

# Memoria P1

## Sistemas informáticos I

Práctica realizada por:

Ignacio González Porras

Jorge Ibarreta Gallego

### Introducción

En la realización de la práctica se han tratado temas relacionados con los sistemas distribuidos y los microservicios. Hemos empezado a manejar la biblioteca Quart, que permite la ejecución de programas asíncronos basados en peticiones. Además, hemos aprendido a desarrollar y ejecutar sistemas con ayuda de Docker, que automatiza la ejecución y la comunicación entre procesos.

### Guía de ficheros

Los ficheros que se han utilizado están divididos principalmente en 3 directorios.

El primero, es el directorio /user el cual contiene, el programa user.py, que es el que crea y gestiona los usuarios de la librería, un directorio /users, que almacena ficheros de texto con la información que identifica a cada usuario, y los ficheros Dockerfile y requirements.txt, necesarios para la ejecución con Docker.

El segundo directorio, /file que tiene, el programa file.py, encargado de la gestión de ficheros relacionados con cada usuario, un directorio /libraries, donde guarda directorios relacionados a cada usuario que almacenan los ficheros de cada uno, y los ficheros Dockerfile y requirements.txt para la ejecución con Docker.

Por último, el directorio /client tiene, a parte de los ficheros Dockerfile y requirements.txt para la ejecución con Docker, el programa client.py que actual como test de funcionamiento para comprobar que la funcionalidad de user.py y file.py sea correcta.

En el directorio principal encontramos el fichero docker-compose.yml, que nos permite unificar el funcionamiento de los programas y ejecutarlos al mismo tiempo con los comandos "docker compose build" y "docker compose up"

## Ejecución

Para ejecutar el proyecto, la única condición es activar el modo rootless en la terminal. Una vez tengamos esto, desde el directorio principal ejecutamos "docker compose build", que se encargará de descargar las librerías necesarias y resolver todas las dependencias. Deberíamos ver una pantalla similar a esta:

```
ignacioglezz@ignacioglezz:~/Escritorio/UNIVERSIDAD/So_ño/SI1/SI1/PI$ docker compose build
[+] Building 18.4s (24/24) FINISHED                                docker:default
=> [file internal] load build definition from Dockerfile          0.1s
=> => transferring dockerfile: 186B                               0.0s
=> [file internal] load .dockerignore                             0.1s
=> => transferring context: 2B                                     0.0s
=> [user internal] load build definition from Dockerfile          0.1s
=> => transferring dockerfile: 186B                               0.0s
=> [user internal] load .dockerignore                             0.1s
=> => transferring context: 2B                                     0.0s
=> [client internal] load metadata for docker.io/library/python:3.9-slim 1.7s
=> [client 1/4] FROM docker.io/library/python:3.9-slim@sha256:49f94609e5a997dc16086a66ac9664591854031d48e375945a9dbf4d1d53abbc 6.0s
=> => resolve docker.io/library/python:3.9-slim@sha256:49f94609e5a997dc16086a66ac9664591854031d48e375945a9dbf4d1d53abbc 0.0s
=> => sha256:4c0965d3919510b506d8856ebc050a96e996c7dae96e4fb420882dbe7e037e67 3.51MB / 3.51MB 0.5s
=> => sha256:fdeec85abbad3878f2008f9445f15a19a5a224d1b7e7715ac6b923072333e57 14.74MB / 14.74MB 1.9s
=> => sha256:49f94609e5a997dc16086a66ac9664591854031d48e375945a9dbf4d1d53abbc 10.41kB / 10.41kB 0.0s
=> => sha256:93ab151da4e5310ea79c4ecf306ece628262b86a4d7a49cc601664f19fe44e36 1.75kB / 1.75kB 0.0s
=> => sha256:9d0cb7037cd8e90893e5f430ce4c048a872511e414580c76416752dad0a0351 5.20kB / 5.20kB 0.0s
=> => sha256:302e3ee498053a7b5332ac79e8efebec16e900289fc1ecd1c754ce8fa047fcab 29.13MB / 29.13MB 3.2s
=> => sha256:62a08b8dd4f53ad5493dabf2af00ccde91abb3771fb2187040bcf2fe94a7ced7 248B / 248B 0.7s
=> => extracting sha256:302e3ee498053a7b5332ac79e8efebec16e900289fc1ecd1c754ce8fa047fcab 1.2s
=> => extracting sha256:4c0965d3919510b506d8856ebc050a96e996c7dae96e4fb420882dbe7e037e67 0.1s
=> => extracting sha256:fdeec85abbad3878f2008f9445f15a19a5a224d1b7e7715ac6b923072333e57 0.7s
=> => extracting sha256:62a08b8dd4f53ad5493dabf2af00ccde91abb3771fb2187040bcf2fe94a7ced7 0.0s
=> [user internal] load build context                             0.1s
=> => transferring context: 4.43kB                                 0.0s
=> [file internal] load build context                             0.1s
```

(captura incompleta para no ocupar espacio de más)

Una vez completada la construcción de Docker, podemos proceder a ejecutarlo con "docker compose up", para ver que su funcionamiento es correcto y ver una pantalla como esta:

```
ignacioglezz@ignacioglezz:~/Escritorio/UNIVERSIDAD/So_ño/SI1/SI1/PI$ docker compose up
[+] Running 4/3
✔ Network p1_default Created 0.1s
✔ Container p1-file-1 Created 0.0s
✔ Container p1-user-1 Created 0.0s
✔ Container p1-client-1 Created 0.0s
Attaching to client-1, file-1, user-1
user-1 | [2024-10-11 10:06:22 +0000] [1] [INFO] Running on http://0.0.0.0:5050 (CTRL + C to quit)
file-1 | [2024-10-11 10:06:22 +0000] [1] [INFO] Running on http://0.0.0.0:5051 (CTRL + C to quit)
user-1 | * Serving Quart app 'libreria_SI1_users'
user-1 | * Debug mode: False
user-1 | * Please use an ASGI server (e.g. Hypercorn) directly in production
user-1 | * Running on http://0.0.0.0:5050 (CTRL + C to quit)
user-1 | [2024-10-11 10:06:25 +0000] [1] [INFO] 172.19.0.4:41024 POST /user/antonio 1.1 200 94 2987
user-1 | [2024-10-11 10:06:25 +0000] [1] [INFO] 172.19.0.4:41034 GET /get_user_uid/antonio 1.1 200 47 605
user-1 | [2024-10-11 10:06:25 +0000] [1] [INFO] 172.19.0.4:41050 POST /user/antonio 1.1 200 19 522
user-1 | [2024-10-11 10:06:25 +0000] [1] [INFO] 172.19.0.4:41054 GET /get_user_uid/antonio 1.1 200 19 532
file-1 | * Serving Quart app 'libreria_SI1_files'
file-1 | * Debug mode: False
file-1 | * Please use an ASGI server (e.g. Hypercorn) directly in production
file-1 | * Running on http://0.0.0.0:5051 (CTRL + C to quit)
file-1 | [2024-10-11 10:06:25 +0000] [1] [INFO] 172.19.0.4:39174 POST /create_file 1.1 200 45 1842
file-1 | [2024-10-11 10:06:25 +0000] [1] [INFO] 172.19.0.4:39188 POST /create_file 1.1 200 45 521
file-1 | [2024-10-11 10:06:25 +0000] [1] [INFO] 172.19.0.4:39200 POST /create_file 1.1 200 19 470
file-1 | [2024-10-11 10:06:25 +0000] [1] [INFO] 172.19.0.4:39210 GET /listar_documentos 1.1 200 65 653
user-1 | [2024-10-11 10:06:25 +0000] [1] [INFO] 172.19.0.4:41062 POST /user/biblio_vacia 1.1 200 94 1142
file-1 | [2024-10-11 10:06:25 +0000] [1] [INFO] 172.19.0.4:39222 GET /listar_documentos 1.1 200 68 565
```

Más abajo encontraremos la salida de los test realizados en client.py que nos permitiran ver si el funcionamiento de nuestro sistema basado en microservicios es correcto:

```
client-1 | [2021-10-22 10:00:13.100000] [1] [INFO] 127.0.0.1:43212 -> 127.0.0.1:43212 /api/cliente_usuario/127.0.0.1:43212
client-1 |
client-1 | >>> EMPIEZA EL TEST DE user.py <<<
client-1 |
client-1 | Creando usuario antonio con password 1234...
client-1 | Debe devolver un json con el uid y el token del usuario
client-1 | {'token': 'a98bb0a0-1475-52b2-9f31-3904e49f24c9', 'uid': '7b47602c-11d7-489d-9048-af6190494ee2'}
client-1 |
client-1 | Obteniendo UID de usuario antonio con password 1234...
client-1 | Debe devolver su UID
client-1 | {'uid': '7b47602c-11d7-489d-9048-af6190494ee2'}
client-1 |
client-1 | Intentando iniciar sesión con una contraseña incorrecta...
client-1 | Debe devolver ERROR
client-1 | {'status': 'ERROR'}
client-1 |
client-1 | Intentando obtener el UID de un usuario con una contraseña incorrecta...
client-1 | Debe devolver ERROR
client-1 | {'status': 'ERROR'}
client-1 |
client-1 | >>> TERMINA EL TEST DE user.py <<<
client-1 |
```

```
client-1 |
client-1 | >>> EMPIEZA EL TEST DE file.py <<<
client-1 |
client-1 | Creando el fichero fichero_001.txt con el contenido 'texto de prueba del fichero'
client-1 | Debe devolver OK y el nombre del fichero
client-1 | {'filename': 'fichero_001.txt', 'status': 'OK'}
client-1 |
client-1 | Creando el fichero fichero_002.txt con el contenido 'Segundo fichero de prueba'
client-1 | Debe devolver OK y el nombre del fichero
client-1 | {'filename': 'fichero_002.txt', 'status': 'OK'}
client-1 |
client-1 | Intentando crear un fichero con un token incorrecto...
client-1 | Debe devolver ERROR
client-1 | {'status': 'ERROR'}
client-1 |
client-1 | Buscando la biblioteca de Antonio desde la cuenta de Antonio:
client-1 | Debe devolver OK y una lista con los ficheros de Antonio
client-1 | {'ficheros': ['fichero_002.txt', 'fichero_001.txt'], 'status': 'OK'}
client-1 |
client-1 | Creando usuario nuevo con biblioteca vacía para que no haya ficheros en la biblioteca...
client-1 | Debe devolver ERROR
client-1 | {'Err_message': 'No hay ficheros en la biblioteca', 'status': 'ERROR'}
client-1 |
client-1 | Creando usuario ignacio con password 4321...
client-1 | Debe devolver un json con el uid y el token del usuario
client-1 | {'token': '6abda5d4-a453-5117-a354-89a24d3d955d', 'uid': 'd11473e7-3d47-4505-916b-c495c646761f'}
client-1 |
client-1 | Descargando el fichero fichero_001.txt desde la cuenta de Antonio:
client-1 | Debe devolver OK y el contenido del fichero
client-1 | {'content': 'texto de prueba del fichero', 'status': 'OK'}
client-1 |
client-1 | Descargando el fichero fichero_001.txt desde la cuenta de Ignacio:
client-1 | Debe devolver OK y el contenido del fichero y guardar el fichero en su biblioteca
client-1 | {'content': 'texto de prueba del fichero', 'status': 'OK'}
client-1 |
client-1 | Añadiendo contenido al fichero fichero_001.txt desde la cuenta de Ignacio:
client-1 | Debe devolver OK y el contenido añadido
client-1 | {'content': 'contenido añadido por Ignacio', 'status': 'OK'}
client-1 |
client-1 | Intentando añadir contenido a un fichero que no existe...
client-1 | Debe devolver ERROR
```