



TAREA EVALUABLE SEGUNDO CUATRIMESTRE

DESPLIEGUE DE APLICACIONES WEB

David García Vincelle



Índice

Solución Ejercicio 1

1.1 Servidor Web

1.1.1 Arranca un contenedor que ejecute una instancia de la imagen `php:7.4-apache`, que se llame `servidor` y que sea accesible desde un navegador en el puerto 1234.

1.1.2 Coloca en el directorio raíz del servicio web (`/var/www/html`) de dicho contenedor un sitio web

1.1.3 Coloca en ese mismo directorio raíz un archivo llamado `cabeceras.php` que muestre todas las cabeceras de petición HTTP de la llamada actual. Ver la salida del script en el navegador. Deberás ver algo similar a esto:

1.1.4 Borramos el contenedor

1.2 Servidor de base de datos

1.2.1 Crear un contenedor con la siguientes características

1.2.2 Conéctate con el usuario `root` . Crea una tabla `tabla1` en la base de datos `base1`

1.2.3 Conéctate con el usuario `daw` . Comprueba que existe la base de datos `base1` , y la tabla `tabla1`

1.2.4 Borramos el contenedor

Solución Ejercicio 1

1.1 Servidor Web

1.1.1 Arranca un contenedor que ejecute una instancia de la imagen `php:7.4-apache`, que se llame `servidor` y que sea accesible desde un navegador en el puerto 1234.

Partimo de un entorno donde ya tenemos instalado docker. Realizamos lo solicitado con un solo comando:

```
docker run -d --name servidor -p 1234:80 php:7.4-apache
```

1.1.2 Coloca en el directorio raíz del servicio web (`/var/www/html`) de dicho contenedor un sitio web

*En la página `index.html` debe aparecer el texto: Tarea de Docker de **Nombre Apellido***

- Vamos a descargar una plantilla que a continuación personalizaremos

```
wget https://www.free-css.com/assets/files/free-css-templates/download/page289/4uhost.zip
```

- La descomprimos en el directorio de usuario y personalizamos el fichero `index.html` para que aparezca el texto: **Tarea de Docker de David García Vincelle**

En mi caso me encuentro en: `/home/cliente-david`

Para personalizar el fichero `index.html`, utilizo el editor de textos de ubuntu

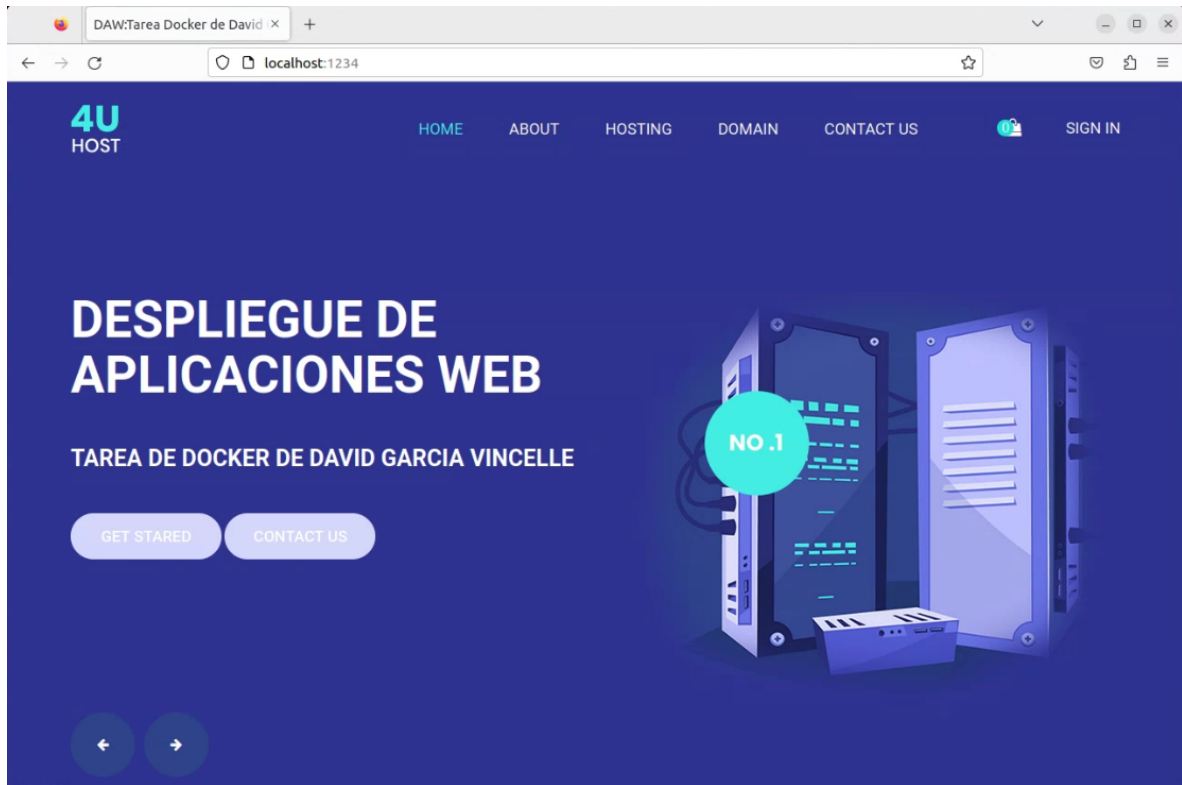
```
gedit 4uhost-html/index.html
```

- Una vez personalizado el fichero `index.html`, copiamos la carpeta `4uhost-html/`, recursivamente, en la ubicación `/var/www/html/` de nuestro `servidor`

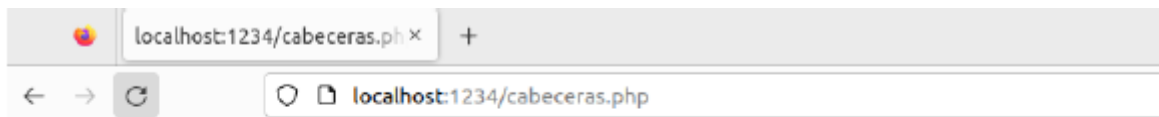
```
docker cp 4uhost-html/. servidor:/var/www/html/.
```

- Ahora **comprobamos que el sitio web está operativo** y que el `index.html` de la plantilla que descargamos se encuentra modificado con el texto que queríamos

Abrimos el navegador web Firefox y escribimos la dirección de nuestro servicio web: `localhost:1234`



1.1.3 Coloca en ese mismo directorio raíz un archivo llamado `cabeceras.php` que muestre todas las cabeceras de petición HTTP de la llamada actual. Ver la salida del script en el navegador. Deberás ver algo similar a esto:



Script Cabeceras - tarea de Inés

```
Host: localhost:1234
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/111.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
```

- Creamos un fichero llamado `cabeceras.php` e insertamos el siguiente código php:

```
<h2>Script Cabeceras - Tarea de David Garcia vincelle</h2><br>
<?php
foreach (getallheaders() as $nombre => $valor) {
    echo "$nombre: $valor\n";?><br><?php
}
```

Para crear el fichero `index.html` e implementar el código, utilizo el editor de textos de ubuntu

```
gedit cabeceras.php
```

- Una vez creado, copiamos el fichero `cabeceras.php` al contenedor, en el directorio raíz del servicio web

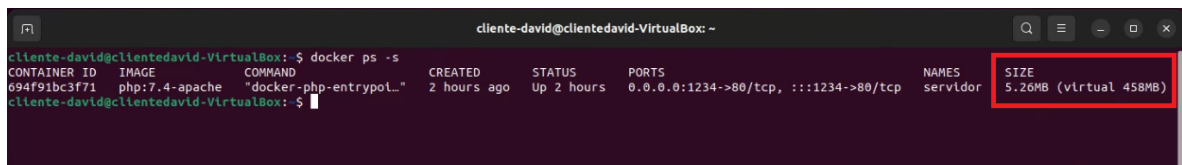
```
docker cp cabeceras.php servidor:/var/www/html/.
```

- Vemos la salida del script en el navegador



- Visualizamos el tamaño del contenedor, ahora que tenemos añadido nuestro sitio y el script php

```
docker ps -s
```



1.1.4 Borramos el contenedor

- Paramos el contenedor, después lo borramos y comprobamos que ya no hay ninguno activo

```
docker stop servidor
docker rm servidor
docker ps
```

```
cliente-david@clientedavid-VirtualBox: ~  
cliente-david@clientedavid-VirtualBox:~$ docker stop servidor  
servidor  
cliente-david@clientedavid-VirtualBox:~$ docker rm servidor  
servidor  
cliente-david@clientedavid-VirtualBox:~$ docker ps -a  
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

1.2 Servidor de base de datos

Visita la página de la imagen oficial de [mariadb](#) en [Docker Hub](#)

1.2.1 Crear un contenedor con la siguientes características

- Que se llame **bbdd** y que ejecute una instancia de la imagen **mariadb**
- Establece las variables de entorno necesarias para que:
- La contraseña de root sea **root**
- Se cree una base de datos automáticamente al arrancar que se llame **base1**
- Se cree el usuario **daw** con la contraseña **labora11**

- Para crear este contenedor, vamos a utilizar docker compose, que previamente ha de estar instalado

```
sudo apt install docker-compose
```

- Creamos un directorio llamado **mariadb**, donde entraremos y crearemos el fichero **docker-compose.yml**

```
mkdir mariadb  
cd mariadb  
gedit docker-compose.yml
```

- Una vez cargado el editor de textos de ubuntu, implementamos el siguiente código:

```
version: '3.1'  
services:  
  db:  
    container_name: bbdd  
    image: mariadb  
    restart: always  
    environment:  
      MYSQL_ROOT_PASSWORD: root  
      MYSQL_DATABASE: base1  
      MYSQL_USER: daw  
      MYSQL_PASSWORD: labora11
```

- Una vez guardado, procedemos a lanzar el script de `docker-compose` y entramos en el contenedor por medio de su consola de comandos

```
docker-compose up -d
docker exec -it bbdd bash
```

1.2.2 Conéctate con el usuario `root` . Crea una tabla `tabla1` en la base de datos `base1`

- Al estar dentro de la consola de comandos del contenedor, **ya podemos conectarnos a MariaDB** con el usuario `root`

```
mysql -u root -p
```

- Una vez conectados, creamos la tabla `tabla1` dentro de la base de datos `base1`, y **mostramos que están creadas** tanto la base de datos `base1` como la tabla `tabla1`

```
CREATE TABLE base1.tabla1 (columna1 INT NOT NULL, PRIMARY KEY (columna1));
show databases;
use base1;
show tables;
```

```
root@3f7394167284: /
mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.11.2-MariaDB-1:10.11.2+maria-ubu2204 mariadb.org binary distribution
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE TABLE base1.tabla1 (columna1 INT NOT NULL,PRIMARY KEY (columna1));
Query OK, 0 rows affected (0.022 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| base1    |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
5 rows in set (0.001 sec)

MariaDB [(none)]> use base1;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [base1]> show tables;
+-----+
| Tables_in_base1 |
+-----+
| tabla1          |
+-----+
1 row in set (0.001 sec)

MariaDB [base1]>
```

1.2.3 Conéctate con el usuario `daw` . Comprueba que existe la base de datos `base1` , y la tabla `tabla1`

- Vamos a desconectar de MariaDB

```
exit
```

- Volveremos a conectar, pero ésta vez con el usuario `daw`

```
mysql -u daw -p
```

- Comprobamos que éste usuario tiene acceso a la base de datos y a la tabla que mostramos anteriormente

```
show databases;
use base1;
show tables;
```



```
root@3f7394167284: /
root@3f7394167284:/# mysql -u daw -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.11.2-MariaDB-1:10.11.2+maria~ubu2204 mariadb.org binary distribution
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| base1 |
| information_schema |
+-----+
2 rows in set (0.001 sec)

MariaDB [(none)]> use base1;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [base1]> show tables;
+-----+
| Tables_in_base1 |
+-----+
| tabla1 |
+-----+
1 row in set (0.001 sec)

MariaDB [base1]>
```

1.2.4 Borramos el contenedor

- Primero vamos a salir de mariaDB

```
exit
```

- A continuación salimos de la consola de comandos del contenedor y lo paramos

```
exit
docker stop bbdd
```

- Intentamos borrar la imagen mariadb mientras el contenedor bbdd está creado

```
docker rmi mariadb
```

No nos dejará

- eliminamos el contenedor

```
docker rm bbdd
```

```
cliente-david@clientedavid-VirtualBox: ~/marladb
cliente-david@clientedavid-VirtualBox:~/marladb$ docker stop bddd
bddd
cliente-david@clientedavid-VirtualBox:~/marladb$ docker rmi marladb
Error response from daemon: conflict: unable to remove repository reference "marladb" (must force) - container 93146cda3264 is using its referenced image 4a632f970181
cliente-david@clientedavid-VirtualBox:~/marladb$ docker rm bddd
bddd
cliente-david@clientedavid-VirtualBox:~/marladb$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS          NAMES
cliente-david@clientedavid-VirtualBox:~/marladb$
```