



José Gonzalo Almendros Navas

Paso 1: Crea tu archivo docker-compose y comprueba que tu contenedor con tu primer servicio se levanta con normalidad.

Importante poner la etiqueta services =)

```
services:
  sftp:
    image: atmoz/sftp
    container_name: t4-sftp
    ports:
      - "2222:22"
    volumes:
      - ./data/ftp/alumnos:/data
    command: >
      alumno1:Alumno123:1000:1000:/data/alumno1
      alumno2:Alumno123:1001:1001:/data/alumno2
      profesor:Profe123:1002:1002:/data/profesor
    restart: unless-stopped
```

Paso 2: Conéctate a SFTP desde la terminal y por el cliente gráfico.
Comprueba la subida y bajada de archivos.

Terminal:

```
o alezco@Alezc0s-MacBook-Air alumno1 % sftp -P 2222 alumno1@localhost
alumno1@localhost's password:
Connected to localhost.
sftp> get prueba.txt
File "/prueba.txt" not found.
sftp> get ./prueba.txt
File "../prueba.txt" not found.
sftp> put prueba.txt
Uploading prueba.txt to /prueba.txt
dest open "/prueba.txt": Permission denied
sftp> pwd
Remote working directory: /
sftp> ls
data
sftp> cd data/ftp/alumnos/alumno1
realpath /data/ftp/alumnos/alumno1: No such file
sftp> ls
data
sftp> cd data
sftp> ls
alumno1
sftp> cd alumno1
sftp> put prueba.txt
Uploading prueba.txt to /data/alumno1/prueba.txt
prueba.txt
sftp> 
```

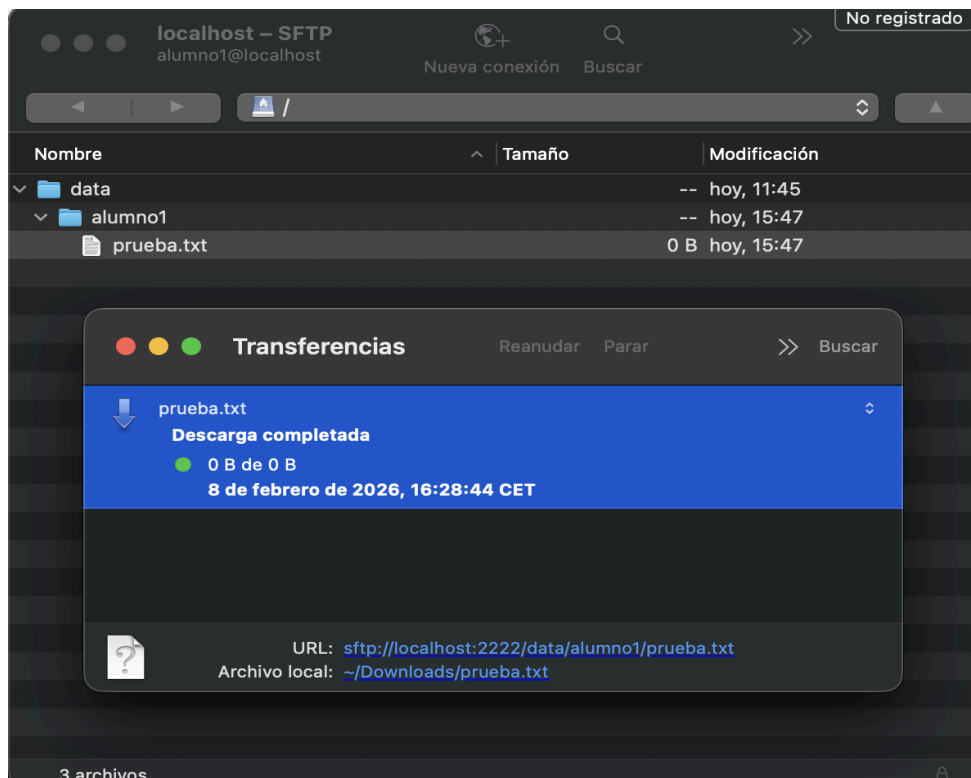
INICIO SESIÓN CON alumno1

PRUEBAS (no entendía)

SUBO prueba.txt a ALUMNO1

100% 0 0.0KB/s 00:00

Interfaz: (he usado Cyberduck)



Paso 3: *Añade el nuevo servicio webdav a tu docker-compose y comprueba su correcto funcionamiento.*

La imagen predeterminada de nginx-alpine no me funcionaba porque no incluye el módulo `nginx-dav-ext-module` necesario para usar webDAV 🤔 (mejor dicho, para navegar entre carpetas con webDAV). Tampoco funcionó con la imagen de ugeek, dgraziotin ni bytemark (esta modificación viene a coalición del punto 4, donde no podía entrar al `localhost:8080`).

```
webdav:
  build: ./webdav
  container_name: t4-webdav
  ports:
    - "8080:8080"
  volumes:
    - ./data/webdav/documentos:/data
  restart: unless-stopped
```

He tenido que generar (con ayuda de Claude) un Dockerfile que use alpine, instale a través de apk el módulo antes mencionado y lea los archivos de configuración y contraseñas hasheadas

```
dor_sftp_despligue > webdav > 📄 Dockerfile
FROM nginx:alpine

# Instalar dependencias necesarias
RUN apk add --no-cache \
    nginx-mod-http-dav-ext \
    apache2-utils

# Crear directorio de datos
RUN mkdir -p /data /tmp/client_body_temp

# Copiar configuración personalizada
COPY nginx.conf /etc/nginx/nginx.conf
COPY htpasswd /etc/nginx/htpasswd

# Dar permisos
RUN chmod 644 /etc/nginx/htpasswd

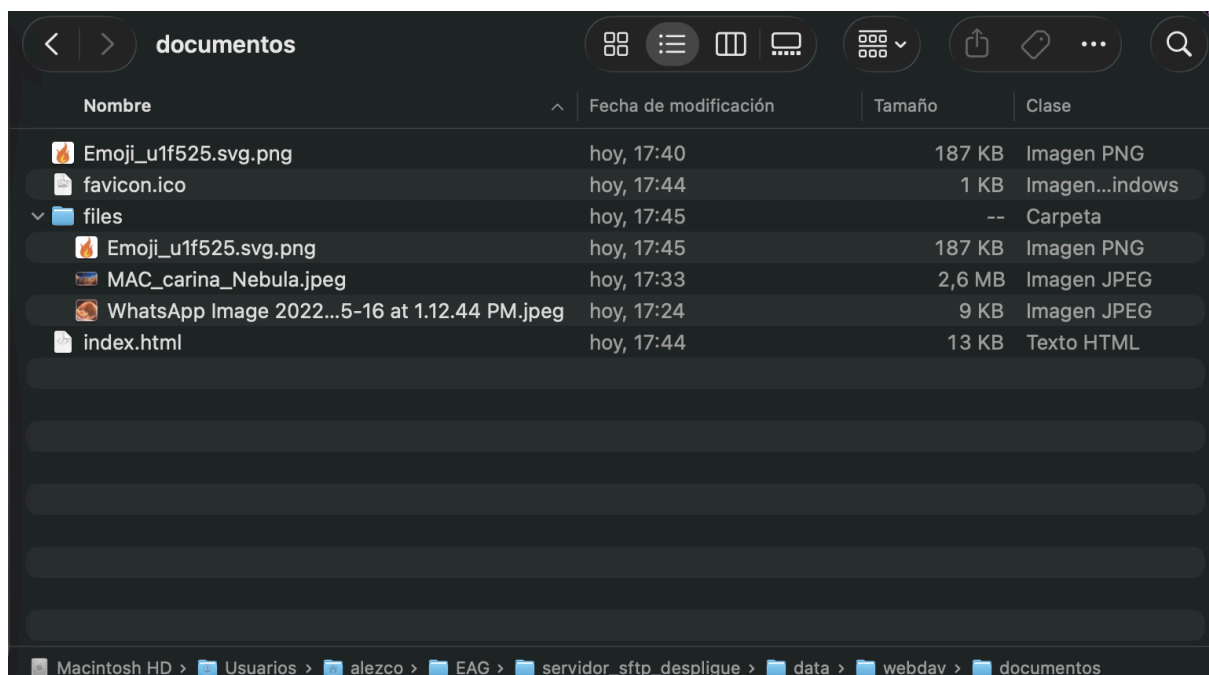
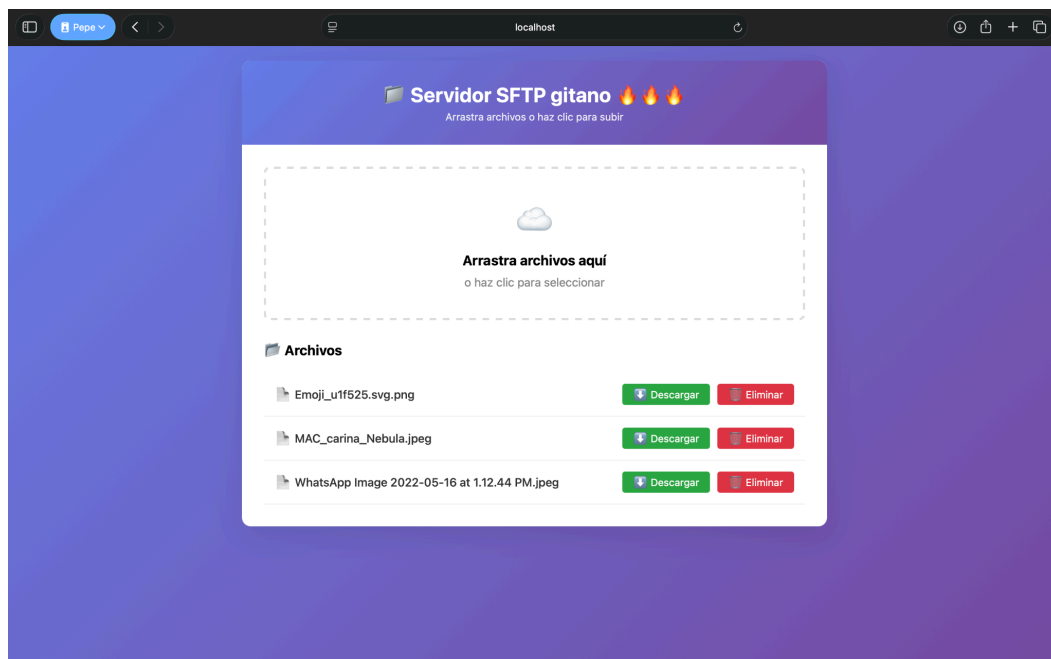
EXPOSE 8080

CMD ["nginx", "-g", "daemon off;"]
```

Contraseñas hasheadas

```
servidor_sftp_despligue > webdav > ≡ htpasswd  
1 alumno1:$apr1$t1l14xDh$dT.L6sAYwSBdUadzHnQi.1  
2 alumno2:$apr1$eEAzbgkm$8F0MbUB2kzPRwYYWFdS19.  
3 profesor:$apr1$2a39jGBm$sVIEWEC5//eUTb/pWdF4q1
```

Paso 4: Una vez se levante con normalidad, accede al servidor y realiza subidas y bajadas de archivos.



Como me daba pereza usar los comandos para subir y bajar archivos, he creado un index.html para cargarlo al localhost y hacer la interacción bonita.