

# Docker Compose

Documento realizado por Roberto Delgado Sánchez - Alumno de Despliegue de Aplicaciones Web - DAW

## Docker Compose

1. Enunciado
2. Despliegue del contenedor
3. Probando la aplicación

## 1. Enunciado

---


En este ejercicio vamos a trabajar con **FileBrowser**, aplicación de código abierto que permite administrar y compartir archivos desde un navegador web de forma sencilla y segura, es decir, que sirve para crear una nube particular. Proporciona una interfaz intuitiva que soporta tareas como subir, descargar, mover, copiar, renombrar y borrar ficheros y carpetas.

Asimismo, incluye características avanzadas como al edición de archivos de texto en el navegador, la creación de enlaces compartidos con configuraciones personalizables y la configuración de permisos de acceso para varios usuarios.


Es compatible con múltiples sistemas operativos y puede integrarse en entornos como servidores domésticos, servicios en la nube o contenedores Docker, ofreciendo flexibilidad para distintos usos.

El proyecto está alojado en <https://hub.docker.com/r/hurlenko/filebrowser>.

← → ↺ 🏠 📄 hub.docker.com/r/hurlenko/filebrowser

 **dockerhub** Explore My Hub

[Explore](#) / [hurlenko](#) / filebrowser



## hurlenko/filebrowser


By [hurlenko](#) · Updated 3 months ago

Web-based file manager. Small image size. Supports arm, arm64 and amd64 architectures.

[IMAGE](#)

☆82    ↓ 10M+

Overview    Tags



# File Browser

[filebrowser](#) ↗ inside a [docker container](#)

release **v2.32.0**   image size **25.1 MiB**   docker pulls **16M**   docker stars **82**

### Introduction

filebrowser provides a file managing interface within a specified directory and it can be used to upload, delete, preview, rename and edit your files. It allows the creation of multiple users and each user can have its own directory. It can be used as a standalone app or as a middleware.

## 2. Despliegue del contenedor

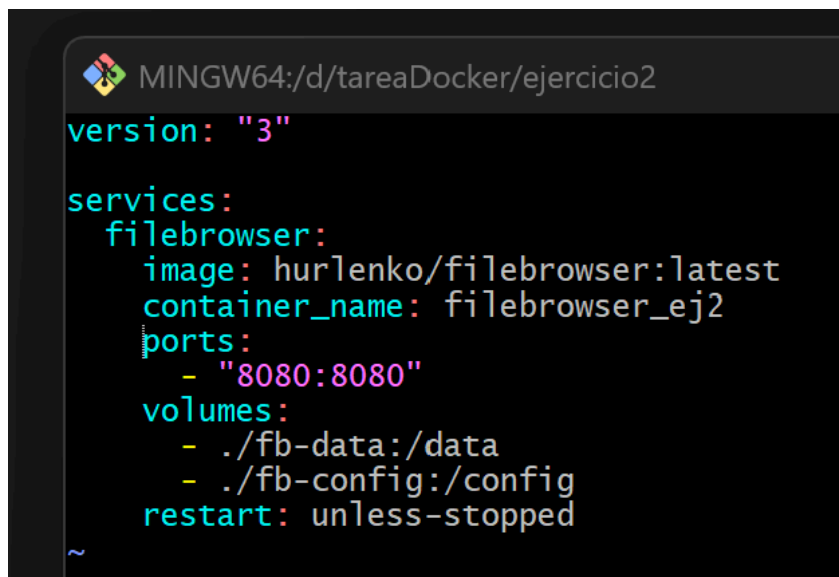
Para poder desplegar esta aplicación debemos crear un archivo `docker-compose.yaml` con el siguiente contenido, creado a partir del modelo básico que se muestra en la web del proyecto:

- `image` : imagen de partida para crear el contenedor que permitirá ejecutar la aplicación.
- `container_name` : nombre que queremos reciba el contenedor.
- `ports` : puerto en el que "escuchará" las peticiones esta aplicación.
- `volumes` : carpetas del host local en las que se guardarán los datos (usamos `bind-mount` ).
- `restart` : el contenedor se reinicia al detener su ejecución salvo que lo detenga un administrador.

```
services:
  filebrowser:
    image: hurlenko/filebrowser:latest
    container_name: filebrowser_ej2
    ports:
      - "8080:8080"
    volumes:
      - ./fb-data:/data
      - ./fb-config:/config
    restart: unless-stopped
```

Copiamos este archivo `docker-compose.yaml` en la carpeta del `ejercicio2` y ejecutamos este comando para crear el contenedor aprovechando la aplicación **Git Bash**:

```
$ docker compose up -d
```

A screenshot of a Git Bash terminal window. The title bar shows the path "MINGW64:/d/tareaDocker/ejercicio2". The terminal displays the content of a file, which is a Docker Compose configuration. The text is as follows:

```
version: "3"

services:
  filebrowser:
    image: hurlenko/filebrowser:latest
    container_name: filebrowser_ej2
    ports:
      - "8080:8080"
    volumes:
      - ./fb-data:/data
      - ./fb-config:/config
    restart: unless-stopped
```

```

rober@SurfacePro9 MINGW64 /d/tareaDocker/ejercicio2 (ejercicio2)
$ ls -al
total 9
drwxr-xr-x 1 rober 197609 0 Apr 17 21:01 ./
drwxr-xr-x 1 rober 197609 0 Apr 17 20:57 ../
-rw-r--r-- 1 rober 197609 238 Apr 17 21:01 docker-compose.yaml
drwxr-xr-x 1 rober 197609 0 Apr 17 21:01 ejercicio2.assets/
-rw-r--r-- 1 rober 197609 2455 Apr 17 21:01 ejercicio2.md

rober@SurfacePro9 MINGW64 /d/tareaDocker/ejercicio2 (ejercicio2)
$ docker compose up -d
time="2025-04-17T21:01:34+02:00" level=warning msg="D:\\tareaDocker\\ejercicio2\\docker-compose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
filebrowser Pulling
bbe6e0b5af75 Pulling fs layer
1f3e46996e29 Pulling fs layer
7b276281f0bb Pulling fs layer
bbe6e0b5af75 Download complete
7b276281f0bb Downloading [=====>] 2.097MB/8.328MB
7b276281f0bb Downloading [=====>] 4.194MB/8.328MB
1f3e46996e29 Downloading [=====>] 2.097MB/3.642MB
7b276281f0bb Downloading [=====>] 6.291MB/8.328MB
1f3e46996e29 Download complete
7b276281f0bb Downloading [=====] 8.328MB/8.328MB
1f3e46996e29 Extracting 1 s
7b276281f0bb Download complete
1f3e46996e29 Extracting 1 s
1f3e46996e29 Pull complete
7b276281f0bb Extracting 1 s
7b276281f0bb Pull complete
bbe6e0b5af75 Pull complete
filebrowser Pulled
Network ejercicio2_default Creating
Network ejercicio2_default Created
Container filebrowser_ej2 Creating
Container filebrowser_ej2 Created
Container filebrowser_ej2 Starting
Container filebrowser_ej2 Started

rober@SurfacePro9 MINGW64 /d/tareaDocker/ejercicio2 (ejercicio2)
$

```

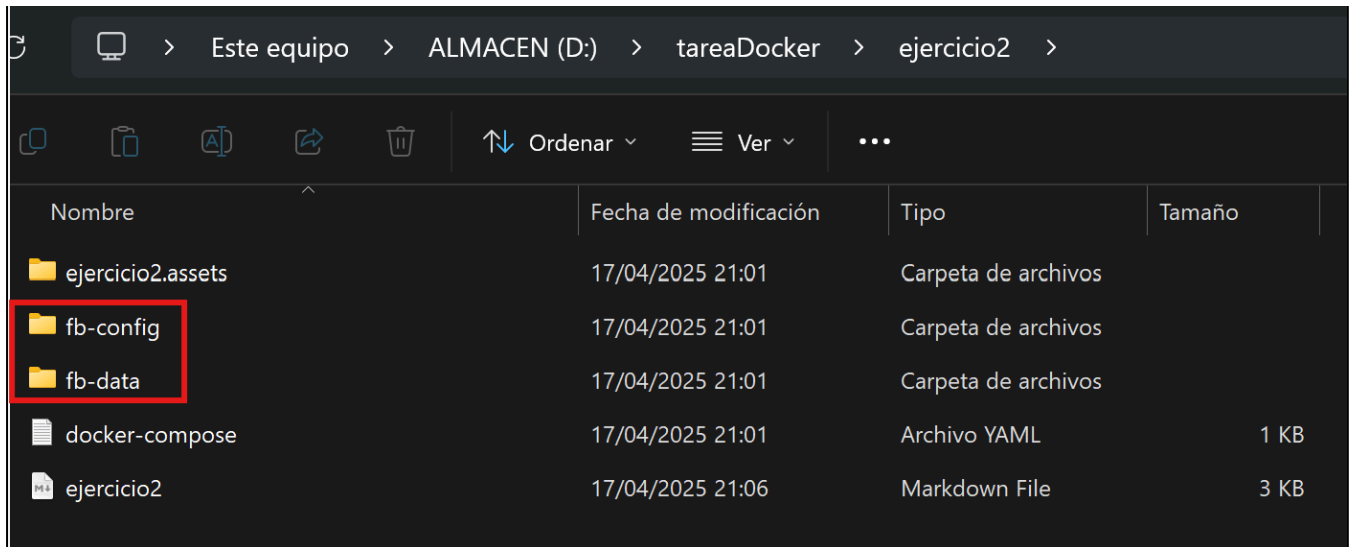
Resulta curioso que, a pesar de que aparece en el archivo `yaml` de ejemplo en la web del proyecto, al crear el contenedor nos diga que el parámetro `version` está obsoleto, por lo que aunque aparezca en la imagen de este archivo en la versión final se elimina.

```

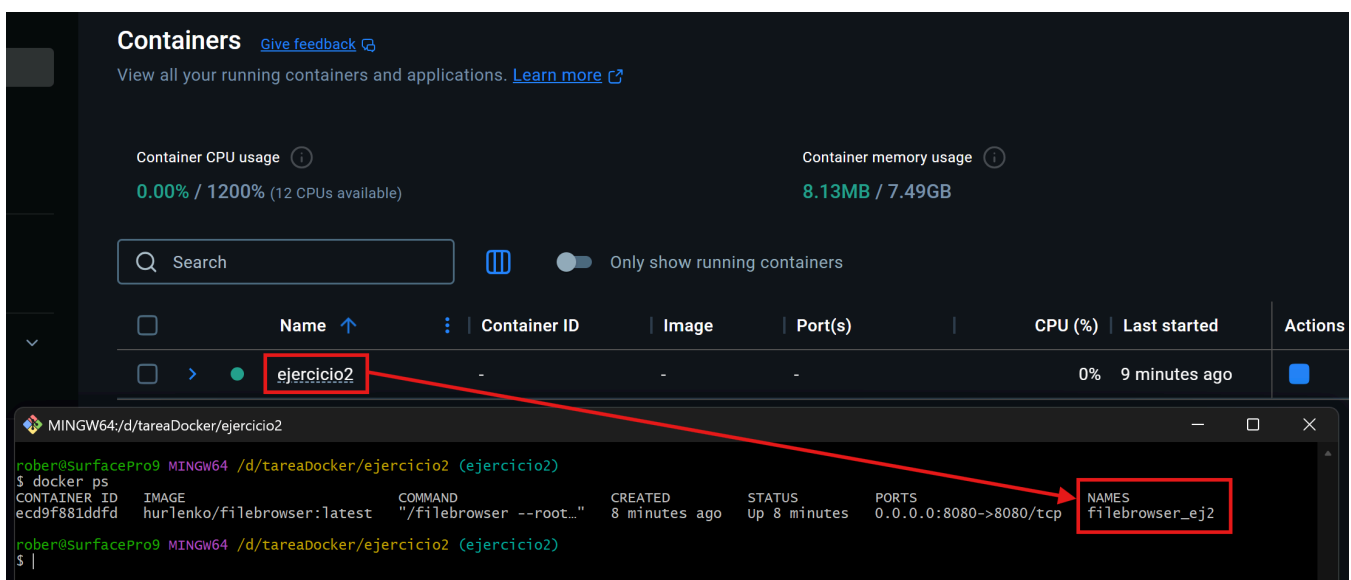
MINGW64:/d/tareaDocker/ejercicio2
services:
  filebrowser:
    image: hurlenko/filebrowser:latest
    container_name: filebrowser_ej2
    ports:
      - "8080:8080"
    volumes:
      - ./fb-data:/data
      - ./fb-config:/config
    restart: unless-stopped
~
docker-compose.yaml[+] [unix] (21:01 17/04/2025)

```

Una vez creado y puesto en marcha el contenedor podemos ver que se han creado las dos carpetas que van a almacenar los datos: `fb-data` y `fb-config`.

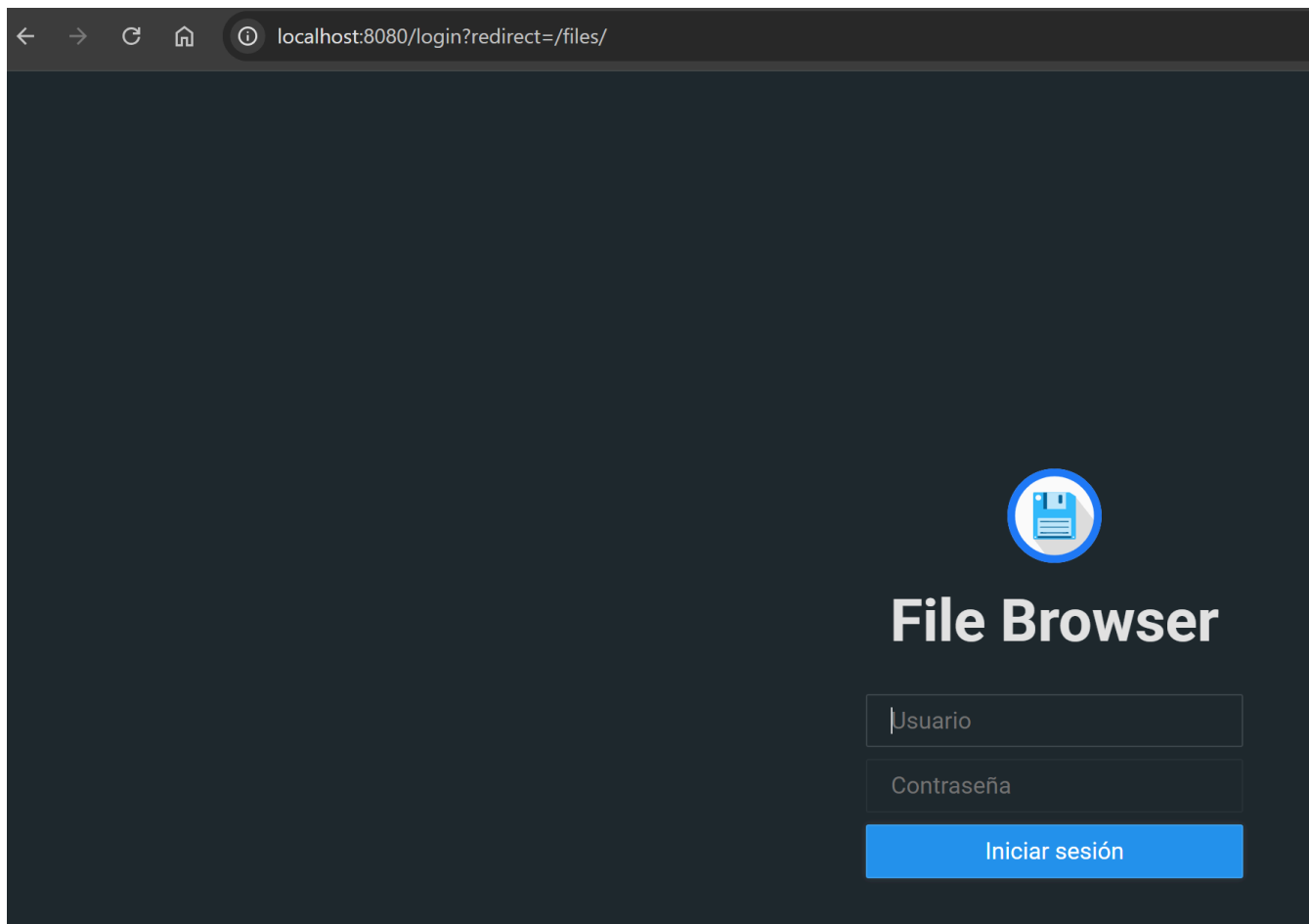


Asimismo, por razones desconocidas, en **Docker Desktop** el contenedor aparece con el nombre `ejercicio2`, mientras que si ejecutamos el comando `docker ps` para saber qué contenedores están en ejecución sí que aparece con el nombre definido en el archivo `yaml`, `filebrowser_ej2`:

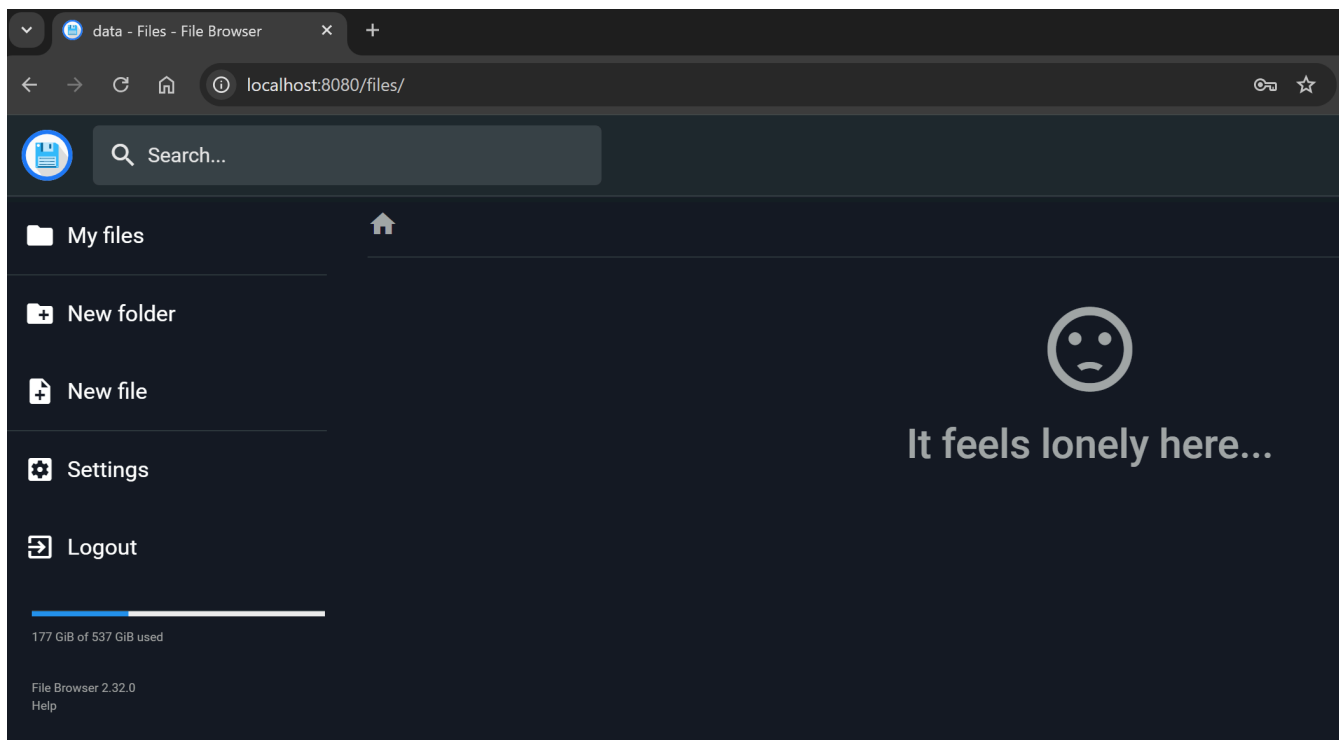


### 3. Probando la aplicación

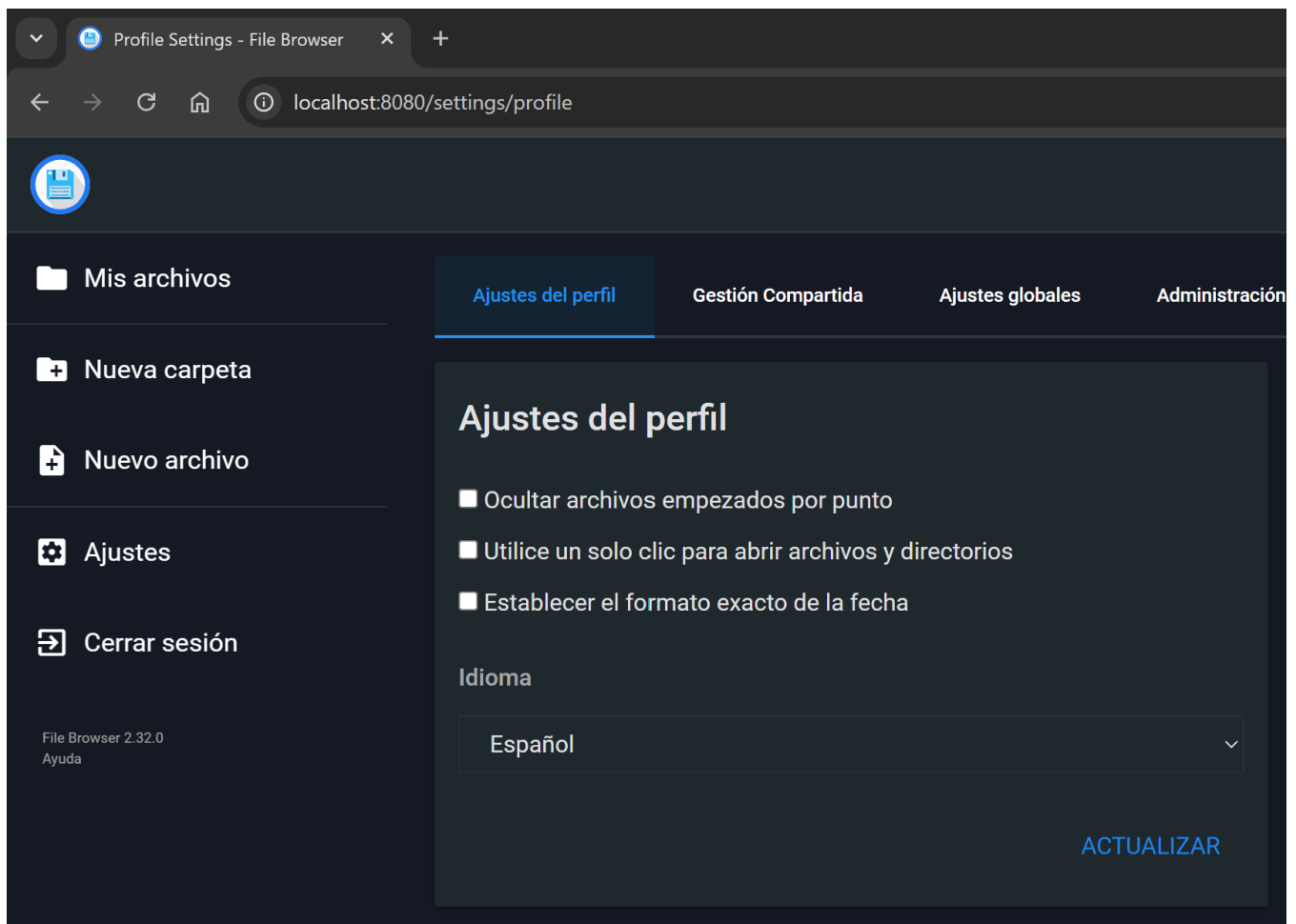
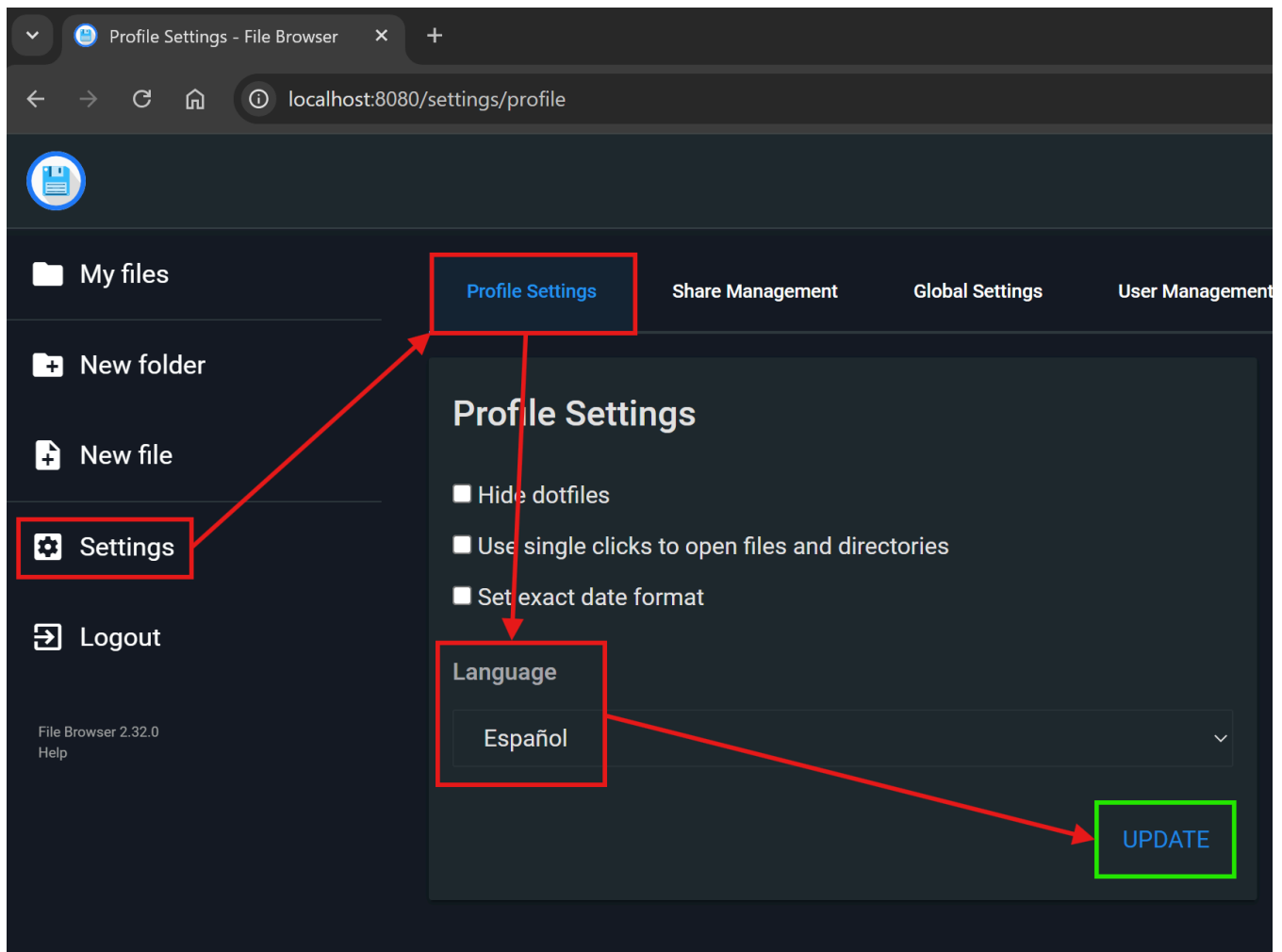
Para comprobar que ya tenemos **FileBrowser** funcionando correctamente sólo tenemos que abrir una ventana del navegador y conectarnos a <http://localhost:8080>:



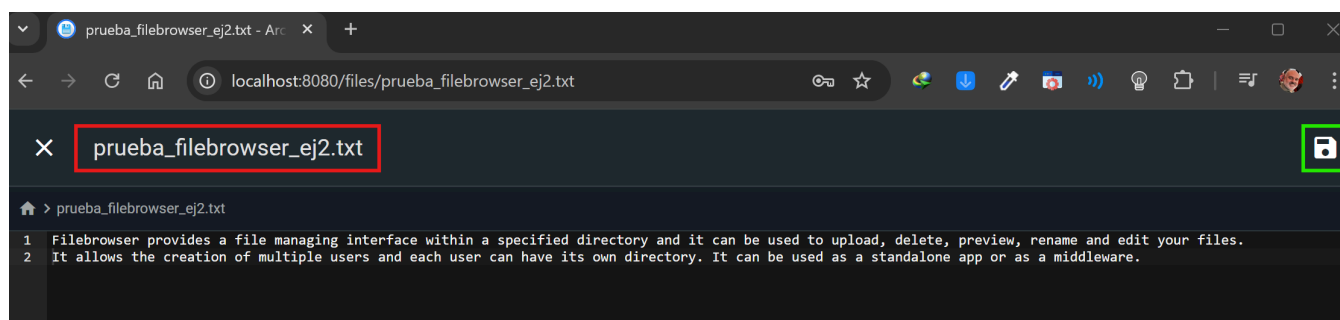
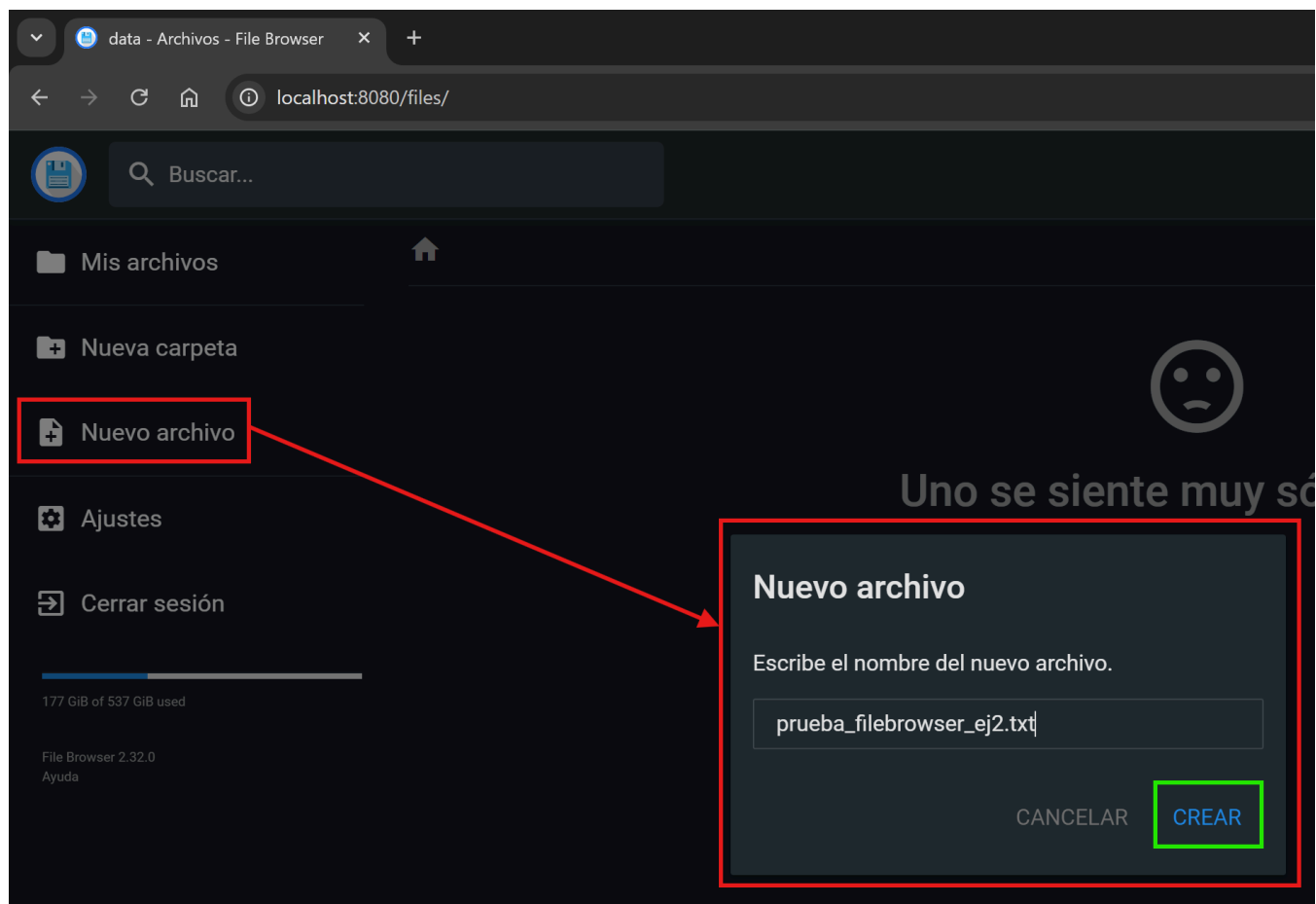
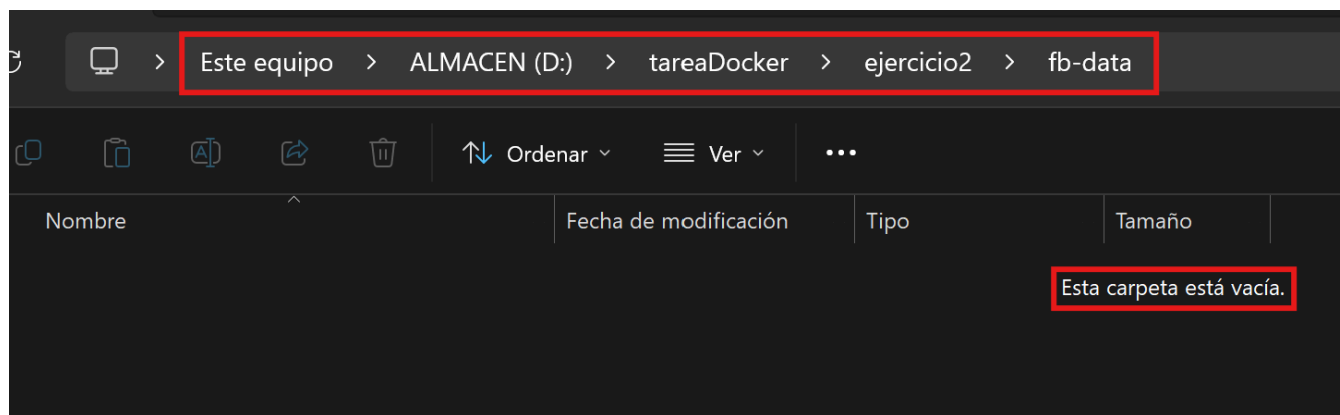
Entramos con los datos proporcionados (**admin/admin**):



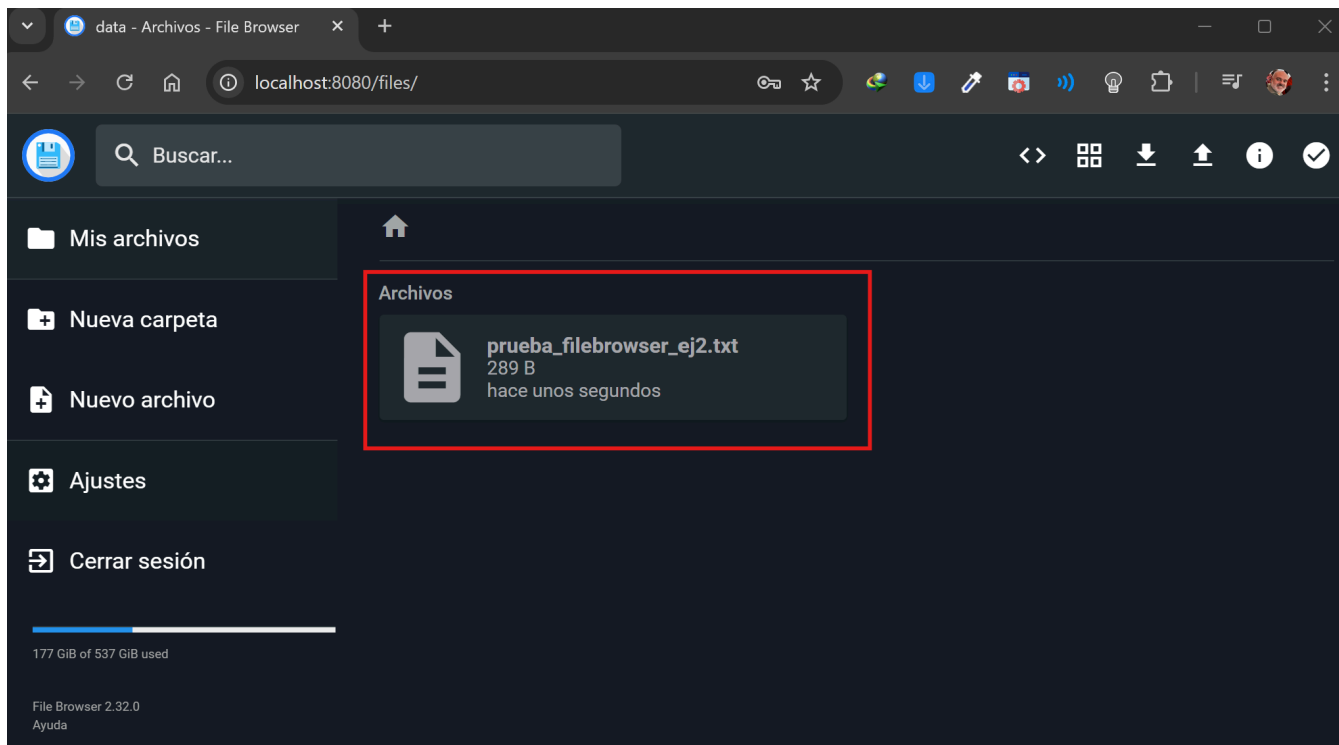
Cambiamos el idioma de la aplicación a **español** :



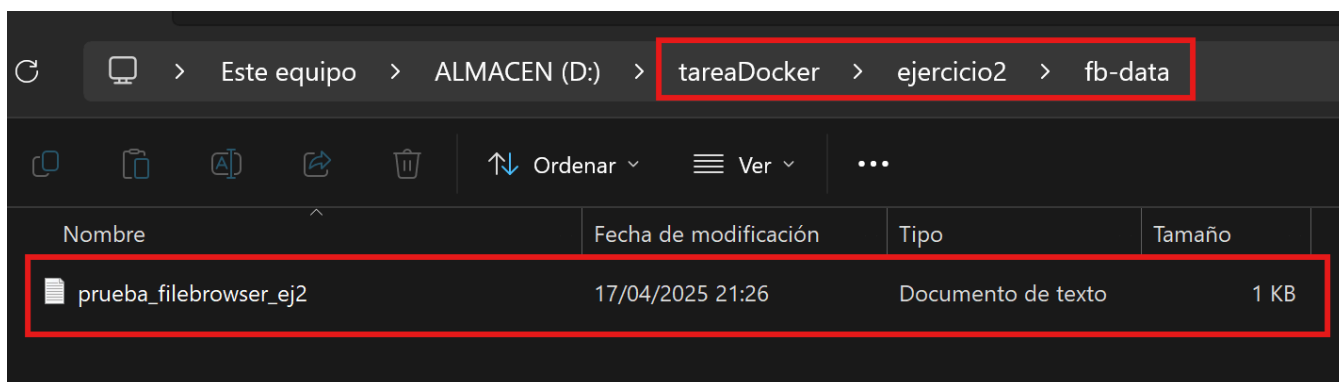
Creamos un archivo de texto (en la carpeta **fb-data** no hay nada todavía):



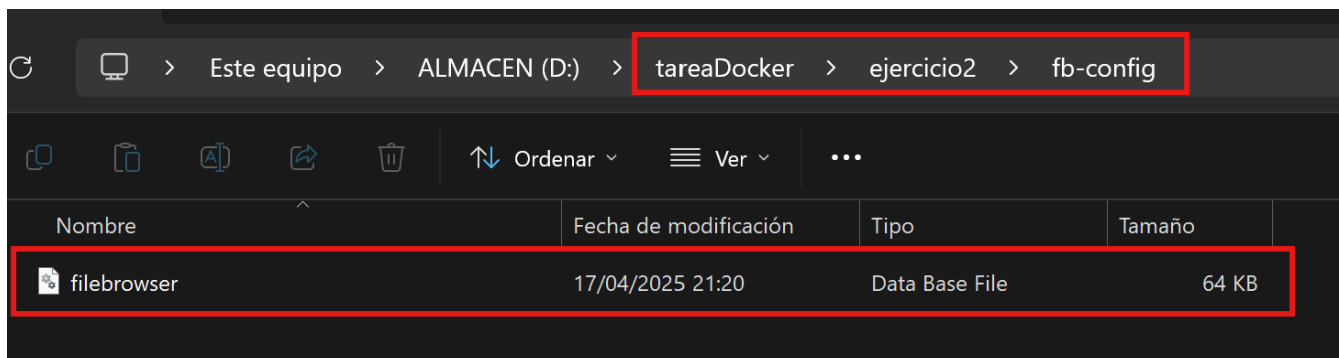




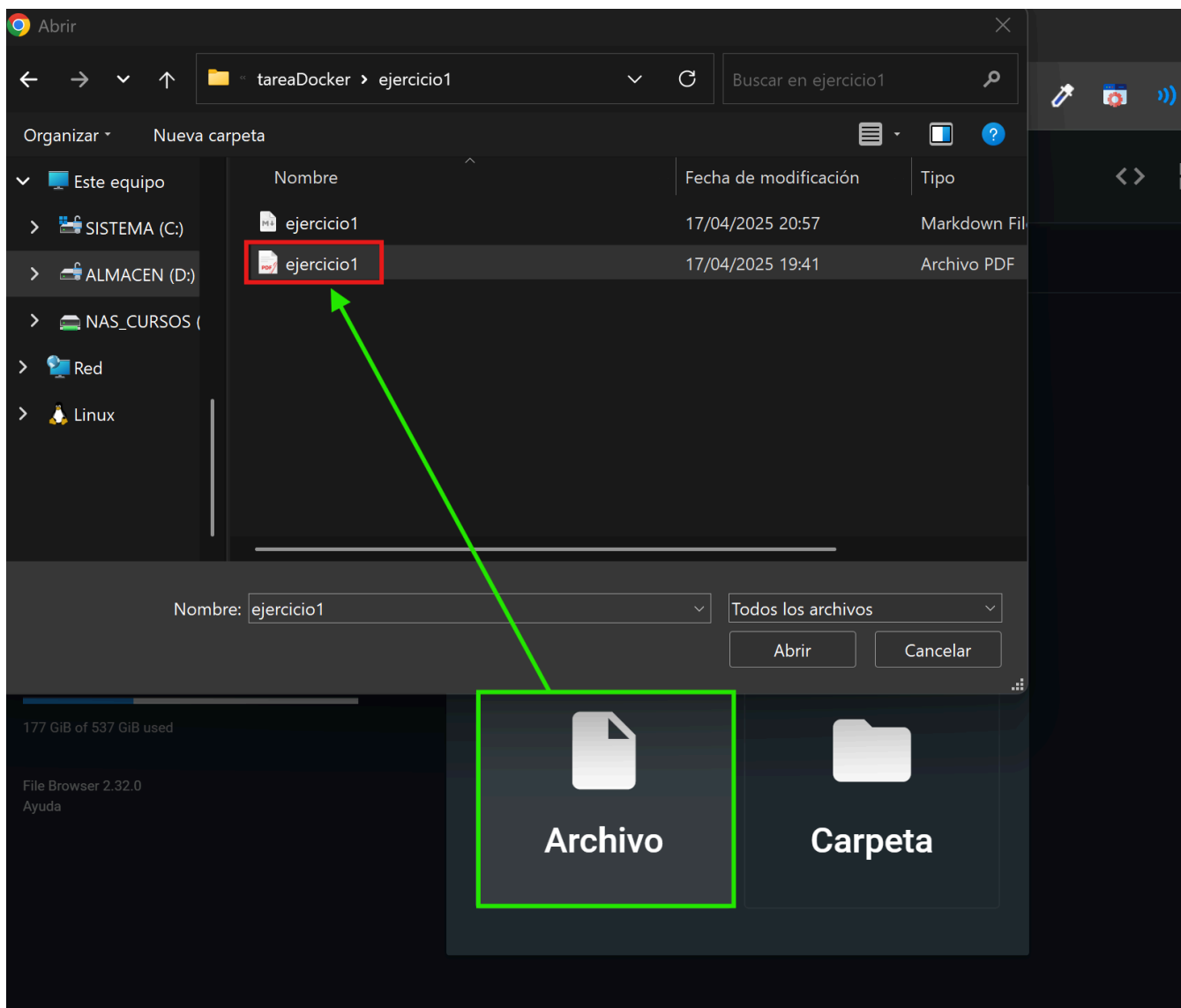
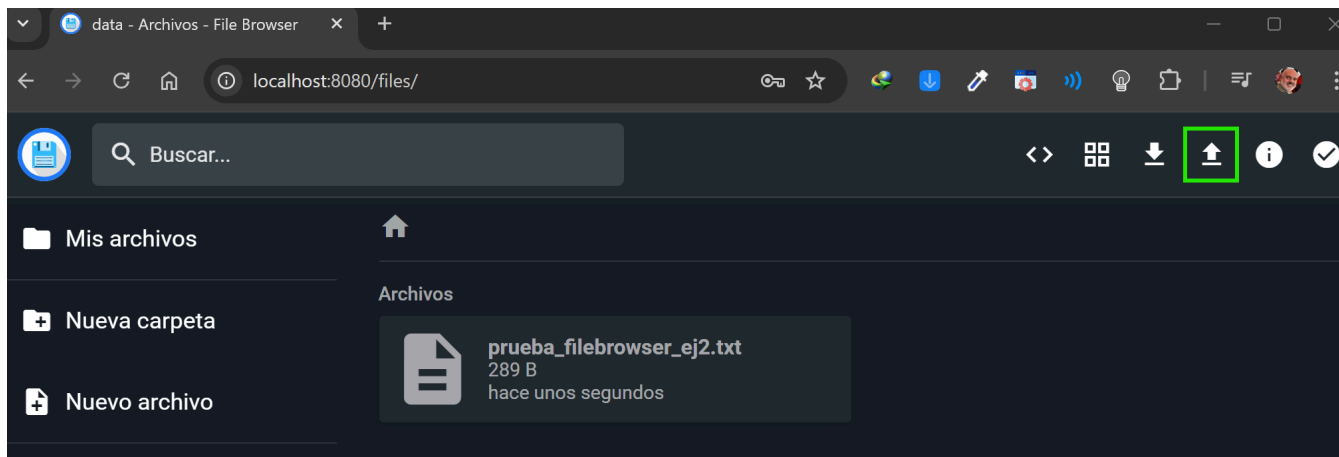
Si ahora vamos a la carpeta **fb-data** vemos que se ha creado ese archivo:

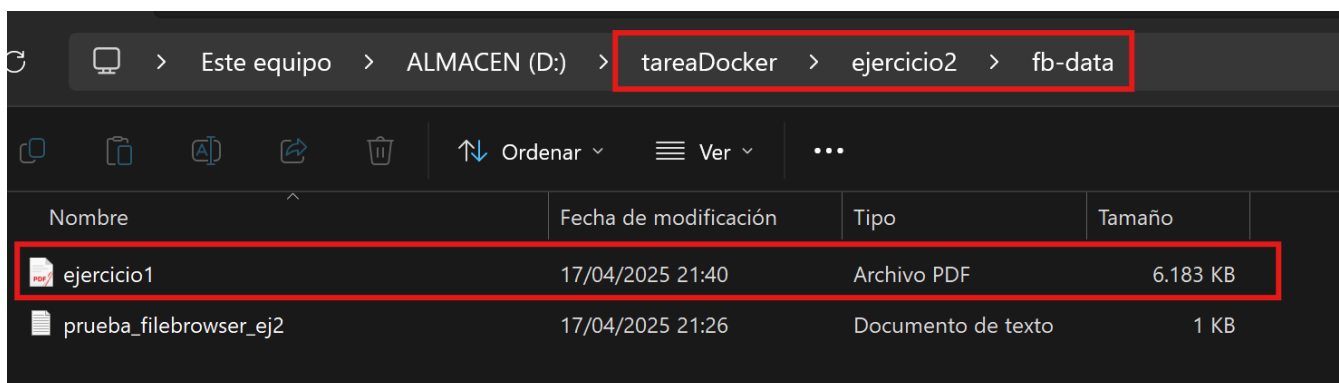
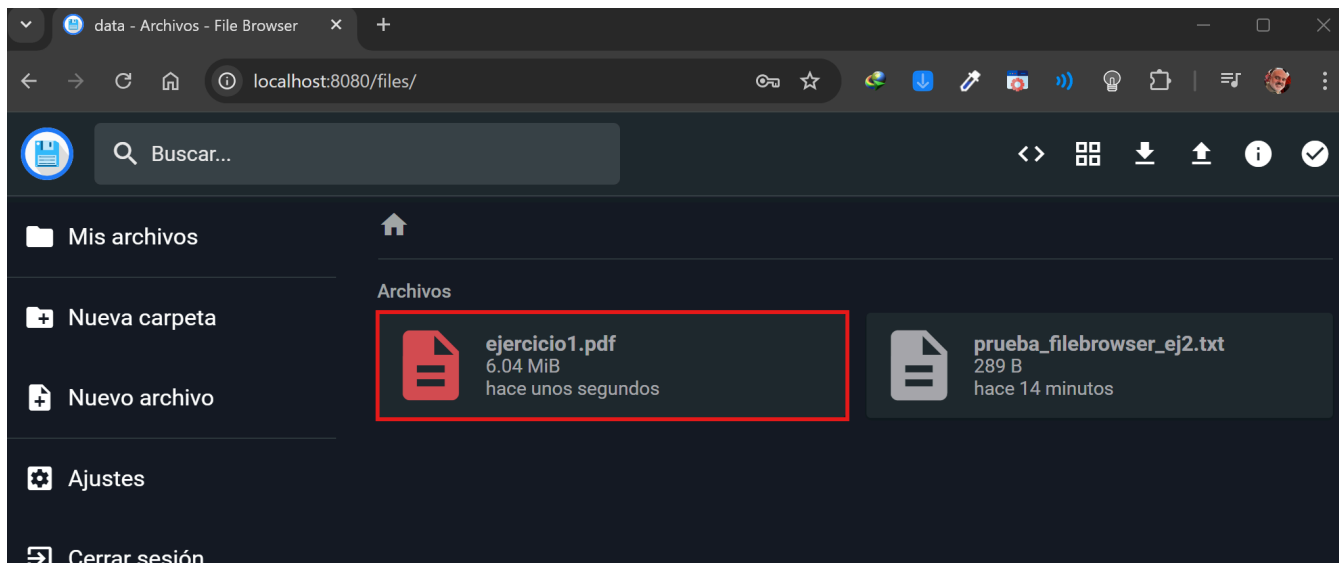


En la carpeta **fb-config** se ha creado automáticamente un archivo de nombre **filebrowser.db** imagino que con parámetros de configuración de la aplicación:



Vamos a subir un archivo a **FileBrowser**:





Y con esto finaliza la realización de este segundo ejercicio.