

# Tarea 1

## DAW - 23/24

PROFESOR: Tacio Camba Espí  
tacio@ciclosmontecastelo.com

---

### Instrucciones

- Fecha de **publicación**:
  - 15:00h del Jueves 07 de marzo de 2024
- Entregar en **fichero comprimido apellidos-nombre\_t1.zip** (todo en minúsculas) a través de la **tarea** creada a tal efecto en Google **Classroom**
- Fecha de **entrega**:
  - 23:59 del jueves 14 de marzo de 2024
- Fecha **límite de entrega (PENALIZADAS)**:
  - 23:59 del domingo 17 de marzo de 2024
- El plagio se castigará con una calificación de 0 tanto para el que copia como para el que es copiado

## 1. Docker. Imágenes (1.5 ptos)

1. (0.5 pts) Descarga del registro de imágenes por defecto el servidor web **apache httpd** en su **versión 2.4 bullseye**. Toma nota del tamaño que ocupa la imagen en disco
2. (0.25 pts) ¿Cuál es el **ID** de la imagen que acabas de descargar? ¿Y su **hash** o **digest**? ¿Qué relación tienen entre sí?
3. (0.5 pts) Descarga ahora la versión **3.19 alpine** de la imagen ¿Cuánto ocupa en disco? ¿Cuántas veces más pequeña o grande es la imagen de alpine respecto de la de bullseye? ¿A qué se debe esta diferencia?
4. (0.25 pts) Elimina ambas imágenes del registro local

Demuestra que has realizado cada apartado con capturas de pantalla de la terminal

## 2. Docker. Containers y redes (3.25 ptos)

1. (0.25 pts) Descarga la última versión de la imagen **containous/whoami**.
2. (0.5 pts) Levanta un contenedor llamado **testserver1** a partir de la imagen anterior asignando un **puerto aleatorio** del sistema host (**opción -P**). Ejecútalo en segundo plano
3. (0.5 pts) Inspecciona el contenedor anterior para determinar el puerto del host que se ha asignado. Accede a la URL <http://localhost:PUERTO> desde tu navegador
4. (0.75 pts) Crea un contenedor idéntico al anterior con el nombre **testserver2**, pero en esta ocasión enlaza los puertos **8080** y **4433** del **host** a los puertos **80** y **443** del **contenedor**.

Accede a las URLs <http://localhost:8080> y <https://localhost:4433> desde tu navegador

5. (0.5 pts) Levanta un contenedor a partir de la imagen [wbitt/network-multitool](https://hub.docker.com/r/wbitt/network-multitool) con el nombre **nettools**. No es necesario que publiques ningún puerto en el host
6. (0.25 pts) Utilizando el comando **docker exec -it nettools bash** accede al contenedor **nettools** y ejecuta el comando **ping testserver1** ¿Obtienes respuesta? En caso negativo, ¿por qué no?
7. (0.25 pts) Repite el punto anterior pero en esta ocasión haz ping a la IP del contenedor **testserver1** (tendrás que utilizar el comando **docker inspect** para conocer la IP) ¿Obtienes respuesta? ¿Por qué?
8. (0.25 pts) Crea una red llamada **testnet** y une los contenedores **testserver1** y **nettools** a la misma. Repite el **apartado 6**. Explica qué ocurre en esta ocasión

Demuestra que has realizado cada apartado con capturas de pantalla de la terminal y del navegador (en las que se aprecie bien la URL para las capturas de navegador)

### 3. Docker. Dockerfile y volúmenes (3 ptos)

1. Crea un **Dockerfile** que basado en la **última versión de alpine** de **httpd** que sirva una página web de muestra (no importa el contenido)
2. (0.25 pts) Construye la imagen a partir del Dockerfile anterior con el tag **{tunombre}/web5:latest**
3. (0.75 pts) Levanta un contenedor de la imagen anterior llamado **web5test** que publique el puerto **80** en el **8080** del **host**. ¿Puedes acceder a la web en <http://localhost:8080> ? En caso negativo ¿Qué está sucediendo? ¿Cómo lo arreglarías?
4. (0.25 pts) Detén el contenedor y elimínalo. Haz un cambio cualquiera en el fichero html de la web y vuelve a levantarlo (sin reconstruir la imagen). ¿Se aplicaron los cambios en la web? En caso negativo ¿qué ha sucedido?
5. (0.75 pts) A continuación levanta un nuevo contenedor con el mismo nombre (**web5test**) pero en esta ocasión enlaza o mapea la ruta del host que contiene los estáticos de la web con el directorio del contenedor desde el que se sirven

Este paso lo puedes realizar directamente por consola utilizando el comando **docker run** con la imagen oficial de **httpd** (es decir, puedes prescindir de la imagen **custom** construida a partir del **Dockerfile**). Comprueba que puedes acceder al sitio web que acabas de servir en **localhost**

6. (0.25 pts) Sin eliminar o detener el contenedor anterior, modifica el fichero html y verifica que el cambio se aplica en el sitio web (puede ser necesario que actualices la página) ¿Se aplica el cambio? ¿Por qué?

Demuestra que has realizado cada apartado con capturas de pantalla de la terminal y del navegador (en las que se aprecie bien la URL para las capturas de navegador). Incluye en las respuestas el Dockerfile del apartado 1 como captura de pantalla o directamente como texto