

Ejercicio 5 - imagen con Dockerfile

Ejercicio 5 - imagen con Dockerfile

PARTE I. ARRANQUE Y AUTOMATIZACIÓN DE IMAGEN EN DOCKERFILE

PARTE II. DESCARGA DE IMAGEN Y CREACIÓN DEL CONTENEDOR

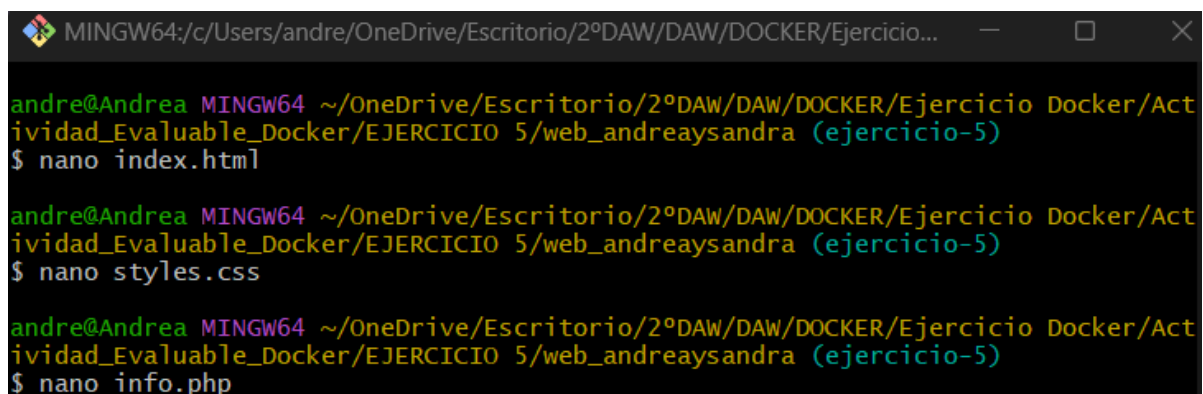
PARTE I. ARRANQUE Y AUTOMATIZACIÓN DE IMAGEN EN DOCKERFILE

En este apartado se nos pide arrancar un contenedor que ejecute una instancia de la imagen php:7.4-apache, que se llame web y que sea accesible desde un navegador en el puerto 8000, en nuestro caso, 8800.

```
docker run -d --name web -p 8800:80 php:7.4-apache
```

Creamos el sitio web y, una vez creado, hacemos lo mismo con el index, hoja de estilos y el script PHP.

```
mkdir web_andreaysandra  
cd web_andreaysandra
```



The screenshot shows a terminal window with the following commands and output:

```
andre@Andrea MINGW64 ~/OneDrive/Escritorio/2ºDAW/DAW/DOCKER/Ejercicio Docker/Actividad_Evaluable_Docker/EJERCICIO 5/web_andreaysandra (ejercicio-5)  
$ nano index.html  
  
andre@Andrea MINGW64 ~/OneDrive/Escritorio/2ºDAW/DAW/DOCKER/Ejercicio Docker/Actividad_Evaluable_Docker/EJERCICIO 5/web_andreaysandra (ejercicio-5)  
$ nano styles.css  
  
andre@Andrea MINGW64 ~/OneDrive/Escritorio/2ºDAW/DAW/DOCKER/Ejercicio Docker/Actividad_Evaluable_Docker/EJERCICIO 5/web_andreaysandra (ejercicio-5)  
$ nano info.php
```

Creamos index.html y lo editamos:

```
nano index.html  
<!DOCTYPE html>  
<html lang="es">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Probando PHP</title>  
  <link rel="stylesheet" href="styles.css">  
</head>  
<body>  
  <h1>Bienvenidos a nuestra WEB</h1>  
  <form action="script.php" method="post">  
    <h2>Hola, somos Sandra y Andrea</h2>  
    <p>Somos estudiantes de 2º de DAW y estamos probando nuestra página usa</p>  
</body>  
</html>
```

```
MINGW64/c/Users/andre/OneDrive/Escritorio/2ºDAW/DAW/DOCKER/Ejercicio Docker/Actividad_Evaluable_Docker...
GNU nano 8.2 index.html Modified
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Probando PHP</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>Bienvenidos a nuestra WEB</h1>
  <form action="script.php" method="post">
    <h2>Hola, somos Sandra y Andrea</h2>
    <p>Somos estudiantes de 2º de DAW y estamos probando nuestra página usando PHP.</p>
  </form>
</body>
</html>

File Name to Write: index.html
^G Help          M-D DOS Format  M-A Append      M-B Backup File
^C Cancel        M-M Mac Format  M-P Prepend     ^T Browse
```

Ahora creamos el css:

```
nano styles.css
body {
  font-family: Arial, sans-serif;
  background-color: #7d2181;
  text-align: center;
  margin: 0;
  padding: 0;
}

h1 {
  color: black;
  margin-top: 20px;
}

p {
  font-size: 18px;
  color: #555;
  background: black;
  padding: 10px;
  border-radius: 8px;
  display: inline-block;
```

```
MINGW64:/c/Users/andre/OneDrive/Escritorio/2ºDAW/DAW/DOCKER/Ejercicio...
GNU nano 8.2 styles.css Modified
body {
    font-family: Arial, sans-serif;
    background-color: #7d2181;
    text-align: center;
    margin: 0;
    padding: 0;
}

h1 {
    color: black;
    margin-top: 20px;
}

p {
    font-size: 18px;
    color: #555;
    background: black;
    padding: 10px;
    border-radius: 8px;
    display: inline-block;
}
File Name to Write: styles.css
^G Help          M-D DOS Format   M-A Append       M-B Backup File
^C Cancel        M-M Mac Format   M-P Prepend      ^T Browse
```

Y, finalmente, el script PHP:

```
nano info.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>";
?>
```

```
MINGW64:/c/Users/andre/OneDrive/Escritorio/2ºDAW/DAW/DOCKER/Ejercicio...
GNU nano 8.2 info.php Modified
<?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>";
?>

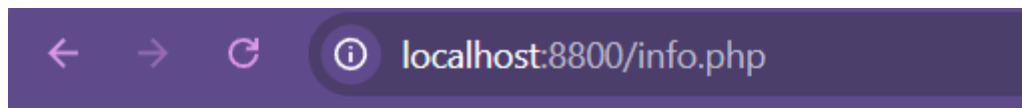
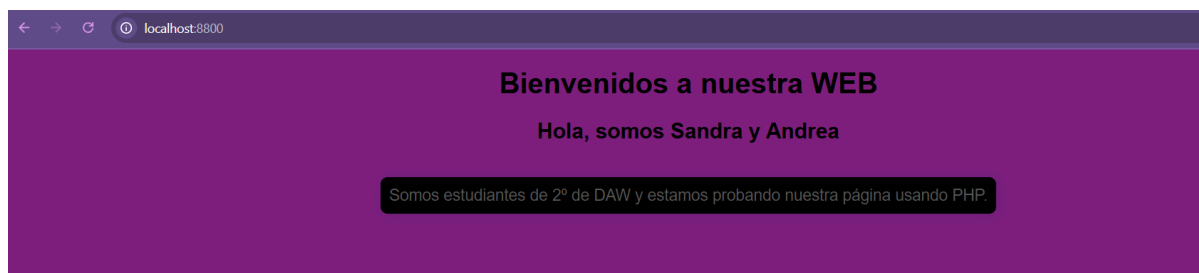
File Name to Write: info.php
^G Help          M-D DOS Format   M-A Append       M-B Backup File
^C Cancel        M-M Mac Format   M-P Prepend      ^T Browse
```

Una vez creados, copiamos los archivos en el contenedor, paso que hay que repetir cada vez que se modifique uno de ellos:

```
docker cp index.html web:/var/www/html/  
docker cp styles.css web:/var/www/html/  
docker cp info.php web:/var/www/html/
```

```
andre@Andrea MINGW64 ~/OneDrive/Escritorio/2ºDAW/DAW/DOCKER/Ejercicio Docker/Actividad_Evaluable_Docker/EJERCICIO 5/web_andreaysandra (ejercicio-5)  
$ docker cp index.html web:/var/www/html/  
Successfully copied 2.05kB to web:/var/www/html/  
  
andre@Andrea MINGW64 ~/OneDrive/Escritorio/2ºDAW/DAW/DOCKER/Ejercicio Docker/Actividad_Evaluable_Docker/EJERCICIO 5/web_andreaysandra (ejercicio-5)  
$ docker cp styles.css web:/var/www/html/  
Successfully copied 2.05kB to web:/var/www/html/  
  
andre@Andrea MINGW64 ~/OneDrive/Escritorio/2ºDAW/DAW/DOCKER/Ejercicio Docker/Actividad_Evaluable_Docker/EJERCICIO 5/web_andreaysandra (ejercicio-5)  
$ docker cp info.php web:/var/www/html/  
Successfully copied 2.05kB to web:/var/www/html/
```

Una vez realizados estos pasos, es hora de comprobar en el navegador que se están aplicando correctamente. Visitamos <http://localhost:8800/> y <http://localhost:8800/info.php>.



Información

Hoy es 19/02/2025

El mes es: **February**

Hora: 23:37:55

Ahora vamos a proceder con la automatización en Docker. Para subir la imagen a la cuenta de Docker hay que crear un fichero Dockerfile y editarlo.

```
$ touch Dockerfile  
$ nano Dockerfile
```

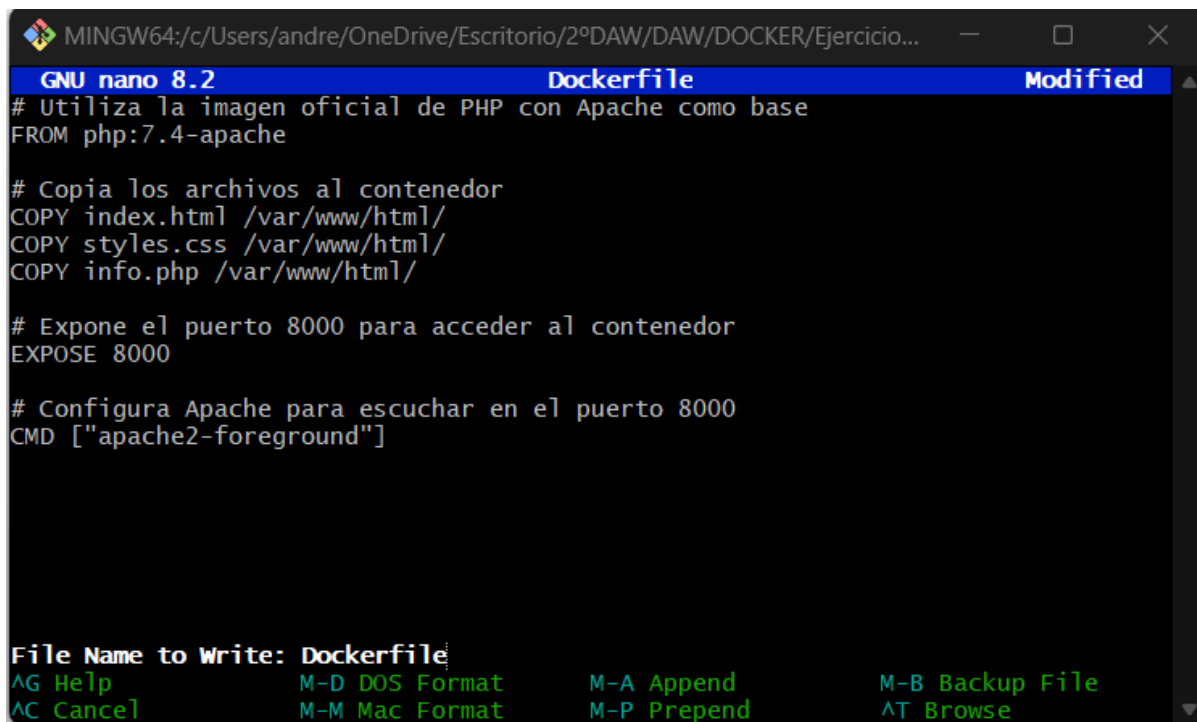
```
andre@Andrea MINGW64 ~/OneDrive/Escritorio/2ºDAW/DAW/DOCKER/Ejercicio Docker/Actividad_Evaluable_Docker/EJERCICIO 5/web_andreaysandra (ejercicio-5)  
$ touch Dockerfile  
  
andre@Andrea MINGW64 ~/OneDrive/Escritorio/2ºDAW/DAW/DOCKER/Ejercicio Docker/Actividad_Evaluable_Docker/EJERCICIO 5/web_andreaysandra (ejercicio-5)  
$ nano Dockerfile
```

```
# Utiliza la imagen oficial de PHP con Apache como base
FROM php:7.4-apache

# Copia los archivos al contenedor
COPY index.html /var/www/html/
COPY styles.css /var/www/html/
COPY info.php /var/www/html/

# Expone el puerto 8000 para acceder al contenedor
EXPOSE 8000

# Configura Apache para escuchar en el puerto 8000
CMD ["apache2-foreground"]
```



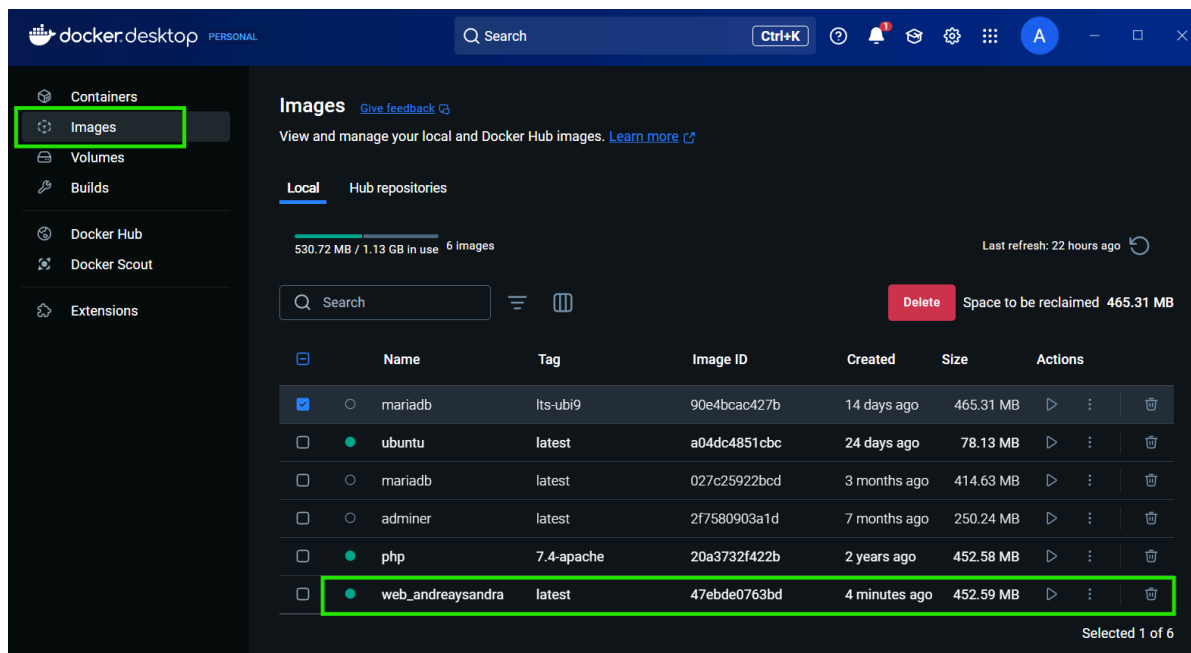
The screenshot shows a terminal window with the title 'MINGW64:/c/Users/andre/OneDrive/Escritorio/2ºDAW/DAW/DOCKER/Ejercicio...'. The editor is 'GNU nano 8.2' editing a file named 'Dockerfile'. The content of the file matches the code block above. At the bottom, there is a prompt 'File Name to Write: Dockerfile' and a menu with options: 'AG Help', 'AC Cancel', 'M-D DOS Format', 'M-M Mac Format', 'M-A Append', 'M-P Prepend', 'M-B Backup File', and 'AT Browse'.

Una vez creado y editado, ya podemos crear la imagen.

```
$ docker build -t web_andreaysandra .
```

```
andre@Andrea MINGW64 ~/OneDrive/Escritorio/2ºDAW/DAW/DOCKER/Ejercicio Docker/Actividad_Evaluable_Docker/EJERCICIO 5/web_andreaysandra (ejercicio-5)
$ docker build -t web_andreaysandra .
[+] Building 0.7s (9/9) FINISHED docker:desktop-linux
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 385B 0.0s
=> [internal] load metadata for docker.io/library/php:7.4-apache 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load build context 0.1s
=> => transferring context: 401B 0.0s
=> CACHED [1/4] FROM docker.io/library/php:7.4-apache 0.0s
=> [2/4] COPY index.html /var/www/html/ 0.1s
=> [3/4] COPY styles.css /var/www/html/ 0.1s
=> [4/4] COPY info.php /var/www/html/ 0.1s
=> exporting to image 0.2s
=> => exporting layers 0.1s
=> => writing image sha256:47ebde0763bdeac02aea9b4ddb2ab92473150c088d9af 0.0s
=> => naming to docker.io/library/web_andreaysandra 0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/02zacjfjmj3uy5v9rtm9fit7n5
```



PARTE II. DESCARGA DE IMAGEN Y CREACIÓN DEL CONTENEDOR

El siguiente paso es la descarga de la imagen y la creación del contenedor con dicha imagen. Para ello, hemos seguido los siguientes pasos:

Actualizamos el repositorio con los cambios realizados para descargarnos la imagen creada por nuestro compañero/a:

```
$git pull origin main
```

```
PS C:\Users\34615\Desktop\2º DAW\DAW\SEGUNDA EVALUACION\UT8. DOCKER\TAREA EVALUABLE DOCKER\EJERCICIO 5\web_andreaysandra> git pull origin main
```

Para asegurarnos de que la imagen se construyó correctamente, listamos las imágenes disponibles ejecutando el siguiente comando:

```
$docker images
```

```
PS C:\Users\34615\Desktop\2º DAW\DAW\SEGUNDA EVALUACION\UT8. DOCKER\TAREA EVALUABLE DOCKER\EJERCICIO 5\web_andreaysandra> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
web_andreaysandra	latest	a8442eed1771	58 seconds ago	453MB
mariadb	latest	c914c6f6d43b	6 days ago	336MB
wordpress	latest	5fb7d355caa4	9 months ago	685MB
mysql	latest	3f4db31842cd	10 months ago	256MB
postgres	16.2	d60dc4bd84c0	12 months ago	431MB
mysql	5.7	5107333e08a8	14 months ago	501MB
phpmyadmin/phpmyadmin	latest	933569f3a9f6	19 months ago	562MB

Una vez tenemos la imagen construida, ejecutamos un nuevo contenedor con el siguiente comando:

```
$docker run -d --name web_andrea -p 8800:80 web_andreaysandra
```

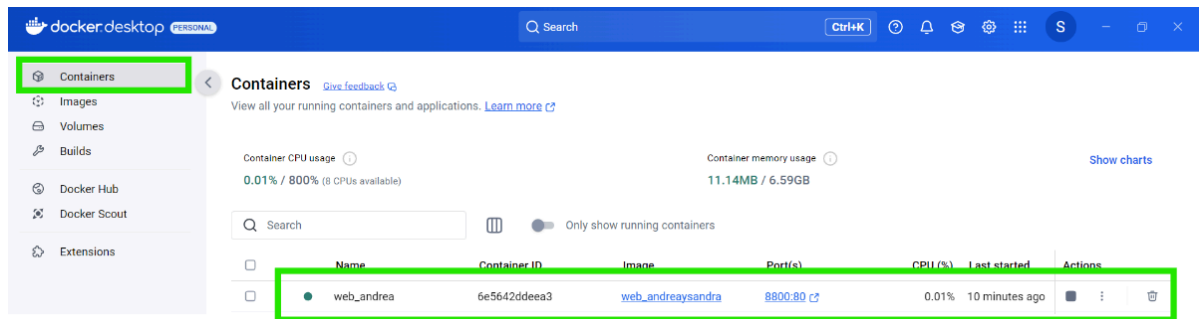
```
PS C:\Users\34615\Desktop\2º DAW\DAW\SEGUNDA EVALUACION\UT8. DOCKER\TAREA EVALUABLE DOCKER\EJERCICIO 5\web_andreaysandra> docker run -d --name web_andrea -p 8800:80 web_andreaysandra
6e5642ddea369b410d82b1fd0273db7c624f2d1539d3272a5f2a80a52567d05
```

Para asegurarnos de que el contenedor está en funcionamiento, lo verificamos con el siguiente comando:

```
$docker ps
```

```
PS C:\Users\34615\Desktop\2ºDAW\DAW\SEGUNDA EVALUACION\UT8. DOCKER\TAREA EVALUABLE DOCKER\EJERCICIO 5\web_andreaysandra> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
6e5642ddeea3   web_andreaysandra "docker-php-entrypoi..." 50 seconds ago Up 48 seconds 8000/tcp, 0.0.0.0:8800->80/tcp      web_andrea
```

Podemos abrir nuestra aplicación de Docker Desktop para comprobar que hemos creado correctamente el contenedor:



Para comprobar que los pasos han sido correctos, abrimos un navegador y pegamos la siguiente url:

- <http://localhost:8800/>

Resultado del acceso al navegador:

