

Actividad Evaluable 3

Docker

Ejercicio 2 - Docker Compose

Despliegue de Aplicaciones Web - DAW Distancia

CIFP Sect. Industrial y Servicios - La Laboral

Curso 2024-2025

01 de abril de 2025

Emilio Zaera Vidal - 46.911.234-C

1. Introducción

2. Metodología

3. Preparativos

Creación de un nuevo repositorio `tareaDocker`

Clonado y vinculación en local

Creación de las ramas

4. Ejercicio 2 - Docker Compose

Breve explicación de la aplicación

`compose.yaml`

Despliegue de FileBrowser

Carpeta y volumen creados

Subida de un fichero y cambio de idioma en FileBrowser

1. Introducción

En el módulo de **Despliegue de Aplicaciones Web**, uno de los objetivos fundamentales es aprender a gestionar entornos de despliegue modernos utilizando tecnologías basadas en contenedores. En este contexto, la herramienta **Docker** se ha convertido en un estándar para la creación, configuración y administración de entornos aislados, facilitando el despliegue y la distribución de aplicaciones web.

La presente tarea evaluable tiene como finalidad reforzar los conocimientos adquiridos sobre Docker mediante la realización de tres ejercicios prácticos. A través de estos ejercicios, se trabajará la creación de contenedores en red, la orquestación de servicios y la construcción de una imagen personalizada. Todo ello permitirá al alumno familiarizarse con el ciclo completo de creación, despliegue y gestión de contenedores, así como con las buenas prácticas de documentación y organización de proyectos en un repositorio.

La **orquestación de servicios** es el proceso de coordinar y gestionar varios contenedores para que funcionen juntos como una única aplicación. En Docker, se realiza con herramientas como **Docker Compose**, que permiten definir y automatizar la configuración y despliegue de todos los servicios desde un solo archivo.

2. Metodología

La metodología propuesta para esta tarea consiste en la resolución práctica de tres ejercicios diferenciados. En cada uno de ellos, se deberá crear un entorno utilizando distintas herramientas de Docker: redes y contenedores mediante Docker Desktop, despliegue con Docker Compose y creación de una imagen personalizada con Dockerfile. Todo el trabajo se documentará y organizará en un repositorio público de GitHub, utilizando ramas para cada ejercicio. Además, como parte de la evaluación, se solicita un videoclip donde el estudiante muestre y explique parte del trabajo realizado.

3. Preparativos


Creación de un nuevo repositorio tareaDocker

Creo un nuevo repositorio público en [mi GitHub](#) para la tarea:

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

| | |
|--|-------------------|
| Owner * | Repository name * |
|  emiliozv ▾ | / tareaDocker |

✔ tareaDocker is available.

Clonado y vinculación en local

Trabajaré en local para, al finalizar, subir todo al repositorio remoto en GitHub. Para ello, creo un repositorio local mediante la línea de comandos de git, genero las carpetas y los ficheros `.md`, y lo vinculo con el remoto:

```
EyM@Sobremesa MINGW64 /d/TRABAJO/Formación/DAW/02 CURSO/Despliegue/Tareas/Evaluables/T3 Docker/repoTareaDocker
$ git init
Initialized empty git repository in D:/TRABAJO/Formación/DAW/02 CURSO/Despliegue/Tareas/Evaluables/T3 Docker/repoTareaDocker/.git/

EyM@Sobremesa MINGW64 /d/TRABAJO/Formación/DAW/02 CURSO/Despliegue/Tareas/Evaluables/T3 Docker/repoTareaDocker (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

EyM@Sobremesa MINGW64 /d/TRABAJO/Formación/DAW/02 CURSO/Despliegue/Tareas/Evaluables/T3 Docker/repoTareaDocker (main)
$ git commit -m "Creación de carpetas"
[main (root-commit) 4be2a88] Creación de carpetas
 4 files changed, 170 insertions(+)
 create mode 100644 Ejercicio1/tarea_evaluable_3_docker_ej1_Emilio_Zaera_Vidal.assets/image-20250401103607546.png
 create mode 100644 Ejercicio1/tarea_evaluable_3_docker_ej1_Emilio_Zaera_Vidal.md
 create mode 100644 Ejercicio2/tarea_evaluable_3_docker_ej2_Emilio_Zaera_Vidal.md
 create mode 100644 Ejercicio3/tarea_evaluable_3_docker_ej3_Emilio_Zaera_Vidal.md

EyM@Sobremesa MINGW64 /d/TRABAJO/Formación/DAW/02 CURSO/Despliegue/Tareas/Evaluables/T3 Docker/repoTareaDocker (main)
$ git branch -M main
```

```
EYM@Sobremesa MINGW64 /d/TRABAJO/Formación/DAW/02 CURSO/Despliegue/Tareas/Evaluables/T3 Docker/repoTareaDocker (main)
$ git remote add origin https://github.com/emiliozv/tareaDocker.git

EYM@Sobremesa MINGW64 /d/TRABAJO/Formación/DAW/02 CURSO/Despliegue/Tareas/Evaluables/T3 Docker/repoTareaDocker (main)
$ git remote -v
origin https://github.com/emiliozv/tareaDocker.git (fetch)
origin https://github.com/emiliozv/tareaDocker.git (push)

EYM@Sobremesa MINGW64 /d/TRABAJO/Formación/DAW/02 CURSO/Despliegue/Tareas/Evaluables/T3 Docker/repoTareaDocker (main)
$ git push -u origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 32 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 57.28 KiB | 19.09 MiB/s, done.
Total 10 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/emiliozv/tareaDocker.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

The screenshot shows the GitHub interface for the repository 'tareaDocker'. At the top, there's a header with the repository name, a 'Public' badge, and buttons for 'Pin' and 'Unwatch'. Below the header, there's a navigation bar with 'main' selected, '1 Branch', and '0 Tags'. A search bar 'Go to file' and buttons 'Add file' and 'Code' are also present. The main content area shows a commit by 'emiliozv' titled 'Creación de carpetas' with a commit hash '4be2a88' and '5 minutes ago'. Below the commit, there's a table listing three files: 'Ejercicio1', 'Ejercicio2', and 'Ejercicio3', each with a description 'Creación de carpetas' and a timestamp '5 minutes ago'.

Creación de las ramas

Creo las 3 ramas y cambio a ellas cuando lo necesite. Ejemplo con rama `ejercicio1`:

```
git branch ejercicio1
git switch ejercicio1
```

4. Ejercicio 2 - Docker Compose

Breve explicación de la aplicación

FileBrowser es una aplicación muy ligera que permite gestionar y visualizar archivos y directorios desde un navegador web, como si fuera un explorador de archivos, pudiendo subir, descargar, renombrar y eliminar archivos, entre otras funciones.

compose.yaml

Voy a utilizar un volumen y un bind-mount para ver ambas opciones. La configuración estará en un volumen y los datos en un bind-mount.

```
version: "3"

services:
  filebrowser:
    image: hurlenko/filebrowser:latest
    container_name: filebrowser
    ports:
      - "8080:8080"
    volumes:
      - file_config:/config          # volumen gestionado por Docker
      - ./data:/data                # Bind mount (carpeta local)
    restart: unless-stopped

volumes:
  file_config:
```

| Formación > DAW > 02 CURSO > Despliegue > Tareas > Evaluables > T3 Docker > repoTareaDocker > Ejercicio2 > | | | |
|--|------------------|---------------|------|
| Name | Date modified | Type | Size |
| tarea_evaluable_3_docker_ej2_Emilio_Zaera_Vida... | 01/04/2025 18:22 | File folder | |
| docker-compose.yaml | 01/04/2025 19:45 | YAML File | 1 KB |
| tarea_evaluable_3_docker_ej2_Emilio_Zaera_Vida... | 01/04/2025 19:46 | Markdown File | 5 KB |

D:\TRABAJO\Formación\DAW\02 CURSO\Despliegue\Tareas\Evaluables\T3 Docker\repoTareaDocker\Ejercicio2\docker-compose.yaml - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

```
1  version: "3"
2
3  services:
4    filebrowser:
5      image: hurlenko/filebrowser:latest
6      container_name: filebrowser
7      ports:
8        - "8080:8080"
9      volumes:
10       - file_config:/config          # Volumen gestionado por Docker
11       - ./data:/data                # Bind mount (carpeta local)
12      restart: unless-stopped
13
14  volumes:
15    file_config:
```

`restart: unless-stopped` hace que el contenedor se reinicie automáticamente si se apaga por error o al reiniciar el equipo, **pero no se reinicia si el administrador lo detiene manualmente.**

Despliegue de FileBrowser

```
docker compose up -d
```

```
C:\Windows\System32\cmd.exe
...Tareas\Evaluables\T3 Docker\repoTareaDocker\Ejercicio2>docker compose up -d
[+] Running 4/4
  filebrowser Pulled
  bbe6e0b5af75 Pull complete
  7b276281f0bb Pull complete
  1f3e46996e29 Pull complete
[+] Running 2/2
  Network ejercicio2_default Created
  Container filebrowser Started
```

Containers

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

Container CPU usage ⓘ
0.00% / 3200% (32 CPUs available)

Container memory usage ⓘ
5.59MB / 30.58GB

Show charts

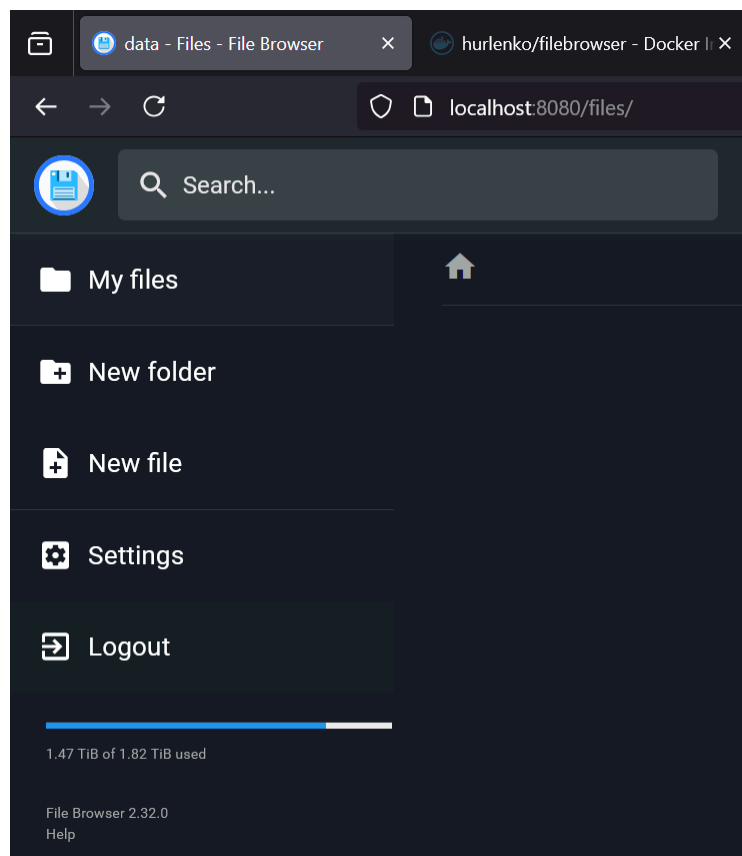
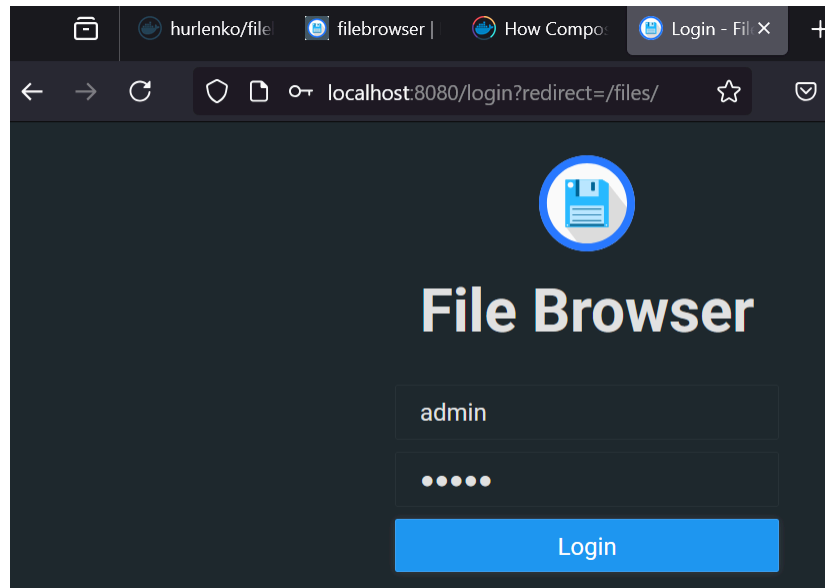
☒ Only show running containers

| <input type="checkbox"/> | Name | Container ID | Image | Port(s) | CPU (%) | Last start | Actions |
|--------------------------|---------------|--------------|-------|---------|---------|------------|---------|
| <input type="checkbox"/> | > ejercicio2 | - | - | - | 0% | 3 minute | |

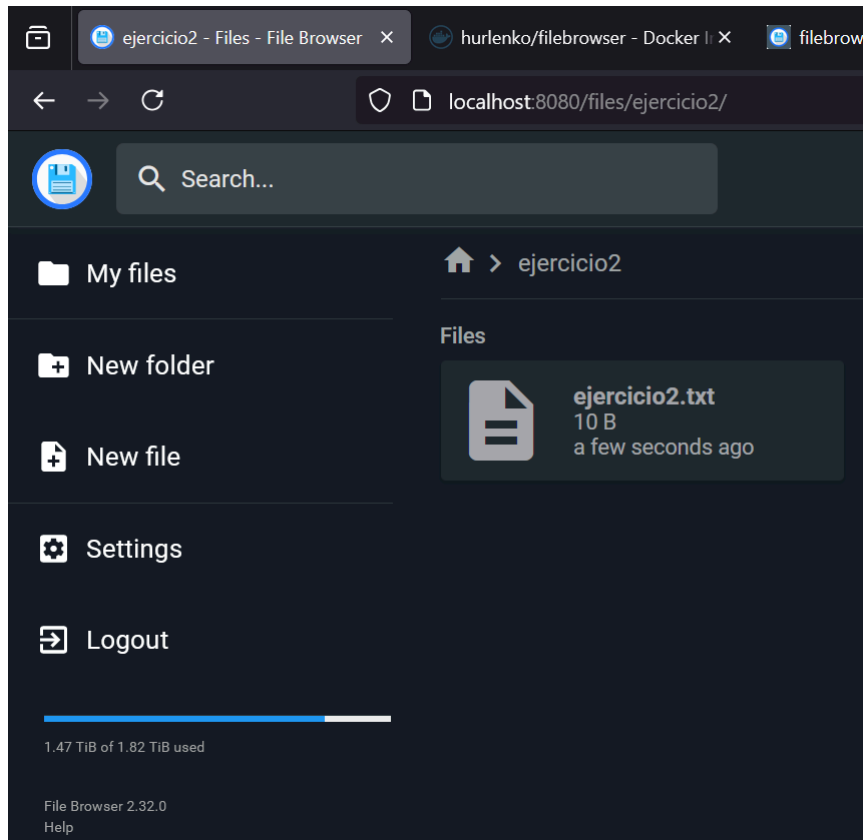
Aunque en Docker Desktop aparezca el `ejercicio2` en el nombre del contenedor, realmente es `filebrowser`, tal como indicamos el el `yaml`. Lo que se ve en **Docker Desktop** (el nombre de la carpeta o el proyecto) es solo un **nombre de grupo** que pone la interfaz para organizar los servicios, pero el contenedor real sí tiene el nombre que yo configuré.

A continuación, desde el navegador entramos en la interfaz gráfica de la aplicación:

```
http://localhost:8080
```



Creo una carpeta `ejercicio2` y un fichero `ejercicio2.txt` con contenido:



Carpeta y volumen creados

```
docker inspect filebrowser
```

```
"Mounts": [
  {
    "Type": "bind",
    "Source": "/var/lib/docker/volumes/ejercicio2_file_config/_data",
    "Destination": "/data",
    "Mode": "rw",
    "RW": true,
    "Propagation": "rprivate"
  },
  {
    "Type": "volume",
    "Name": "ejercicio2_file_config",
    "Source": "/var/lib/docker/volumes/ejercicio2_file_config/_data",
    "Destination": "/config",
    "Driver": "local",
    "Mode": "z",
    "RW": true,
    "Propagation": ""
  }
],
```

Como podemos ver en la captura anterior, en la sección "Mounts" vemos el bind-mount para los datos y el volumen para la configuración.

Directorio del bind-mount para los datos:

| Organize New Open Select | | | |
|--|------------------|---------------|------|
| Formación > DAW > 02 CURSO > Despliegue > Tareas > Evaluables > T3 Docker > repoTareaDocker > Ejercicio2 | | | |
| Name | Date modified | Type | Size |
| data | 01/04/2025 20:02 | File folder | |
| tarea_evaluable_3_docker_ej2_Emilio_Zaera_Vida... | 01/04/2025 20:13 | File folder | |
| docker-compose.yaml | 01/04/2025 19:50 | YAML File | 1 KB |
| tarea_evaluable_3_docker_ej2_Emilio_Zaera_Vida... | 01/04/2025 20:14 | Markdown File | 7 KB |

Como estoy en windows, usando Docker Desktop, el volumen vive dentro de la máquina virtual de Docker, que no es visible desde el explorador de windows. Sin embargo, en Docker Desktop - Volumes, vemos lo siguiente:

PERSONAL

Search

Ctrl+K

?

🔔

📧

⚙️

⋮

E

—

Ask Gordon BETA

Containers

Images

Volumes

Builds

Docker Hub

Docker Scout

Extensions

Volumes

Give feedback

Manage your volumes, view usage, and inspect their contents. [Learn more](#)

Search

≡

📄

| | Name ↑ | Created | Size | Actions |
|--|------------------------|----------------|-------|---------|
| | ejercicio2_file_config | 29 minutes ago | 64 kB | 📄 🗑️ |

📁

ejercicio2_file_config

📄

In use

Created 30 minutes ago

🗑️

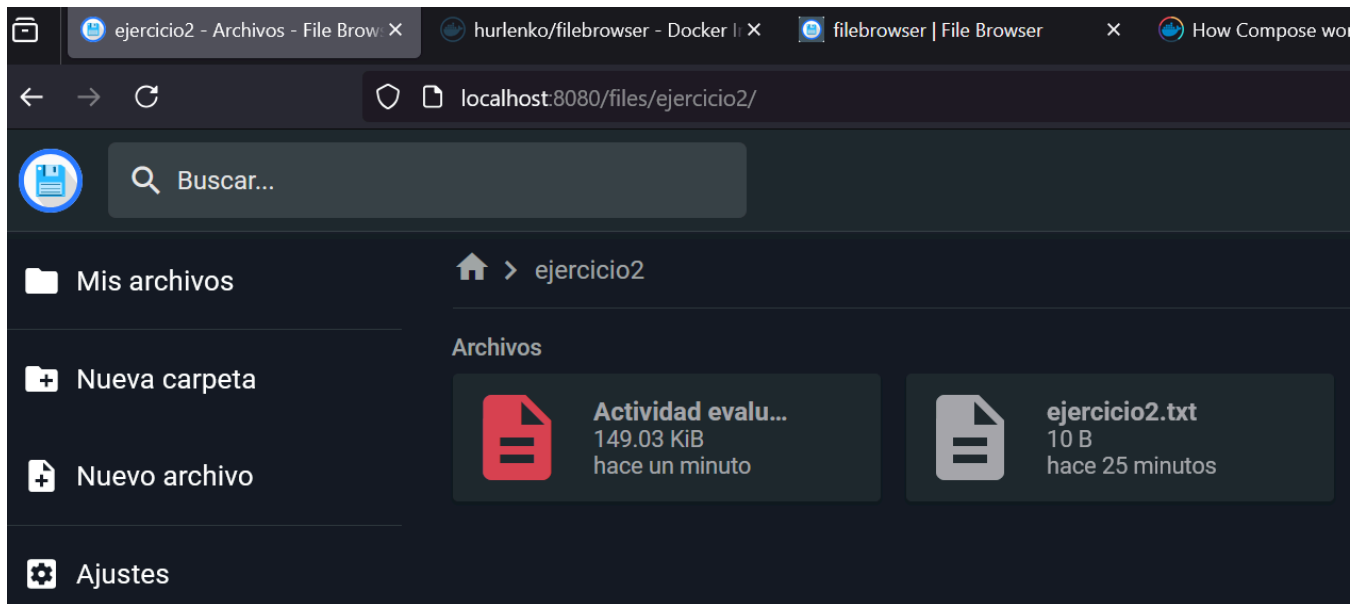
Stored data

Container in-use

Exports

| Name ↑ | Size | Last modified |
|------------------|-------|----------------|
| 📄 filebrowser.db | 64 kB | 28 minutes ago |

Subida de un fichero y cambio de idioma en FileBrowser



Como podemos observar en la captura de pantalla, se ha subido el fichero `pdf` del enunciado de la tarea y ya se ve la interfaz en español.