Ejercicio 3 - imagen con Dockerfile - Apliación web

> *Documento que resuelve el ejercicio 3 de la Tarea 3- Despliegue de Aplicaciones Web*

> *Para la realización de este ejercicio es necesario tener una cuenta en Docker Hub.*
>
> *Se puede resolver utilizando comandos, o Docker Desktop, o combinando ambos.*

Enunciado:

Necesitamos un fichero **Dockerfile** que automatice las siguientes operaciones para crear una imagen que contenga un servidor con un sitio web y un script php. Características de la imagen:

- Usa un contenedor que ejecute una instancia de la imagen `php:7.4-apache` que se llame `ejercicio3` y que sea accesible desde un navegador en el puerto 8000.
- Coloca en el directorio raíz del servicio web (`/var/www/html`) un "sitio web" donde figure tu nombre -el sitio deberá tener al menos un archivo `index.html` sencillo y un archivo `.css`
- Coloca en ese mismo directorio raíz el siguiente script `php`, llámalo `fecha.php`

```php
<?php
  setlocale(LC_TIME, "es_ES.UTF-8");
  $mes_actual = strftime("%B");
  $fecha_actual = date("d/m/Y");
  $hora_actual = date("H:i:s");
  echo "<h1>Información</h1>";
  echo "<p>Hoy es $fecha_actual</p>";
  echo "<p>El mes es: <strong>$mes_actual</strong></p>";
  echo "<p>Hora: $hora_actual</p>";
?>
```

- Ver la salida del script `fecha.php` y de la página `índex.html` en el navegador.
- Una vez creada la imagen, súbela a tu cuenta de Docker Hub:
    - Borra la imagen de tu Docker local.
    - Baja ('pull') de tu cuenta la imagen que acabas de subir.
    - Muestra las imágenes que tienes.
    - Ejecuta un contenedor usando esa imagen.

Resolución:

Bloque de código con el Dockerfile

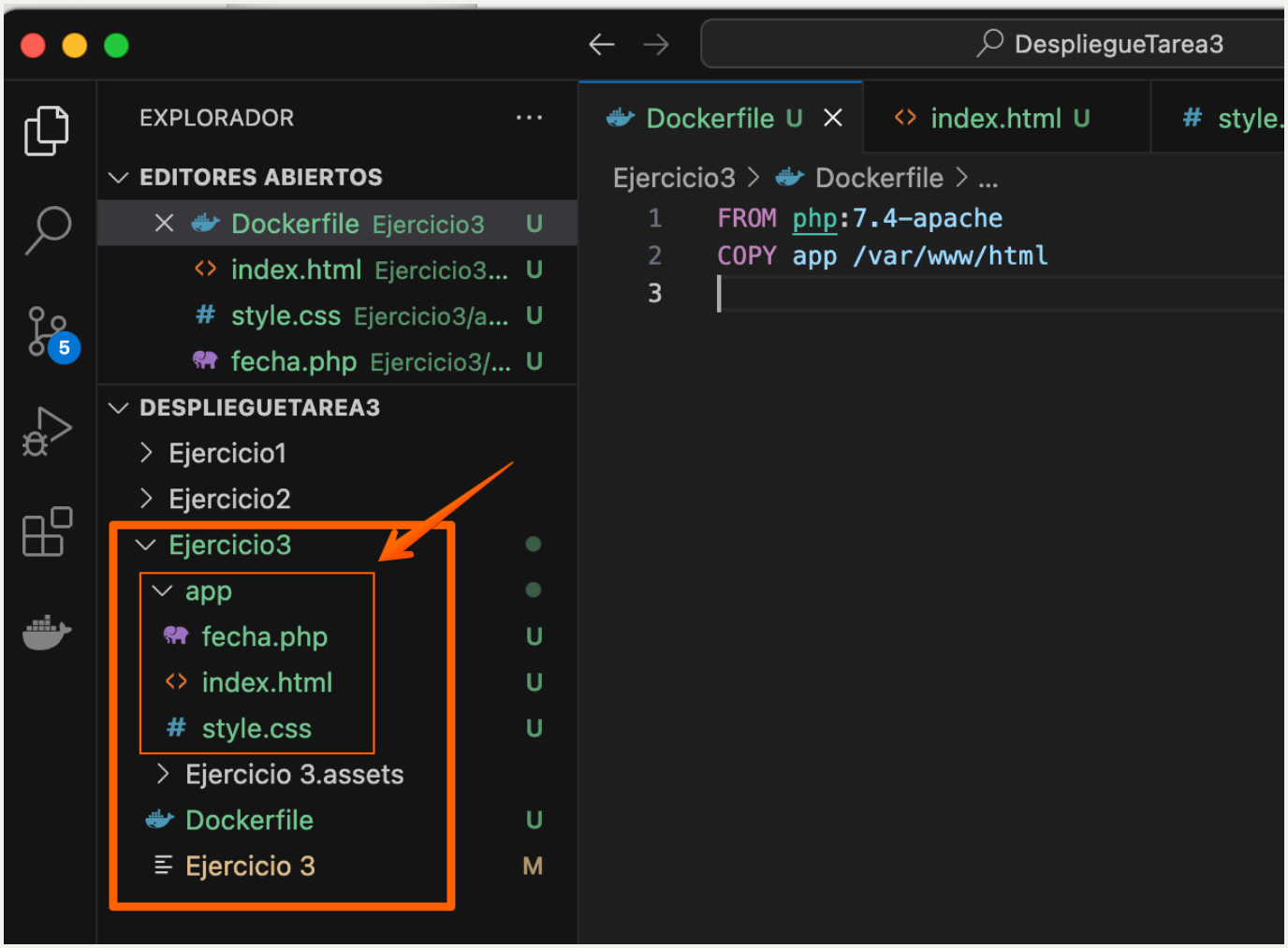
La imagen `php:7.4-apache` tiene ya instalado el servidor web y PHP.

Además, no es necesario indica el `CMD` ya que por defecto el contenedor creado a partir de esta imagen ejecutará el mismo proceso que la imagen base, es decir, la ejecución del servidor web.

El código es:

```
FROM php:7.4-apache
COPY app /var/www/html
```

En `app/` están colgados los archivos `index.html`, `style.css` y `fecha.php`, de forma que el contenido de ese directorio se copiará al `/var/www/html/`



Creación de la nueva imagen

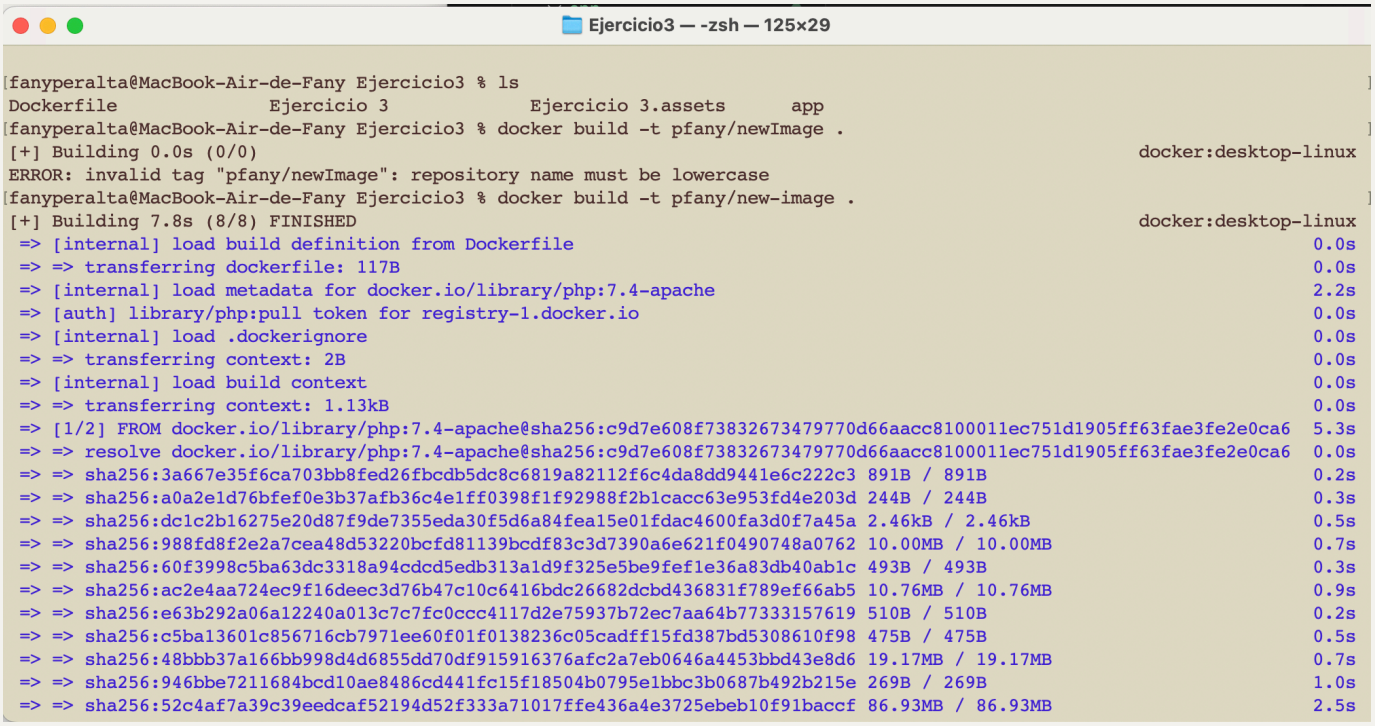
La nueva imagen sería:

```
docker build -t pfany/newImage .
```

Explicación:

- `docker build` -> inicia la construcción de la imagen.
- `-t pfany/newImage .` -> indica el nombre de la imagen y que se cree en el directorio actual donde está el archivo Dockerfile (en `Ejercicio3`).

Las capturas del proceso:



Así pues, el nombre de la imagen es `pfany/new-image`, ya que Docker no permite nombres de imágenes con mayúsculas.

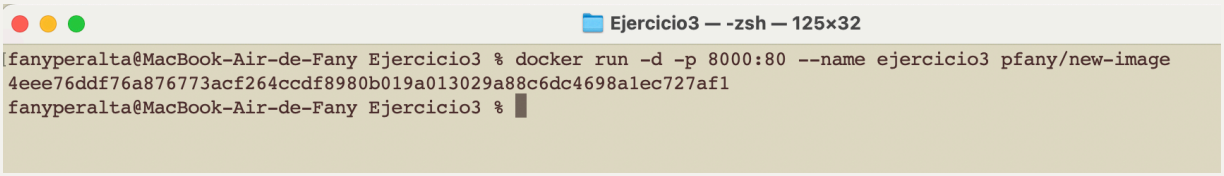
Creación del contenedor inicial

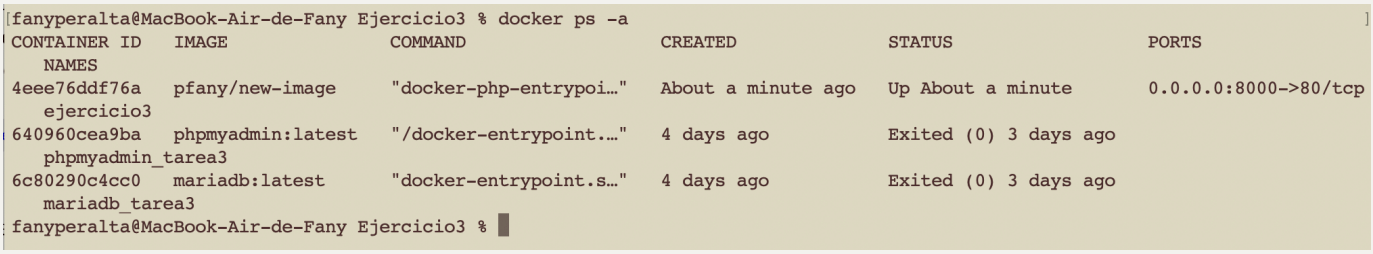
El contenedor llamado `ejercicio3` será accesible desde un navegador en el puerto 8000, con lo que a continuación de crear la imagen, creamos el contenedor llamado `ejercicio3` utilizando la magen `pfany/newImage`

```
docker run -d -p 8000:80 --name ejercicio3 pfany/new-image
```

Al indicar `-p 8000:80` se hace el mapeo de puertos: mapeo el puerto 8000 del host al puerto 80 del contenedor (donde Apache escucha por defecto).

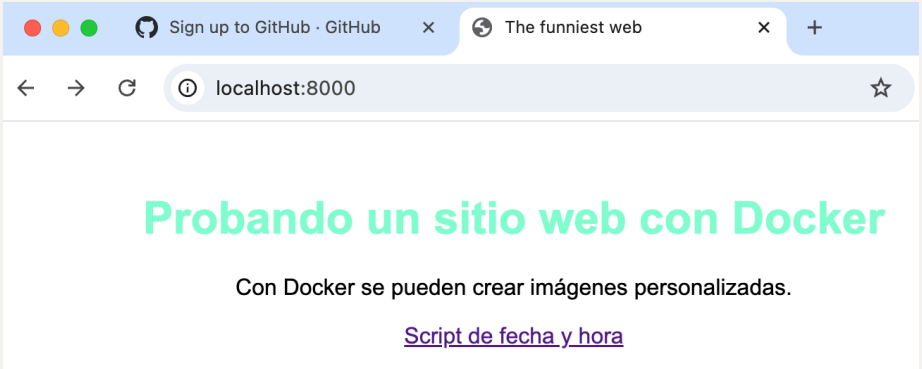
El resultado es el siguiente:



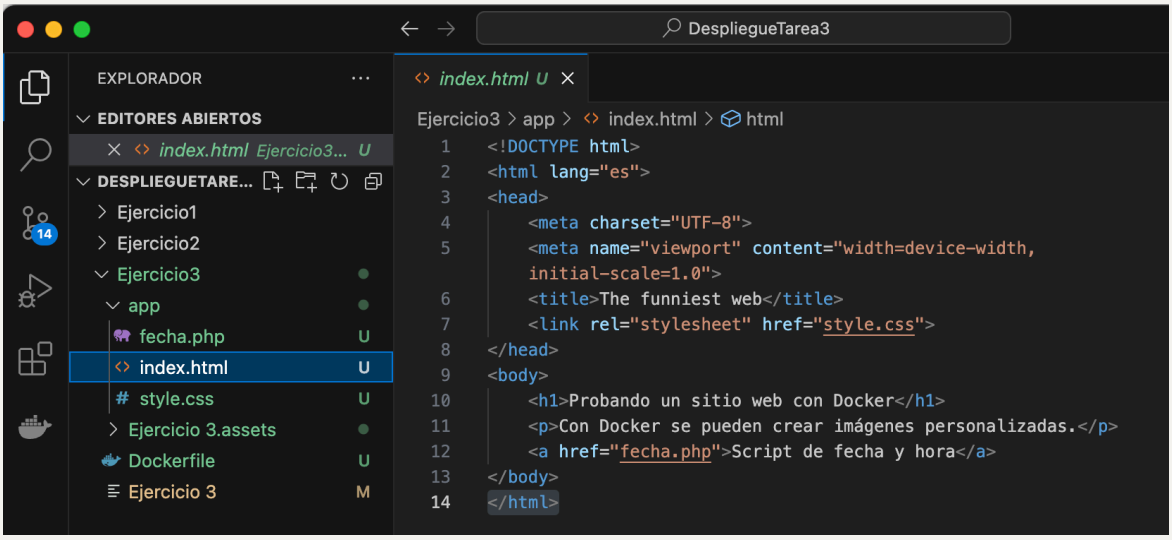


Acceso al navegador con la página `html` y con el script `php`

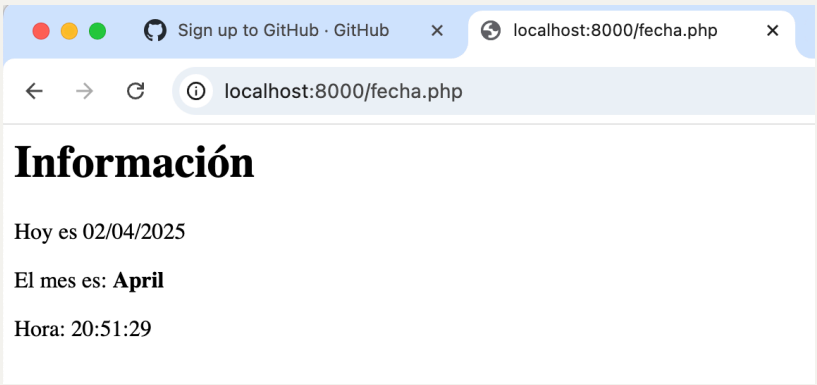
Accedo al navegador con el contenedor inicial:



que corresponde con el `index.html` muy básico creado:



Al clickar en `Script de fecha y hora`, el resultado es:



Subir la imagen creada a mi cuenta en Docker Hub