

# Ejercicio 1 - trabajo con imágenes

## Ejercicio 1 - trabajo con imágenes

[Instrucciones](#)

[Solución](#)

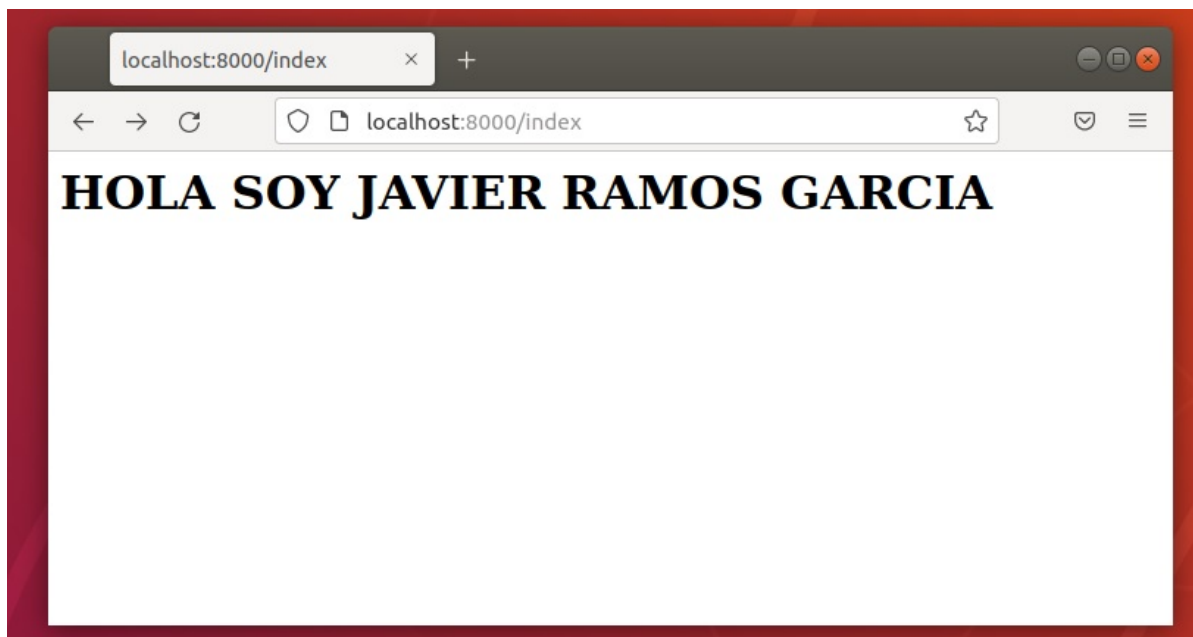
## Instrucciones

- Crear un contenedor llamado *web* con una imagen *php:7.4-apache* y cargar en el navegador un archivo *index.html* con tu nombre desde *localhost:8000*
- En ese mismo contenedor cargar en el navegador un archivo *mes.php* que muestre en pantalla el nombre del mes actual.
- Borrar el contenedor.
- Crear un contenedor llamado *bbdd* con una imagen *mariadb*.
- Establecer las variables de entorno con un usuario llamado *invitado*, con contraseña *invitado*, una contraseña *root* que sea *root* y crear una base de datos llamada *prueba*.

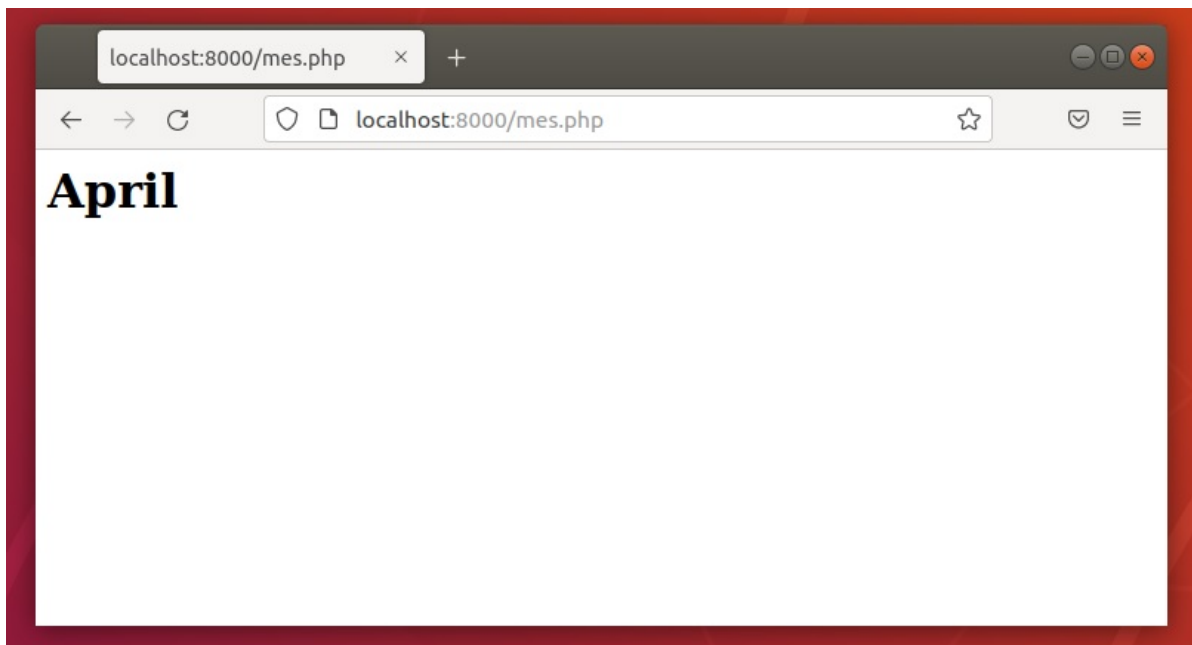
## Solución

1 - Pantallazo donde se vea la ejecución del contenedor con el archivo *index.html*.

```
docker run -d --name web -p 8000:80 php:7.4-apache
```



2 - Pantallazo donde sea la ejecución del contenedor con el archivo *mes.php*

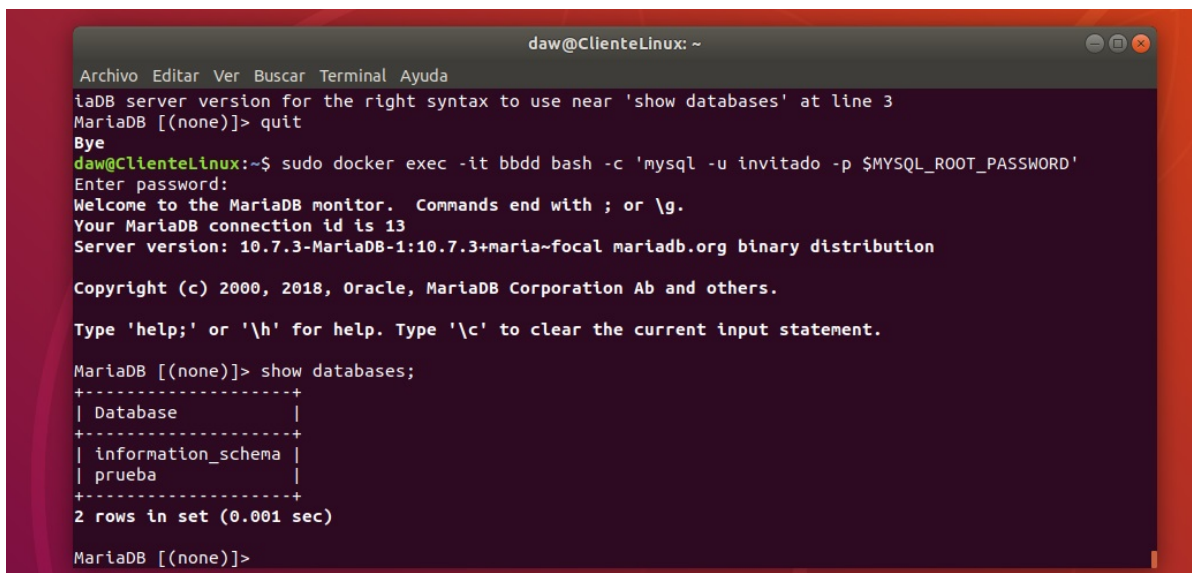


3 - Pantallazo donde se vea el tamaño del contenedor *web* con los dos ficheros creados.

```
daw@ClienteLinux:~$ sudo docker ps -a -s
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES      SIZE
28707fb9b11   mariadb:latest "docker-entrypoint.s..." 3 minutes ago  Up 3 minutes  3306/tcp       bbdd       2B (virtual 414MB)
738777ebba8f   php:7.4-apache "docker-php-entryp..." 36 minutes ago Exited (0) 23 minutes ago  web          0B (virtual 452MB)
```

La captura se hizo después de crear el contenedor de *mariadb*, por eso aparecen los dos contenedores.

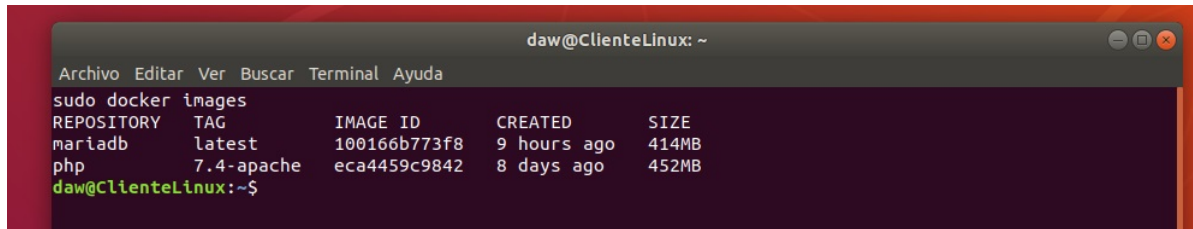
4 - Pantallazo de la consola de la Base de Datos donde se ve que hemos podido conectarnos con el usuario y base de datos creados.



5 - Pantallazo donde se comprueba que no se puede borrar la imagen de *mariadb* mientras el contenedor se está ejecutando.

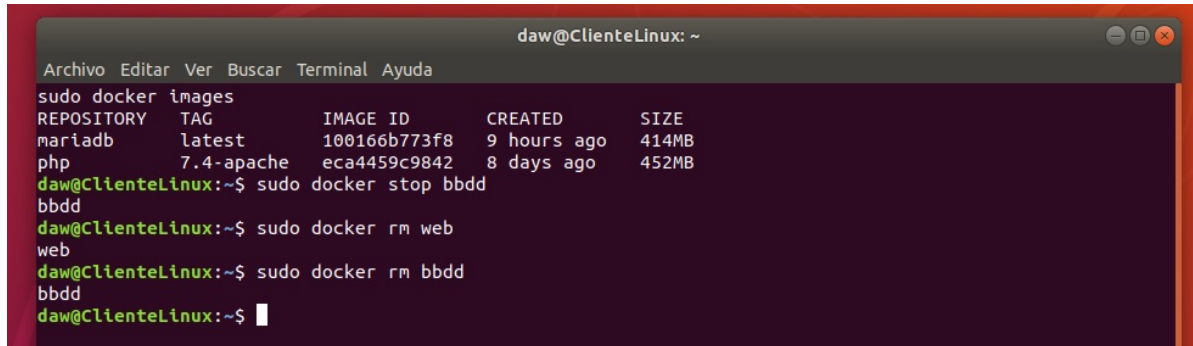
```
daw@ClienteLinux:~$ sudo docker rm bbdd
Error response from daemon: You cannot remove a running container caf446ac2f5a977ec1df9d8a971fc1d9880e81573daadec65a4085d3443d9a7f. Stop the container before attempting removal or force remove
daw@ClienteLinux:~$
```

6 - Pantallazo donde se ven las imágenes que tengo descargadas en mi ordenador.

A screenshot of a terminal window titled 'daw@ClienteLinux: ~'. The terminal shows the command 'sudo docker images' and its output, which is a table of Docker images. The table has five columns: REPOSITORY, TAG, IMAGE ID, CREATED, and SIZE. The data rows are: mariadb latest 100166b773f8 9 hours ago 414MB, and php 7.4-apache eca4459c9842 8 days ago 452MB. The prompt 'daw@ClienteLinux:~\$' is visible at the bottom.

```
daw@ClienteLinux: ~
Archivo Editar Ver Buscar Terminal Ayuda
sudo docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
mariadb latest 100166b773f8 9 hours ago 414MB
php 7.4-apache eca4459c9842 8 days ago 452MB
daw@ClienteLinux:~$
```

7 - Pantallazo donde se ve el proceso de borrado de los contenedores usados.

A screenshot of a terminal window titled 'daw@ClienteLinux: ~'. The terminal shows the command 'sudo docker images' followed by the output table from the previous screenshot. Then, the user enters 'sudo docker stop bbdd', followed by 'bbdd' on a new line. Then 'sudo docker rm web', followed by 'web' on a new line. Then 'sudo docker rm bbdd', followed by 'bbdd' on a new line. The prompt 'daw@ClienteLinux:~\$' is visible at the bottom.

```
daw@ClienteLinux: ~
Archivo Editar Ver Buscar Terminal Ayuda
sudo docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
mariadb latest 100166b773f8 9 hours ago 414MB
php 7.4-apache eca4459c9842 8 days ago 452MB
daw@ClienteLinux:~$ sudo docker stop bbdd
bbdd
daw@ClienteLinux:~$ sudo docker rm web
web
daw@ClienteLinux:~$ sudo docker rm bbdd
bbdd
daw@ClienteLinux:~$
```

Primero paramos el contenedor de la base de datos que estaba en ejecución.