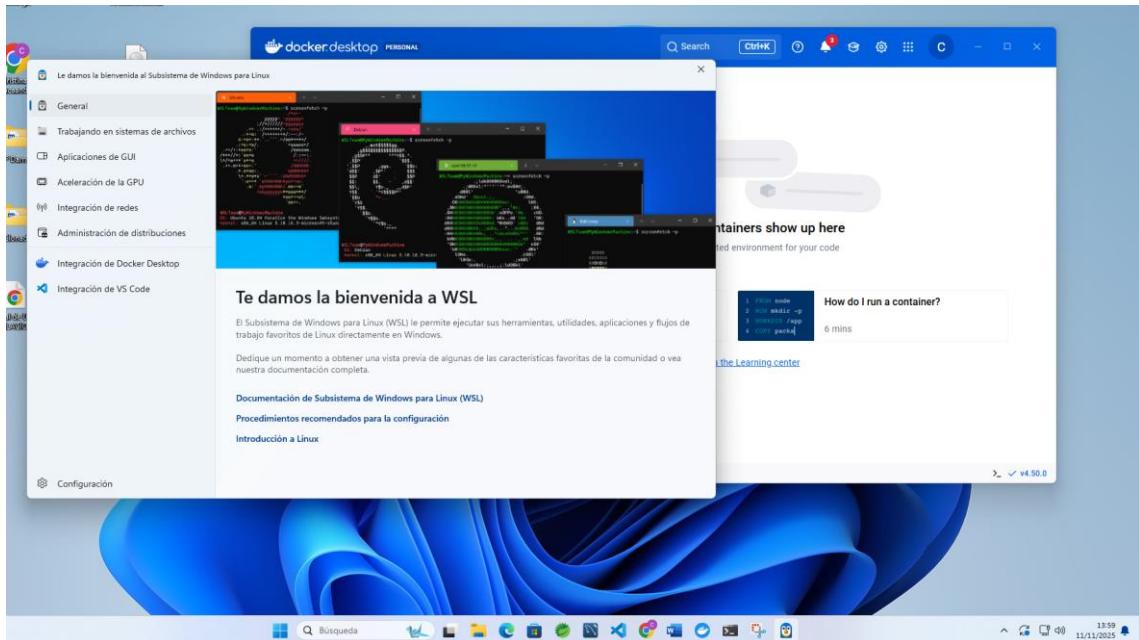
## 2. Pasos

### Paso 1. Descarga e instalación de Docker

Ya tenía instalado Docker Desktop pero una versión antigua entonces he tenido que instalar la actualización y ejecutar el siguiente comando en la PowerShell `wsl --update`



Y ya me deja entrar sin errores



Sigo desde el punto 4. Comprobación del funcionamiento

Compruebo que Docker Desktop está instalado correctamente y la versión que es.

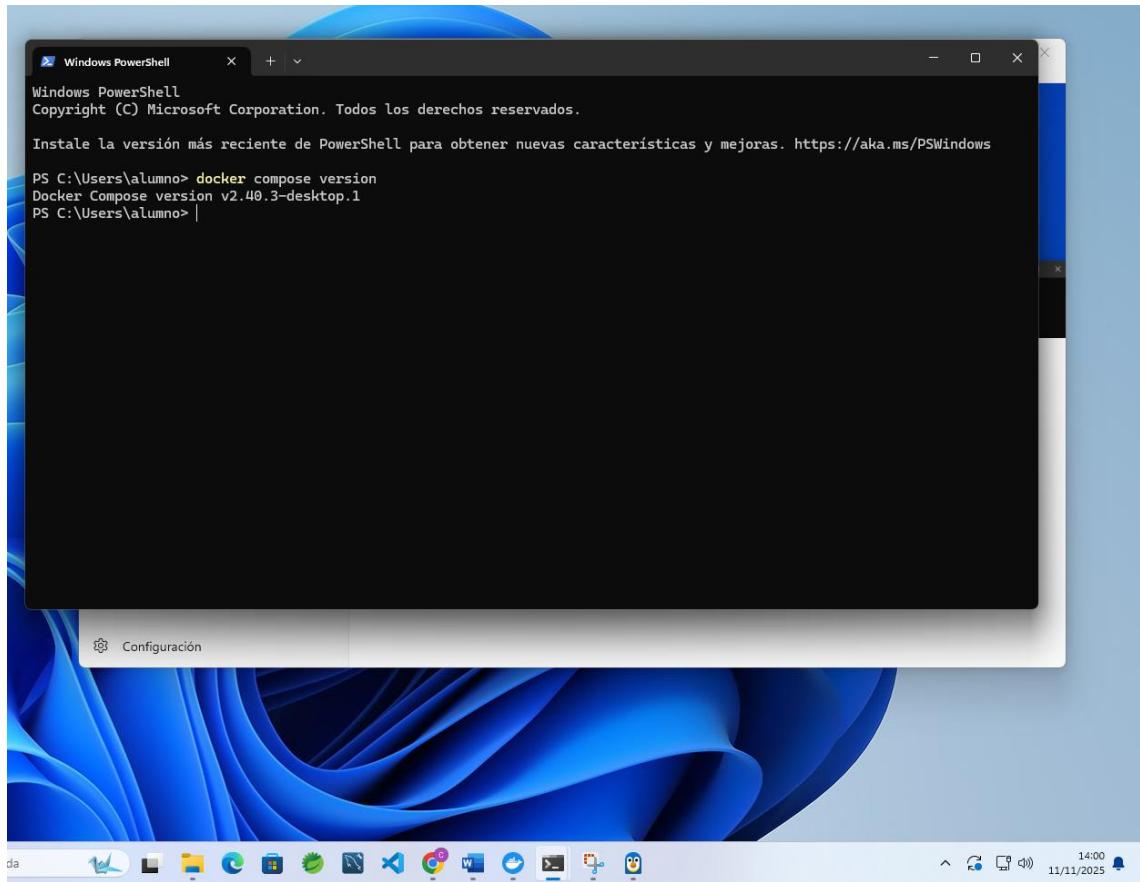
```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindo

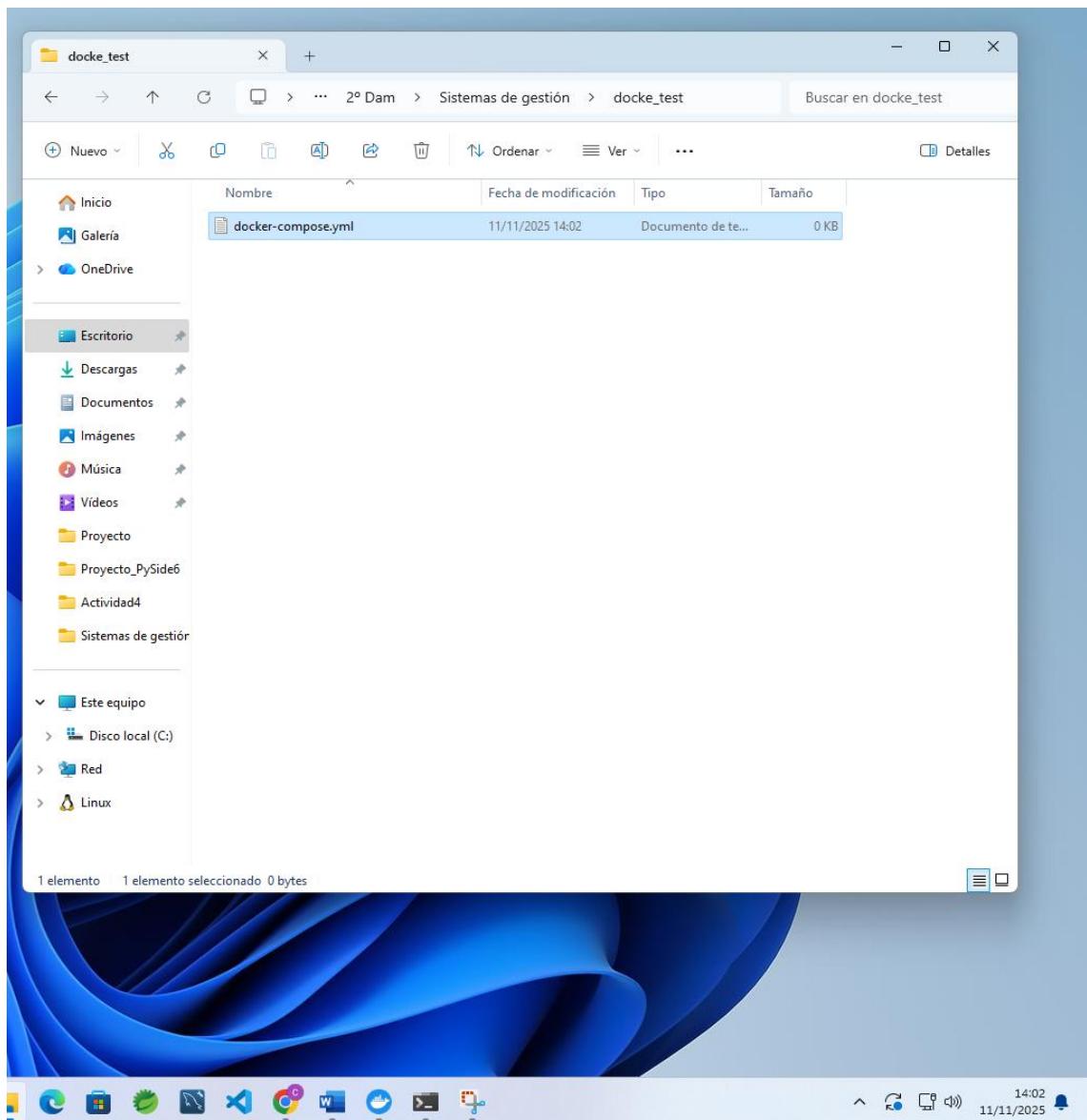
PS C:\Users\alumno> docker --version
Docker version 28.5.1, build e180ab8
PS C:\Users\alumno> |
```

## Paso 2. Instalación y comprobación de Docker Compose en Windows 11

Ahora compruebo que Docker Compose esta correctamente instalado.



Ahora comprobamos que Docker Compose funciona correctamente, para ello creamos una carpeta que se llame “Docker\_test” y dentro de ella creamos un documento llamado “Docker-compose” y le ponemos la extensión. Yml



Ahora abrimos el archivo con visual Studio Code y le ponemos el siguiente contenido:

services:

web:

    image: nginx

    ports:

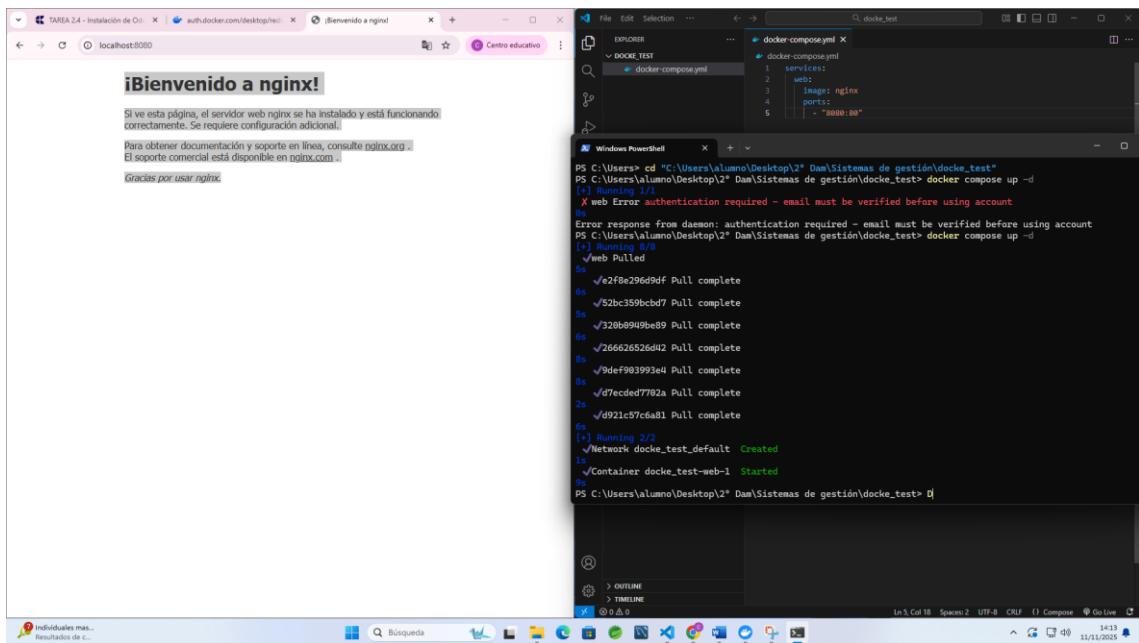
        - "8080:80"

The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left has a 'DOCKE\_TEST' folder expanded, containing a 'docker-compose.yml' file. The main editor area displays the following YAML configuration:

```
1 services:
2   web:
3     image: nginx
4     ports:
5       - "8080:80"
```

The status bar at the bottom shows the file is at line 5, column 18, with settings like 'Spaces: 2', 'UTF-8', 'CRLF', and 'Compose'. The bottom right corner shows the date and time as '11/11/2025 14:05'.

Ahora en la PowerShell vamos a la carpeta donde está el archivo y ponemos el siguiente comando “Docker compose up -d”, esto descargará la imagen de Nginx y levanta el servidor principal de Nginx, si en el navegador vamos a la <http://localhost:8080> vemos la página de bienvenida de Nginx, lo que demuestra que todo ha ido bien

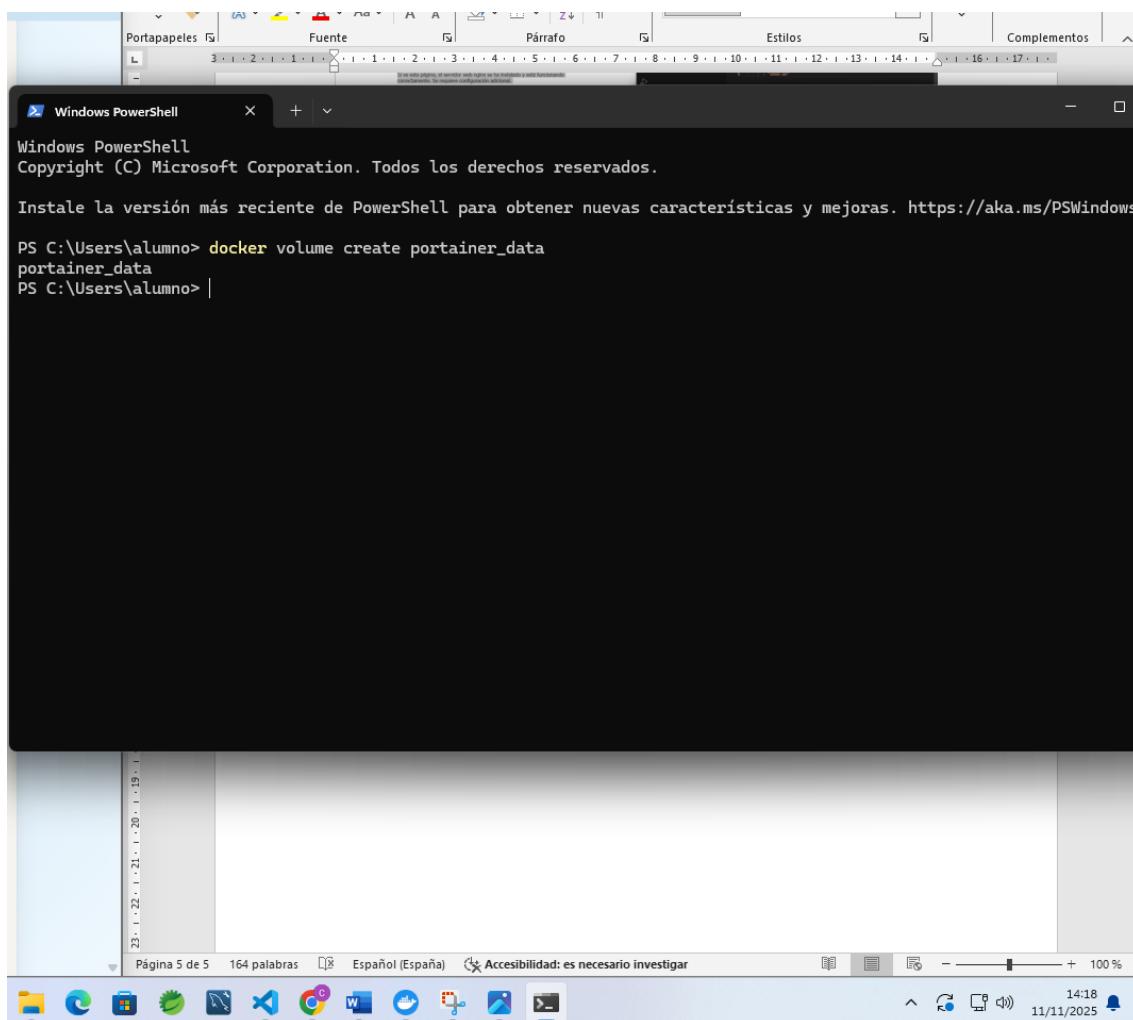


Ahora detenemos y eliminamos el contenedor con el siguiente código “Docker compose down”

```
PS C:\Users\alumno\Desktop\2º Dam\Sistemas de gestión\docke_test> docker compose down
[+] Running 2/2
  ✓ Container docke_test-web-1  Removed
  ✓ Network docke_test_default  Removed
PS C:\Users\alumno\Desktop\2º Dam\Sistemas de gestión\docke_test> |
```

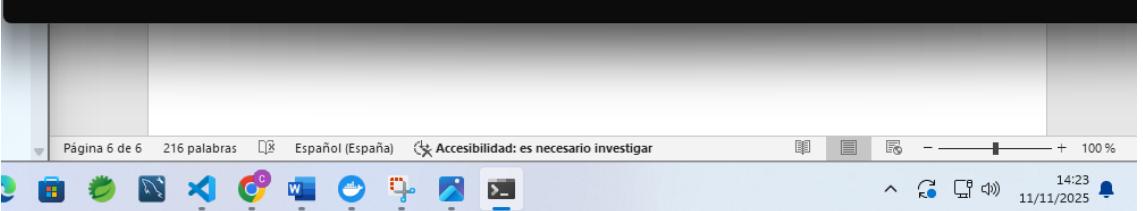
### Paso 3. Instalación y configuración de Portainer CE en Windows 11

empezamos creando un volumen persistente con el siguiente código “docker volume create portainer\_data” y comprobamos que nos funciona porque nos dale portainer\_data



Ahora ponemos el siguiente comando en la PowerShell docker run -d -p 9000:9000 -p 9443:9443 --name portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer\_data:/data portainer/portainer-ce:latest

```
PS C:\Users\alumno> docker run -d -p 9000:9000 -p 9443:9443 --name portainer --restart=always -v /var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest
Unable to find image 'portainer/portainer-ce:latest' locally
latest: Pulling from portainer/portainer-ce
c551c6af3241: Pull complete
f3e139b9cd66: Pull complete
22d4f45bf667: Pull complete
089b96337e3d: Pull complete
2d896f38a01e: Pull complete
d35c85f36345: Pull complete
9f7772ceb88a: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:d38a6876b61df32e4da13c0ca61cf2aa0f27afc103abb4ca7ad4e1cf000e17c3
Status: Downloaded newer image for portainer/portainer-ce:latest
b33c9f71e65ed1a5374eb898255cf5aa5e201307b1cf8561e686bc317c536304
PS C:\Users\alumno> |
```

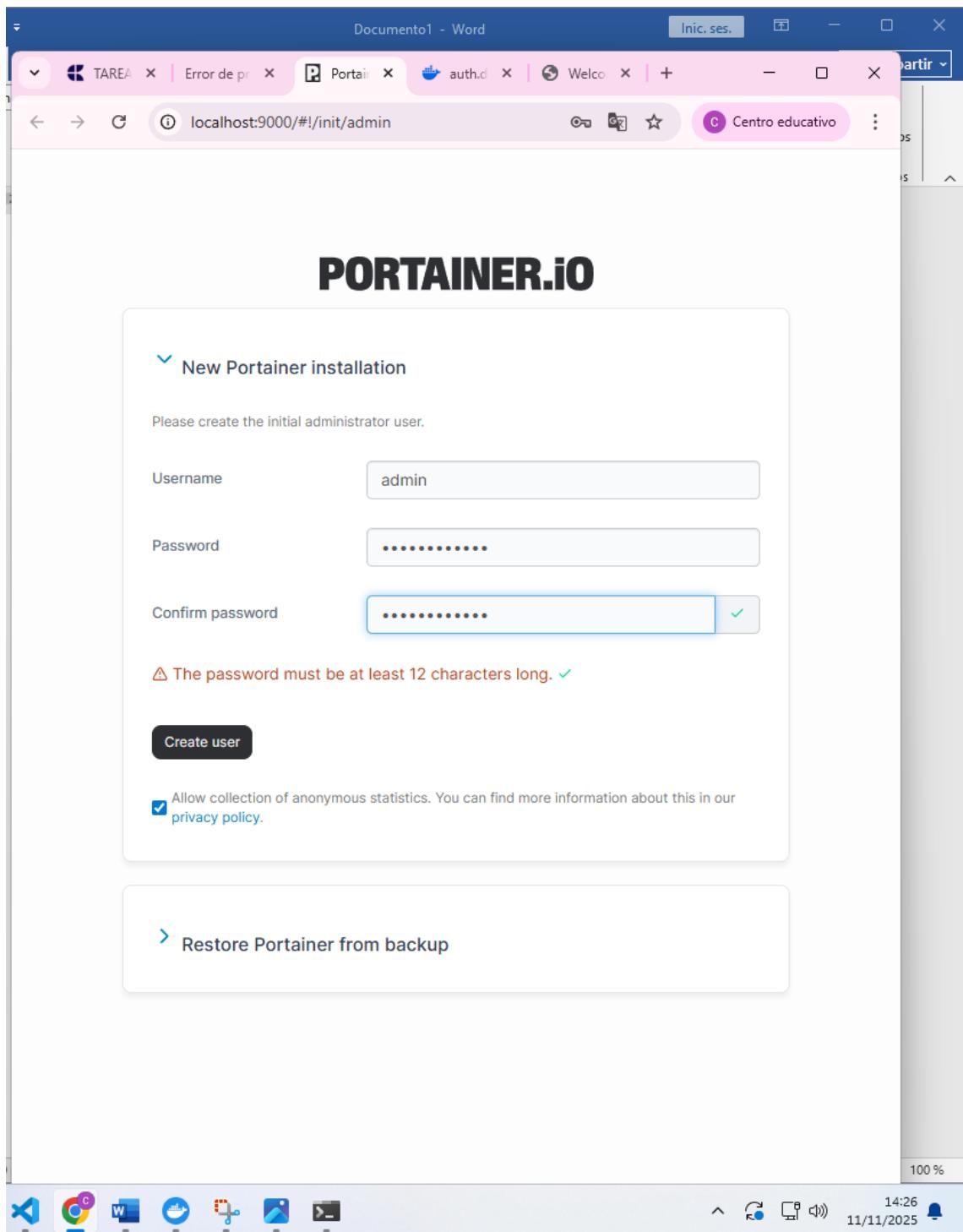


Al terminar la descarga, escribimos “Docker ps” y vemos lo que ocurre

```
PS C:\Users\alumno> docker run -d -p 9000:9000 -p 9443:9443 --name portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest
Unable to find image 'portainer/portainer-ce:latest' locally
latest: Pulling from portainer/portainer-ce
c551c6af3241: Pull complete
f3e139b9cd66: Pull complete
22d4f45bf667: Pull complete
089b96337e3d: Pull complete
2d896f38a01e: Pull complete
d35c85f36345: Pull complete
9f7772ceb88a: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:d38a6876b61df32e4da13c0ca61cf2aa0f27afc103abb4ca7ad4e1cf000e17c3
Status: Downloaded newer image for portainer/portainer-ce:latest
b33c9f71e65ed1a5374eb898255cf5aa5e201307b1cf8561e686bc317c536304
PS C:\Users\alumno> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
b33c9f71e65e        portainer/portainer-ce:latest   "/portainer"        About a minute ago   Up About a minute   0.0.0.0:9000->9000
0/tcp, [::]:9000->9000/tcp, 0.0.0.0:9443->9443/tcp, [::]:9443->9443/tcp   portainer
PS C:\Users\alumno> |
```



Según las instrucciones, todo ha salido bien.



Le damos a Create user y en la siguiente pantalla seleccionamos “Get Started”,

The screenshot shows the Portainer interface with the title "Environment Wizard" at the top. On the left, there's a sidebar with a dark background and a white "P" logo. Below the logo are several icons: a house (Dashboard), a person (Users), a folder (Images), a double arrow (Networks), a document (Logs), a file (Containers), a bell (Events), and a gear (Settings). The main content area has a light gray background. At the top of the main area, it says "Quick Setup" with a circular refresh icon. Below that is a section titled "Environment Wizard" with a gear icon. The main message reads: "Welcome to Portainer. We have connected your local environment of Docker to Portainer. Get started below with your local portainer or connect more container environments." There are two main buttons: "Get Started" on the left and "Add Environments" on the right. The "Get Started" button features a Docker ship icon and the text "Proceed using the local environment which Portainer is running in". The "Add Environments" button features a plug icon and the text "Connect to other environments".

Ahora podemos ver Containers, Images, NetWorks y Volumes.

The screenshot shows the Portainer interface. On the left is a vertical sidebar with icons for Home, Docker, Container, Stack, Image, Volume, Network, and Log. The main area has a header "Environment summary" and "Dashboard". It displays the following information:

- Environment info**: Environment is "local - Standalone 28.5.1", URL is "/var/run/docker.sock", and Tags are "-".
- Stack**: 1 Stack.
- Containers**: 2 Containers. Status: 2 running, 0 stopped, 0 healthy, 0 unhealthy.
- Images**: 2 Images. Total size: 472.6 MB.
- Volume**: 1 Volume.
- Networks**: 4 Networks.

Si accedemos a <https://localhost:9443> e iniciamos sesión, entramos en Portainer

The screenshot shows the Portainer web interface. At the top, there are two tabs: 'Portainer | local' and 'Portainer'. The main content area is titled 'Home'. A 'Latest News From Portainer' box is present, stating: 'Portainer 2.33.0 LTS is now available! This release adds all the new features from the previous LTS releases (including a brand new look, Helm, Edge updating and mTLS improvements, and a new experimental alerting feature) to our production-ready LTS release stream. [Learn more in our documentation.](#)' Below this, the 'Environments' section is shown with a summary for the 'local' environment:

local	<span style="color: green;">Up</span>	2025-11-14 09:40:30	Standalone 28.5.1	/var/run/docker.sock
Group: Unassigned	No tags	Local		
1 stack	2 containers	2	0	0
2 images	16 CPU	16.4 GB RAM		
			Live connect	Disconnected

El comando “Docker stop portainer” detiene el Portainer

```

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

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/psv

PS C:\Users\alumno> Docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS      NAMES
cdfdbc0b70cb   nginx          "/docker-entrypoint..."   11 minutes ago   Up 11 minutes   docke_test-web-1
b33c9f71e65e   portainer/portainer-ce:latest   "/portainer"    2 days ago     Up 21 minutes   0
000/tcp, [::]:9000->9000/tcp, 0.0.0.0:9443->9443/tcp, [::]:9443->9443/tcp   portainer
PS C:\Users\alumno> docker restart portainer
portainer
PS C:\Users\alumno> docker stop portainer
portainer
PS C:\Users\alumno>

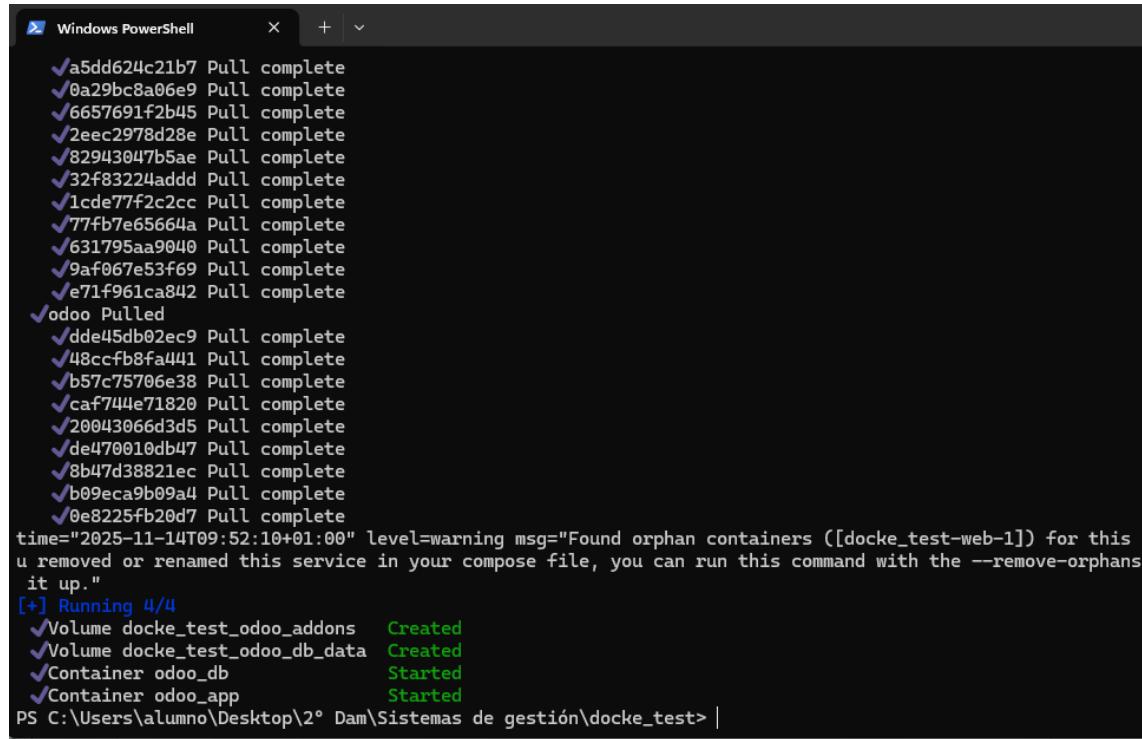
```

#### Paso 4. Creación del contenedor de Odoo y PostgreSQL con Docker Compose

En la carpeta que creamos anteriormente esta el documento Docker-compose.yml que también creamos antes, entonces nos vamos a visual studio code y reemplazamos el código que tuvimos por el código nuevo.

Después abrimos la PowerShell, naveganos hasta la carpeta donde esta el archivo Docker-compose.yml y ejecutamos el siguiente código (Docker compose up -d)

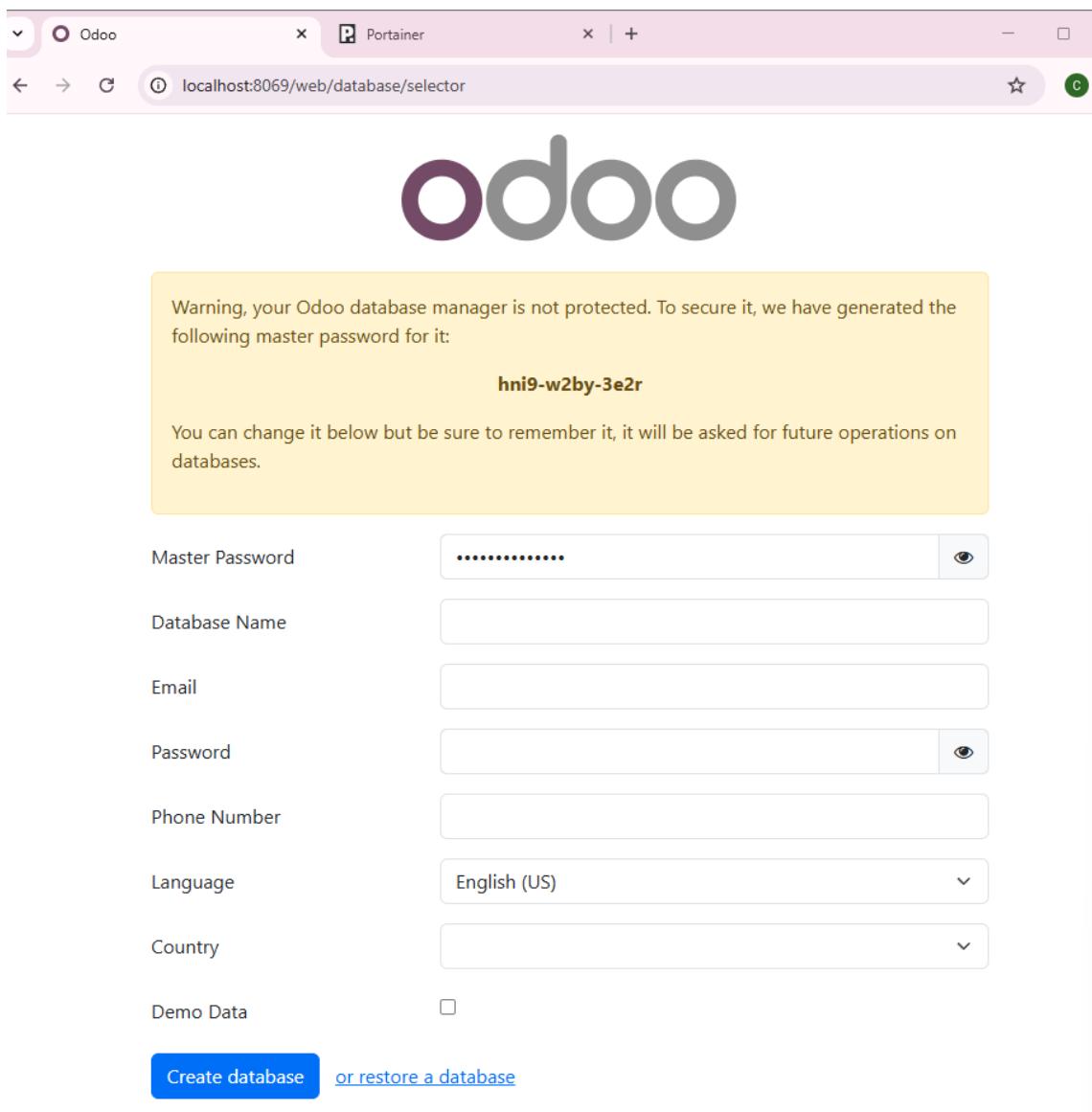
Hay que esperar un par de minutos para que se complete



```
Windows PowerShell

✓ a5dd624c21b7 Pull complete
✓ 0a29bc8a06e9 Pull complete
✓ 6657691f2b45 Pull complete
✓ 2eec2978d28e Pull complete
✓ 82943047b5ae Pull complete
✓ 32f83224addd Pull complete
✓ 1cde77f2c2cc Pull complete
✓ 77fb7e65664a Pull complete
✓ 631795aa9040 Pull complete
✓ 9af067e53f69 Pull complete
✓ e71f961ca842 Pull complete
✓ odoored
✓ dde45db02ec9 Pull complete
✓ 48ccfb8fa441 Pull complete
✓ b57c75706e38 Pull complete
✓ caf744e71820 Pull complete
✓ 20043966d3d5 Pull complete
✓ de470010db47 Pull complete
✓ 8b47d38821ec Pull complete
✓ b09eca9b09a4 Pull complete
✓ 0e8225fb20d7 Pull complete
time="2025-11-14T09:52:10+01:00" level=warning msg="Found orphan containers ([docke_test-web-1]) for this project. You removed or renamed this service in your compose file, you can run this command with the --remove-orphans option to remove it up."
[+] Running 4/4
✓ Volume docke_test_odoo_addons Created
✓ Volume docke_test_odoo_db_data Created
✓ Container odoo_db Started
✓ Container odoo_app Started
PS C:\Users\alumno\Desktop\2º Dam\Sistemas de gestión\docke_test>
```

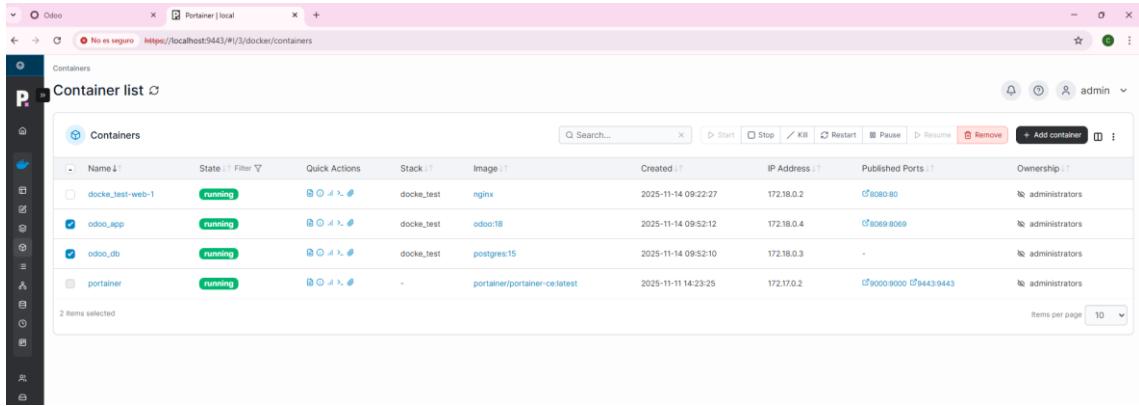
Comprobamos que todo ha ido bien porque si ponemos localhost:8069 en el buscador se nos abre la página de Odoo.



Si arrancamos de nuevo el portainer con el siguiente comando “docker restart portainer” y entramos en <https://localhost:9443> y entramos en Containers, vemos los siguientes contenedores:

Odoo\_app

Odoo\_db



Por último, pausamos el entorno sin eliminarlo

```
PS C:\Users\alumno\Desktop\2º Dam\Sistemas de gestión\docke_test> docker compose stop
time="2025-11-14T09:59:10+01:00" level=warning msg="C:\\Users\\alumno\\Desktop\\2º Dam\\Sistemas de gestión\\docke_test\\docke-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Stopping 2/2
  ✓ Container odoo_app Stopped
  ✓ Container odoo_db Stopped
0.8s
0.2s
PS C:\Users\alumno\Desktop\2º Dam\Sistemas de gestión\docke_test> |
```

### 3. Problemas y soluciones

- Versión antigua de Docker Desktop** → Al abrirlo daba error. Lo solucioné actualizando WSL con wsl --update y reinstalando Docker Desktop.
- Descargas lentas de imágenes** → Algunas imágenes tardaron más de lo esperado, pero se resolvió simplemente esperando a que Docker completara los procesos.
- Certificados y acceso HTTPS en Portainer** → Al entrar por <https://localhost:9443> aparecía un aviso de seguridad del navegador, lo que es normal porque usa un certificado auto-firmado. Solo tuve que permitir la excepción.
- Tiempo de arranque de Odoo** → Tarda más que otros servicios (nginx por ejemplo). Esperé unos minutos y luego ya cargó correctamente.

Todos los problemas se resolvieron sin necesidad de reinstalar nada.

### 4. Conclusiones personales

Este trabajo me ha servido para entender cómo Docker simplifica el despliegue de aplicaciones complejas como Odoo, evitando instalaciones manuales largas y errores de configuración. Me ha parecido muy útil comprobar cómo cada servicio se ejecuta en su propio contenedor y cómo Docker Compose permite gestionarlos todos juntos. Lo más sencillo ha sido levantar Odoo una vez configurado el archivo yaml, y lo más difícil fue entender bien los puertos y las rutas de los volúmenes. Creo que Docker es una herramienta muy aplicable en otros módulos, especialmente en servidores, bases de

datos o cuando se necesiten entornos rápidos de prueba. Sin duda, es una tecnología muy útil para el trabajo real en empresas.