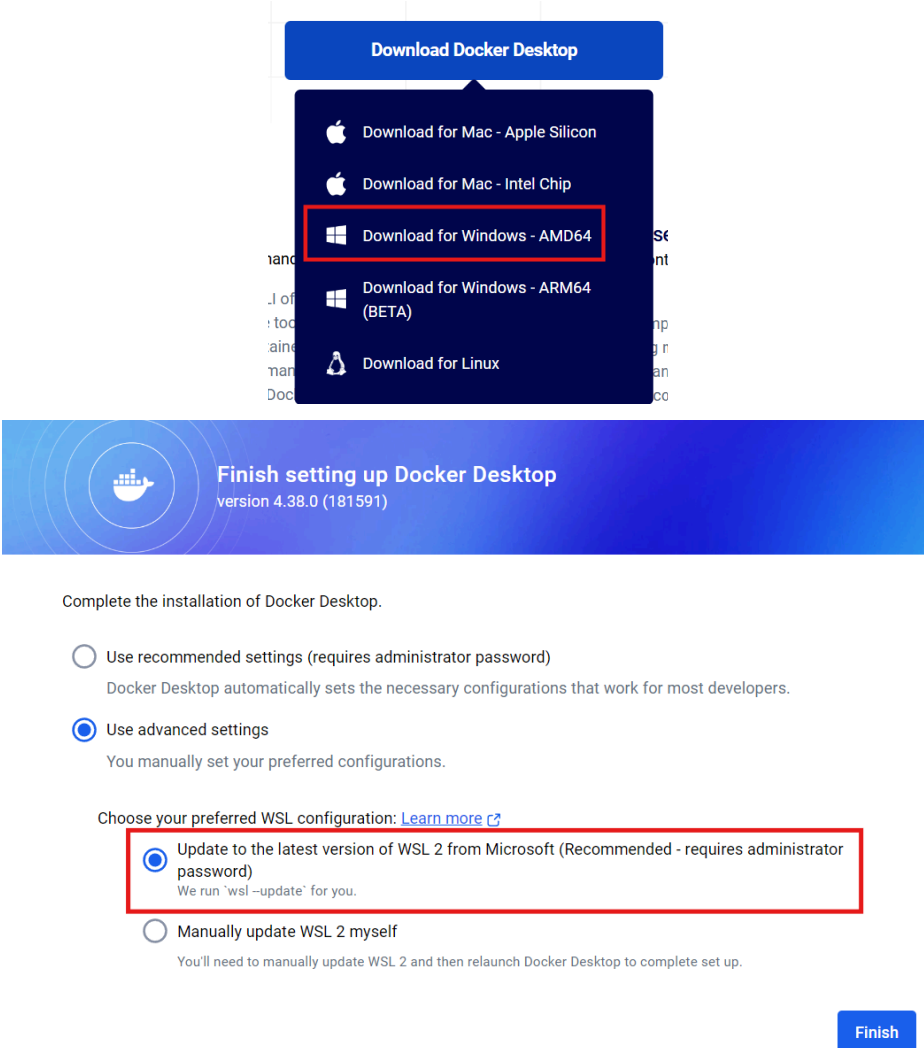


PRIMEROS PASOS EN DOCKER

Parte 1 - Preparación del entorno y primer contenedor

1. Descarga e instala Docker Desktop en tu sistema operativo, añadiendo una captura en la sección en la cual escogemos entre WSL e Hyper-V.



Download Docker Desktop

- Download for Mac - Apple Silicon
- Download for Mac - Intel Chip
- Download for Windows - AMD64
- Download for Windows - ARM64 (BETA)
- Download for Linux

Finish setting up Docker Desktop
version 4.38.0 (181591)

Complete the installation of Docker Desktop.

☐ Use recommended settings (requires administrator password)
Docker Desktop automatically sets the necessary configurations that work for most developers.

☒ Use advanced settings
You manually set your preferred configurations.

Choose your preferred WSL configuration: [Learn more](#)

☒ Update to the latest version of WSL 2 from Microsoft (Recommended - requires administrator password)
We run 'wsl --update' for you.

☐ Manually update WSL 2 myself
You'll need to manually update WSL 2 and then relaunch Docker Desktop to complete set up.

Finish

He elegido la opción de WSL porque Docker se encarga de actualizar WSL 2 por mí, sin que tenga que hacerlo manualmente. Así evito los posibles errores y me aseguro de que todo funcione con la versión más reciente. Además, es la opción recomendada, por lo que es la más segura para evitar problemas más adelante.

2. Una vez hayas iniciado sesión y estés en el panel de control, abre un terminal (cmd o PowerShell) y crea y ejecuta un nuevo contenedor con la imagen 'hello-world'.

Primero hay que iniciar sesión en docker con el comando: **docker login**

```
PS C:\Users\sofia> docker login
Authenticating with existing credentials...
Stored credentials invalid or expired

USING WEB-BASED LOGIN
To sign in with credentials on the command line, use 'dock

Your one-time device confirmation code is: LMNL-SNPQ
Press ENTER to open your browser or submit your device cod

Waiting for authentication in the browser...

Login Succeeded
```

Después se podrá ejecutar un nuevo contener con el comando: **docker run hello-world**

```
PS C:\Users\sofia> docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:7e1a4e2d11e2ac7a8c3f768d4166c2defeb09d2a750b010412b6ea13de1efb19
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

PS C:\Users\sofia>
```

3. Muestra en el terminal con el comando correspondiente la lista de contenedores en ejecución. ¿Por qué no aparece el que acabas de crear y ejecutar en el punto 2? ¿Qué parámetro en el comando nos permitirá verlo?

Para ver la lista de contenedores en ejecución, hay que utilizar el comando: **docker ps**

```
PS C:\Users\sofia> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
PS C:\Users\sofia>
```

No aparece el que se ha creado en el paso anterior, porque solo aparecen los contenedores que están actualmente en ejecución. Para ver todos los contenedores, hay que utilizar: **docker ps -a**

```
PS C:\Users\sofia> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
ca96097375ea  hello-world "/hello"               9 minutes ago  Exited (0) 5 minutes ago          keen_turing
PS C:\Users\sofia>
```

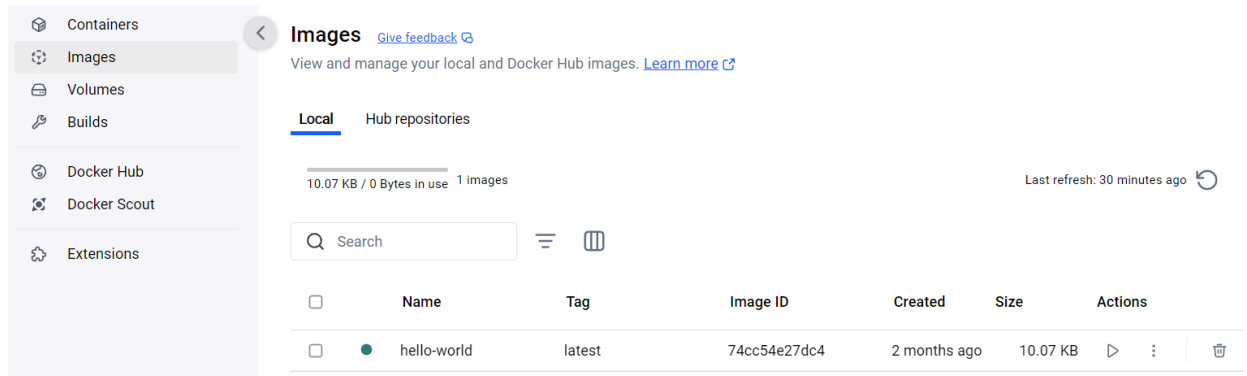
El parámetro -a, permite ver todos los contenedores, no solo los que están en ejecución.

4. Muestra en el panel de control de la interfaz gráfica el contenedor y la imagen creados en el punto 2. ¿Qué nombre tiene el contenedor?

En la sección ‘Containers’ aparecerá el contenedor hello-world que se ha creado en el paso 2.

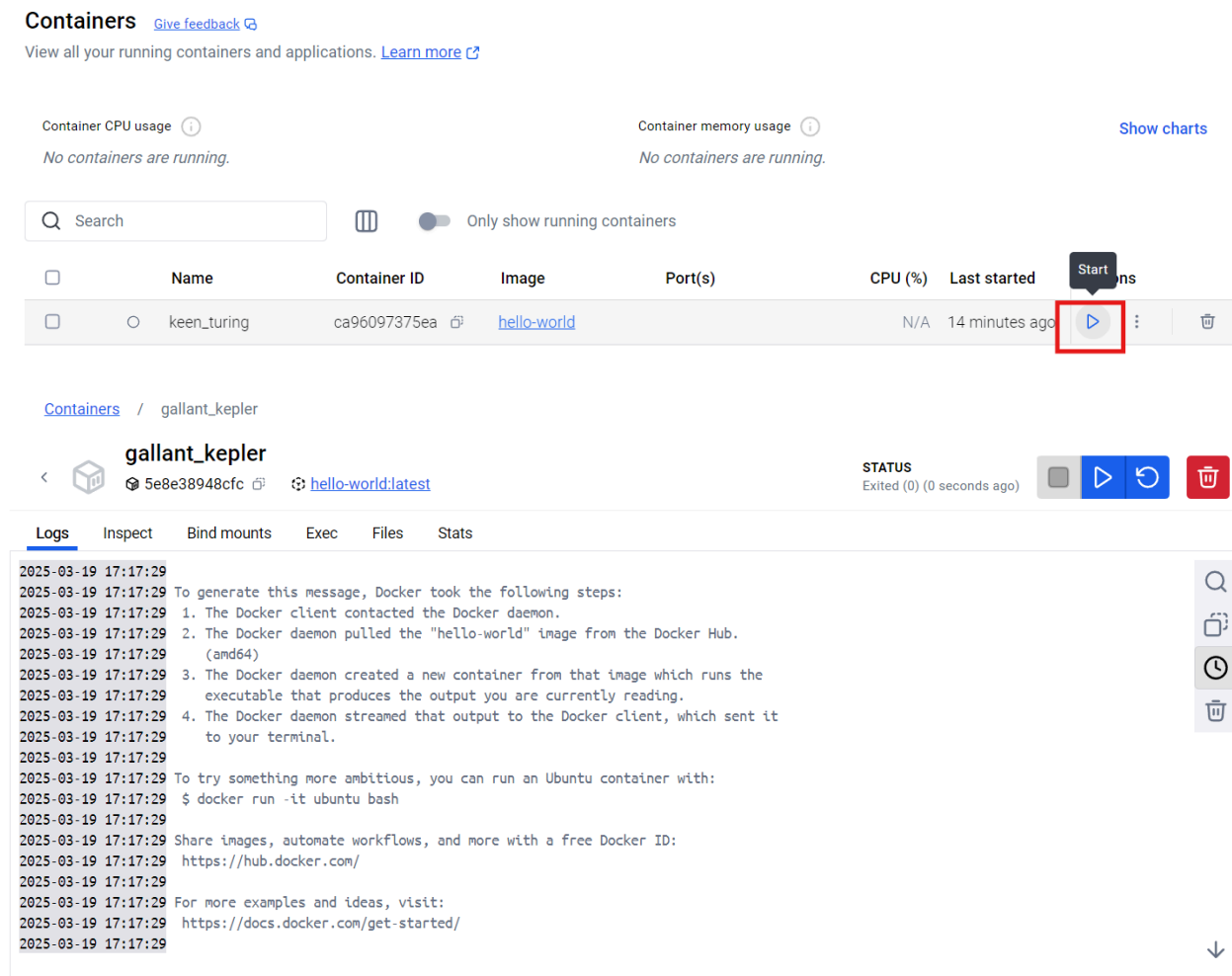
	Name	Container ID	Image	Port(s)	CPU (%)	Last state	Actions
<input type="checkbox"/>	keen_turing	ca96097375ea	hello-world		N/A	1 second	

Y en la sección ‘Images’ aparecerá la imagen creada con el nombre de ‘hello-world’.



5. Arranca el contenedor desde la interfaz gráfica y muestra desde la misma interfaz el resultado obtenido.

Primero hay que darle al botón de start:



Parte 2 - Despliegue de un servidor web con apache

1. Crea un contenedor con las siguientes características:

- Tipo demonio
- Utiliza la imagen httpd
- El nombre del contenedor es 'tunombre-apachewebserver'
- Permitirá el acceso a su contenido desde el puerto 3000 del host

Para crear el contenedor con las características que se han pedido, hay que utilizar el siguiente comando: **docker run -d --name sofia-apachewebserver -p 3000:80 httpd**

```
PS C:\Users\sofia> docker run -d --name sofia-apachewebserver -p 3000:80 httpd
1893341df63784769b6b12f6d0f81d5fe94d83a28ad3e202e1d50387a072a1ad
PS C:\Users\sofia>
```

Containers [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Container CPU usage ⓘ

0.01% / 1600% (16 CPUs available)

Container memory usage ⓘ

30.12MB / 7.26GB

[Show charts](#)

Search



Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	keen_turing	ca96097375ea	hello-world		0%	8 minutes ago	
<input type="checkbox"/>	sofia-apachewebserve	1893341df637	httpd	3000:80	0.01%	1 minute ago	

Explicación del comando por parámetros:

docker run: es el comando para crear y ejecutar un contenedor.

-d: Ejecuta el contenedor en modo desaparecido (demonio). Se ejecuta en segundo plano.

--name sofia-apachewebserver: nombre del contenedor.

-p 3000:80: Mapea el puerto 3000 del host al puerto 80 del contenedor. Esto permite el acceso al servidor web de Apache desde el navegador a través del puerto 3000 de la máquina local.

httpd: Especifica que se utilizará la imagen httpd (Apache HTTP Server).

2. ¿Dónde se encuentra el archivo index.html? Indica su ruta absoluta y editalo para que contenga el siguiente texto: 'Bienvenido/a a mi primera web desplegada con Docker'.

Para acceder al contenedor, hay que utilizar el comando:

docker exec -it sofia-apachewebserver bash.

```
PS C:\Users\sofia> docker exec -it sofia-apachewebserver bash
root@1893341df637:/usr/local/apache2# /usr/local/apache2/htdocs/index.html
bash: /usr/local/apache2/htdocs/index.html: Permission denied
root@1893341df637:/usr/local/apache2# echo "Bienvenido/a a mi primera web desplegada con Docker" > /usr/local/apache2/htdocs/index.html
root@1893341df637:/usr/local/apache2#
```

Una vez dentro, hay que buscar el directorio donde se encuentra el archivo index.html, con el comando: **/usr/local/apache2/htdocs/index.html**

Como tengo el permiso denegado, se puede editar de una manera que no requiere permisos adicionales:

echo "Bienvenido/a a mi primera web desplegada con Docker" > /usr/local/apache2/htdocs/index.html

3. Conéctate desde el navegador del ordenador host y muestra el contenido de tu servidor web en Docker.

Poniendo: http://localhost:3000 en el navegador, tiene que aparecer:



Bienvenido/a a mi primera web desplegada con Docker

4. Muestra desde la interfaz gráfica de Docker Desktop info del contenedor

Containers / sofia-apachewebserver

sofia-apachewebserver

1893341df637 [http:latest](#) **STATUS** Running (17 minutes ago)

3000:80

Logs Inspect Bind mounts Exec Files Stats

```
2025-03-19 17:37:04 AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
2025-03-19 17:37:04 AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
2025-03-19 17:37:04 [Wed Mar 19 16:37:04.214174 2025] [mpm_event:notice] [pid 1:tid 1] AH00489: Apache/2.4.63 (Unix) configured -- resuming normal operations
2025-03-19 17:37:04 [Wed Mar 19 16:37:04.215658 2025] [core:notice] [pid 1:tid 1] AH00094: Command line: 'httpd -D FOREGROUND'
2025-03-19 17:51:25 172.17.0.1 - - [19/Mar/2025:16:51:25 +0000] "GET / HTTP/1.1" 200 52
2025-03-19 17:51:26 172.17.0.1 - - [19/Mar/2025:16:51:26 +0000] "GET /favicon.ico HTTP/1.1" 404 196
2025-03-19 17:52:18 172.17.0.1 - - [19/Mar/2025:16:52:18 +0000] "-" 408 -
```

El despliegue del servidor web con Apache en Docker se ha realizado con éxito. La página de inicio ha sido personalizada correctamente y el servidor ha respondido sin problemas. A pesar de los mensajes en los logs, el servidor está funcionando correctamente, y el contenedor está accesible a través del puerto 3000.

Parte 3 - Despliegue de un servidor FTP con vsftpd






FTP (File Transfer Protocol) es el protocolo de transferencia de archivos por excelencia. En esta parte, enviaremos archivos desde el host al contenedor utilizando este protocolo.

1. Crea un contenedor con las siguientes características:

- Tipo demonio
- Utiliza la imagen 'fauria/vsftpd'
- El nombre del contenedor es 'tunombre-ftpserver'
- Implementa las siguientes variables de entorno
 - FTP_USER=tunombre
 - FTP_PASS=1234
 - PASV_ENABLE=YES
 - -p 30000-30005:30000-30005
 - -e PASV_MIN_PORT=30000 -e PASV_MAX_PORT=30005
- Permitirá el acceso a su contenido desde el puerto 21 del host

```
C:\Users\sofia>docker run -d --name sofia-ftpserver -e FTP_USER=sofi -e FTP_PASS=1234 -e PASV_ENABLE=YES -e PASV_MIN_PORT=30000 -e PASV_MAX_PORT=30005 -p 21:21 -p 30000-30005:30000-30005 fauria/vsftpd
Unable to find image 'fauria/vsftpd:latest' locally
latest: Pulling from fauria/vsftpd
2d473b07cdd5: Pull complete
33b95e46f70b: Pull complete
e22029c8d9a7: Pull complete
e1871c5d8fc9: Pull complete
c17c1255c529: Pull complete
ddcbab051542: Pull complete
1c68b0b593f1: Pull complete
dadb66293c59: Pull complete
99a54b7a405b: Pull complete
200facf93d0a: Pull complete
16ecaf7d0305: Pull complete
Digest: sha256:6d71d7c7f1b0ab2844ec7dc7999a30aef6d758b6d8179cf5967513f87c79c177
Status: Downloaded newer image for fauria/vsftpd:latest
48f178872a059cc36f6647b44797af92ef5eff8a35b0cce5829a8f7bbf5c439c

C:\Users\sofia>
```

		sofia-ftpserver	48f178872a05	fauria/vsftpd 21:21 	0% 7 minute		
Show all ports (7)							

Comando utilizado:

```
docker run -d --name sofia-ftpserver \
-e FTP_USER=sofia \
-e FTP_PASS=1234 \
-e PASV_ENABLE=YES \
-e PASV_MIN_PORT=30000 \
-e PASV_MAX_PORT=30005 \
-p 21:21 \
-p 30000-30005:30000-30005 \
fauria/vsftpd
```

Explicación de los parámetros:

docker run -d: Crea y ejecuta el contenedor en modo demonio (en segundo plano).

--name sofia-ftpserver: el nombre "sofi-ftpserver" para el contenedor.

-e FTP_USER=sofia: se crea un usuario FTP "sofia".

-e FTP_PASS=1234: contraseña "1234" del usuario FTP.

-e PASV_ENABLE=YES: activa el modo pasivo, necesario para conexiones FTP detrás de NAT.

-e PASV_MIN_PORT=30000 -e PASV_MAX_PORT=30005: Define el rango de puertos pasivos de 30000 a 30005.

-p 21:21: Expone el puerto 21 (FTP) del contenedor en el puerto 21 del host.

-p 30000-30005:30000-30005: Expone los puertos pasivos 30000-30005 necesarios para las conexiones FTP.

fauria/vsftpd: Usa la imagen de vsftpd desde Docker Hub.

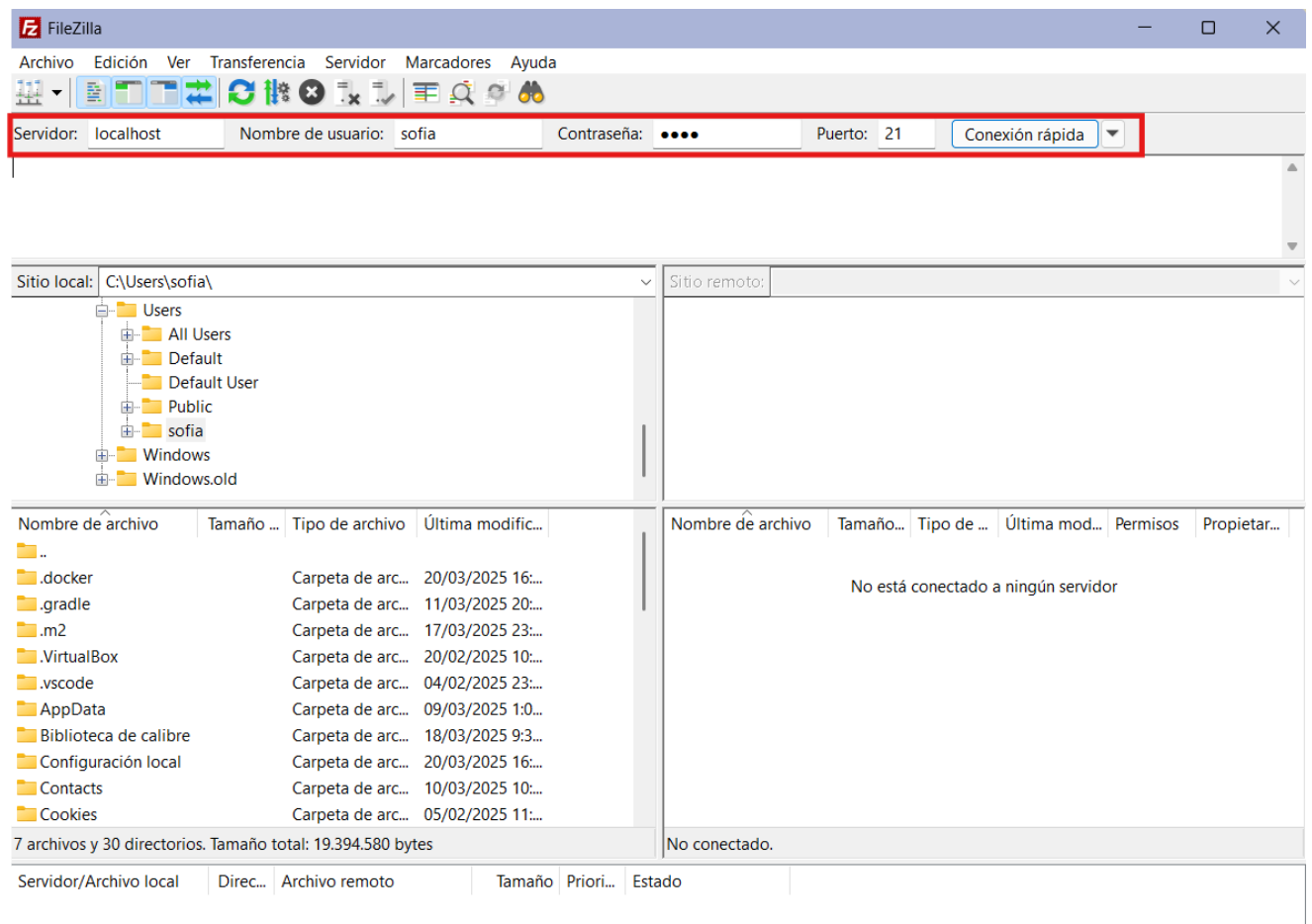
2. Instala el cliente FTP 'Filezilla' en tu máquina principal y conéctate con las credenciales indicadas al crear el contenedor.

Descargar en el enlace: <https://filezilla-project.org/>



The screenshot shows the FileZilla website. At the top, there is a red banner for "FileZilla® Pro" with the tagline "The Best FTP Solution" and a "GET IN NOW >" button. Below the banner, the "Overview" section welcomes visitors to the homepage of FileZilla®, describing it as a free FTP solution that supports FTP, FTPS, and SFTP. It also mentions "FileZilla Pro" with additional protocol support for WebDAV, Amazon S3, Backblaze B2, Dropbox, Microsoft OneDrive, Google Drive, and Google Cloud Storage. The "Quick download links" section is highlighted with a red box, showing two buttons: "Download FileZilla Client" and "Download FileZilla Server", both labeled "All platforms". Below these buttons, a note states: "Pick the client if you want to transfer files. Get the server if you want to make files available for others."

Una vez abierta la aplicación, hay que poner esto para conectarse al servidor y pulsar en conexión rápida:



Estado: Conexión establecida, esperando el mensaje de bienvenida...

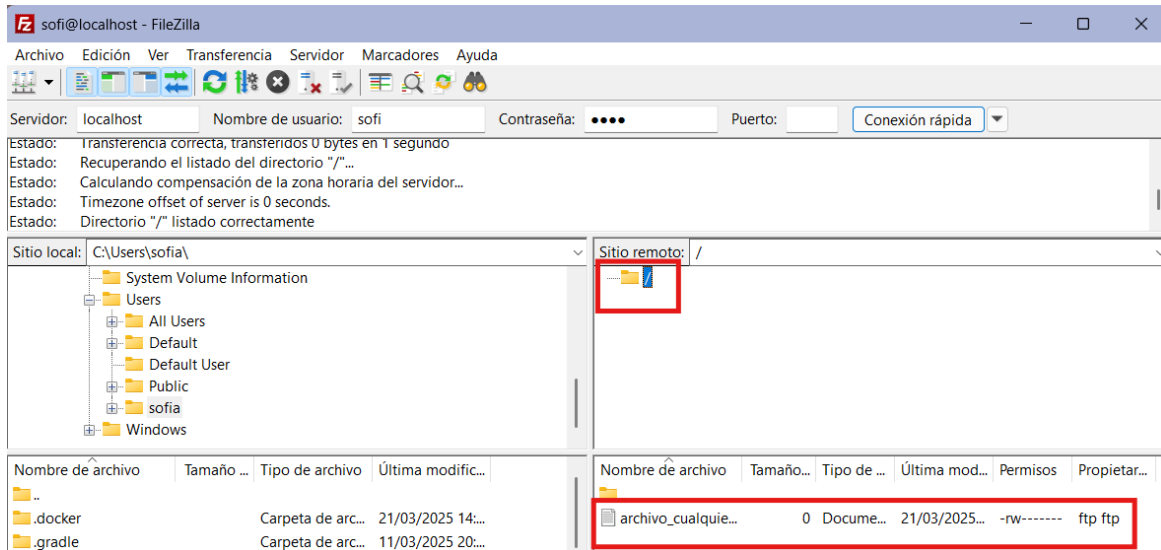
Estado: Servidor no seguro, no soporta FTP sobre TLS.

Estado: Registrado en

Estado: Recuperando el listado del directorio...

3. Sube un archivo al contenedor (en la ruta que quieras):

1. He creado un archivo de texto cualquiera para subirlo al contenedor.
2. Hay que arrastrar el archivo a la parte derecha, donde está el directorio / del servidor FTP.
3. Después hay que asegurarse de que aparece en la lista de archivos del servidor.



4. Verifica desde el terminal que el archivo se ha subido mostrando una lista de los archivos de la carpeta en la cual lo has puesto. Adjunta el comando y explica los parámetros utilizados.

Para entrar en el contenedor, se utiliza el comando: **docker exec -it sofia-ftpserver /bin/sh**

Y dentro del contenedor, hay que ir al directorio donde se ha subido el archivo. Comando:
ls -l /home/vsftpd/

Al ejecutar este comando, tiene que aparecer el archivo que se ha subido con FileZilla. Si aparece en la lista, significa que la transferencia se ha hecho bien.

En mi caso, el archivo está dentro de la carpeta 'sofi'. Para ver el archivo dentro de la carpeta, utilizo el comando: **ls -l /home/vsftpd/sofi/**

```
PS C:\Users\sofia> docker exec -it sofia-ftpserver /bin/sh
sh-4.2# ls -l /home/vsftpd/
total 4
drwxr-xr-x 2 ftp ftp 4096 Mar 21 14:10 sofia
sh-4.2# ls -l /home/vsftpd/sofi/
total 0
-rw-rw-rw- 1 ftp ftp 0 Mar 21 14:10 archivo_cualquiera_sofia.txt
sh-4.2#
```