

**Duración:** 2 sesiones

La práctica se realizará de manera individual. Tiene un peso del **15%** del total de prácticas.

**Requisitos Previos:**

- Instalación de Docker y Docker Compose en el sistema.
- Conocimientos básicos de Docker, Apache, PHP, Dockerfile y Docker Compose.

## Introducción

En el dinámico mundo del desarrollo web, la capacidad de crear y manejar entornos de servidor flexibles y eficientes es crucial. Esta práctica tiene como objetivo introducir el uso de Docker, una tecnología líder en contenedores, aplicada al despliegue de servidores web con Apache y PHP. A través de esta práctica, se aprenderá cómo Docker puede simplificar el proceso de desarrollo y despliegue, asegurando consistencia entre entornos de trabajo y producción.

**Objetivos de la Práctica:**

Esta práctica se centrará en crear una infraestructura web básica utilizando contenedores Docker.

1. Aprender y aplicar conceptos fundamentales de Docker y contenedores.
2. Configurar servicios web utilizando Apache y PHP dentro de contenedores Docker.
3. Experimentar con la creación y gestión de redes Docker y asignar contenedores a estas redes.
4. Practicar la creación y uso de Dockerfile y docker-compose.yml para la configuración y despliegue de servicios.

**Descripción de la Práctica:**

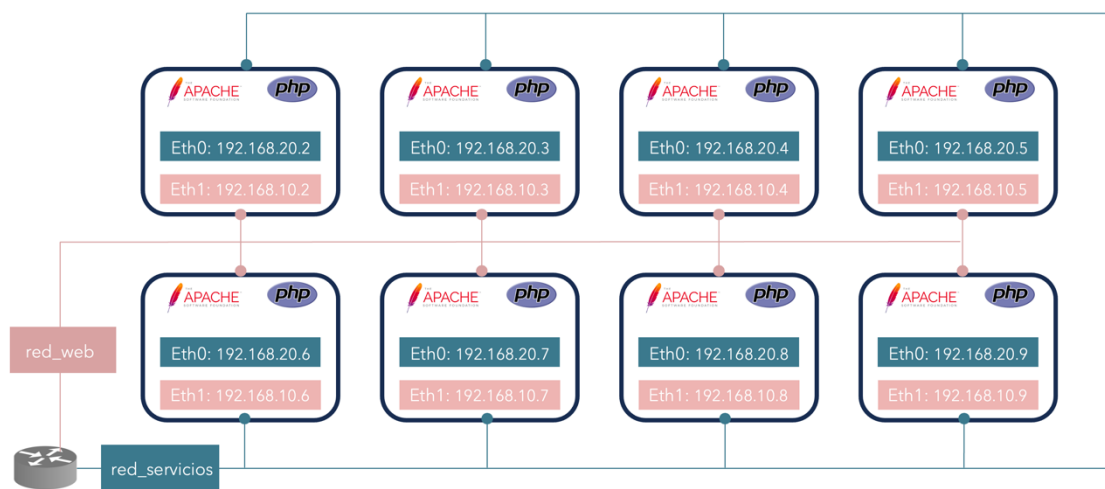
Se comenzará por configurar un entorno Docker, donde se desplegarán 8 réplicas de contenedores. Cada contenedor ejecutará un servidor web Apache con soporte PHP. Se montará un directorio local dentro de los contenedores para servir una página web sencilla, lo que permitirá entender cómo los datos pueden ser gestionados y persistidos en Docker.

Se configurarán dos redes Docker de tipo bridge, **red\_web** y **red\_servicios**, para simular un entorno de red más complejo y realista. Se asignarán contenedores a estas redes y se explorarán cómo la comunicación entre contenedores puede ser gestionada en Docker.

La práctica finalizará con la validación y pruebas de los contenedores desplegados, asegurando que cada instancia del servidor web funcione según lo esperado y sirva la página web correctamente.



## Esquema general de la práctica:



## Desarrollo:

Se pretende crear 8 réplicas de contenedores Docker que ejecuten Apache y PHP, conectados a dos redes específicas: `red_web` y `red_servicios`. Cada contenedor servirá una página web simple desde un directorio montado.

### Parte 1: Creación del directorio y archivo `index.php`

- Crear un directorio en tu máquina local llamado `web_usuarioUGR`.
- Dentro de este directorio, crear un archivo `index.php` que muestre "SWAP - nombre del usuario" y la dirección IP del servidor Apache".

### Parte 2: Creación del Dockerfile

- Crear un archivo `Dockerfile` en la raíz del proyecto llamado `DockerfileApache_usuarioUGR`.
- Usar una imagen base de Linux, instalar Apache, PHP y herramientas de red para comprobar conectividad entre máquinas.

### Parte 3: Configuración con Docker Compose

1. Crear un archivo `docker-compose.yml` con las siguientes características:
  - Crear una imagen llamada `usuarioUGR-apache-image:p1` a partir del Dockerfile `DockerfileApache_usuarioUGR`.
  - Crear 8 contenedores llamados `webX`, donde X es un número de 1 a 8 con volúmenes donde se monte el directorio "web\_usuarioUGR" en el directorio raíz de Apache en el contenedor.
  - Añadir las dos redes al contenedor, una red llamada `red_web` con dirección 192.168.10.0/24 y otra red llamada `red_servicios` con dirección 192.168.20.0/24.

## Parte 4: Verificación y Pruebas

### 1. Verificar la Configuración de los Contenedores:

- Lanzar los contenedores con **docker-compose up**
- Utilizar **docker ps** para verificar que todos los contenedores estén en ejecución.
- Comprobar que cada contenedor tiene una IP asignada en las redes **red\_web** y **red\_servicios**.
- Comprobar conectividad entre los distintos contenedores.

```
[sotillo@MBPM1deSoTillo P2 % docker compose exec web1 /bin/bash
root@9d635d4ec410:/# ping 192.168.10.3
PING 192.168.10.3 (192.168.10.3) 56(84) bytes of data.
64 bytes from 192.168.10.3: icmp_seq=1 ttl=64 time=0.209 ms
64 bytes from 192.168.10.3: icmp_seq=2 ttl=64 time=0.159 ms
64 bytes from 192.168.10.3: icmp_seq=3 ttl=64 time=0.381 ms
^C
--- 192.168.10.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2082ms
rtt min/avg/max/mdev = 0.159/0.249/0.381/0.095 ms
root@9d635d4ec410:/#
```

### 2. Probar la Página Web:

- Acceder a la página web de cada contenedor usando su dirección IP y verificar que muestra la información correcta.



## Uso de Inteligencia Artificial Generativa.

Para cada práctica es OBLIGATORIO usar herramientas de IA generativa (ChatGPT, Copilot u otras) e incluir enlace al chat/prompt utilizado. También se debe analizar y justificar el resultado que proporciona la herramienta con el resultado final que opta el estudiante para la práctica.

Es OBLIGATORIO incluir en el guion una sección titulada: "Análisis propuesta IA" donde se incluya enlace al chat/prompt con las consulta/as realizada/as, resultado que proporciona la IA y un párrafo con un análisis crítico y detallado del resultado proporcionado.

## Comentarios en las configuraciones de archivos

Para cada configuración realizada, es OBLIGATORIO añadir comentario describiendo en detalle la configuración.

## Evaluación

La práctica se realizará de manera individual. Tiene un peso del **15%** del total de prácticas.

Para superar la práctica se deben realizar las siguientes **tareas básicas**:

### B1. Configuración del Entorno:

- Instalar Docker y Docker Compose si aún no están instalados.
- Crear el directorio **web\_usuarioUGR** y el archivo **index.php**.

### B2. Creación del Dockerfile:

- Escribir un Dockerfile **DockerfileApache\_usuarioUGR** que use una imagen base de Linux, e instale Apache, PHP y herramientas de red.

### B3. Uso de Docker Compose:

- Escribir un archivo **docker-compose.yml** que defina la construcción de la imagen **usuarioUGR-apache-image:p1** y la creación de los 8 contenedores **webX**.

### B4. Despliegue y verificación de Contenedores:

- Ejecutar **docker-compose up** para iniciar los contenedores.
- Usar **docker ps** para asegurarse de que todos los contenedores están en ejecución.
- Verificar que cada contenedor tiene una dirección IP asignada en las redes **red\_web** y **red\_servicios**.

### B5. Pruebas Básicas:

- Acceder a la página web de cada contenedor usando su dirección IP y verificar que muestra la información correcta.

Se proponen, opcionalmente, las siguientes **tareas avanzadas**:

### A1. Personalización del Dockerfile:

- Modificar el Dockerfile para incluir configuraciones personalizadas de Apache o PHP.

### A2. Creación de contenedores con otros servidores web

- Crear contenedores con otros servidores web (nginx, lighttpd, etc.)

### A3. Gestión Avanzada de Redes:

- Configurar reglas específicas de enrutamiento o restricciones de acceso entre las dos redes **red\_web** y **red\_servicios**.

### A4. Automatización con Scripts:

- Crear scripts para tareas de mantenimiento automatizado, como limpieza de logs, monitoreo de la salud del contenedor, o actualizaciones automáticas de paquetes.
- Escribir scripts para automatizar la creación de contenedores o la configuración de la red.

### A5. Monitoreo y Logging:

- Configurar herramientas de monitoreo y logging para rastrear el rendimiento y los eventos de los contenedores.
- Utilizar herramientas como **htop**, **netstat**, o **apache2ctl** dentro de los contenedores para monitorear y diagnosticar el estado del servidor.

## Normas de entrega

Se desarrollará un documento siguiendo el guion de la práctica y **detallando** e indicando, en su caso, los **aspectos básicos y avanzados realizados**, comandos de terminal ejecutados, resultados de ejecución, etc.

- Por ejemplo, si se ha realizado la tarea básica de configuración del entorno, el documento .pdf con la memoria de prácticas debe aparecer una sección titulada: *Tareas Básicas - B1. Configuración del Entorno* donde aparezcan detalladas las configuraciones y explicaciones sobre ellas. De igual forma, si por ejemplo, se han realizado tareas avanzadas sobre automatizaciones con Scripts, debe aparecer *Tareas Avanzadas - A4. Automatización con Scripts*, detalles de las configuraciones, explicaciones sobre ellas.

Se recomienda utilizar herramientas de control de Tiempo (por ejemplo, clockify) para contabilizar el tiempo de dedicado a la realización de la práctica.

Se deja a **libre elección** la **estructura y formato** del documento el cual reflejará el correcto desarrollo de la práctica a modo de diario/tutorial siguiendo los puntos descritos anteriormente. Asimismo, se recomienda incluir capturas de pantalla que reflejen el correcto desarrollo de los distintos apartados de la práctica. La **primera página** del documento debe incluir, al menos, **nombre, apellidos y tiempo dedicado a la práctica** medido con herramientas de control de tiempo.

Para la entrega se habilitará una tarea en PRADO cuya entrega debe seguir **OBLIGATORIAMENTE** el formato especificado.

1. Un archivo .pdf con el documento desarrollado siguiendo el formato **ApellidosNombreP1.pdf**
2. Un archivo .zip con los distintos archivos de configuraciones, carpetas, etc. necesarios para la ejecución de la práctica siguiendo el formato **ApellidosNombreP1.zip**

## Uso de Inteligencia Artificial Generativa.

Para cada práctica es **OBLIGATORIO** usar herramientas de IA generativa (ChatGPT, Copilot u otras) e incluir enlace al chat/prompt utilizado. También se debe analizar y justificar el resultado que proporciona la herramienta con el resultado final que opta el estudiante para la práctica.

Es **OBLIGATORIO** incluir en el guion una sección titulada: **"Análisis propuesta IA"** donde se incluya enlace al chat/prompt con las consulta/as realizada/as, resultado que proporciona la IA y un párrafo con un análisis crítico y detallado del resultado proporcionado.

## Evaluación

La práctica se evaluará mediante el uso de rúbrica específica (accesible por el estudiante en la tarea de entrega) y una defensa final de prácticas.

Cuestiones sobre la calificación obtenida en cada práctica se realizarán **UNICAMENTE** en la sesión dedicada a recuperación/defensa al final de curso.

La detección de prácticas copiadas implicará el suspenso inmediato de todos los implicados en la copia (tanto del autor del original como de quien las copió). **OBLIGATORIO ACEPTAR LICENCIA EULA DE TURNITIN** en la entrega. Si la memoria supera un 40% de copia Turnitin implicará el suspenso automáticamente.

## Rúbrica

Criterios de Evaluación	Excelente (Puntuación máxima)	Bueno (75% de la puntuación máxima)	Adecuado (50% de la puntuación máxima)	Deficiente (25% de la puntuación máxima)	No Realizado (0 puntos)
<b>Tareas básicas</b>					
B1. Configuración del Entorno (5 puntos)	Todas las tareas completadas con documentación detallada y sin errores. (5 puntos)	Tareas completadas con pequeños errores o falta de detalle. (3.75 puntos)	Tareas parcialmente completadas o con errores significativos. (2.5 puntos)	Intento de configuración pero con errores graves o falta de componentes clave. (1.25 puntos)	No realizado. (0 puntos)
B2. Creación y Configuración de Dockerfile (15 puntos)	Dockerfile perfectamente configurado según especificaciones. (15 puntos)	Dockerfile bien configurado con menores inexactitudes. (11.25 puntos)	Dockerfile parcialmente configurado o con errores importantes. (7.5 puntos)	Intento de creación de Dockerfile pero con errores graves. (3.75 puntos)	No realizado. (0 puntos)
B3. Uso de Docker Compose (30 puntos)	docker-compose.yml perfectamente configurado y funcionando. (30 puntos)	Configuración adecuada con pequeñas inexactitudes. (22.5 puntos)	Configuración parcial o con errores significativos. (15 puntos)	Intento de configuración pero con errores graves. (7.5 puntos)	No realizado. (0 puntos)
B4. Despliegue y Verificación de Contenedores (5 puntos)	Todos los contenedores desplegados y verificados correctamente. (5 puntos)	Mayoría de contenedores desplegados y verificados con pequeños errores. (3.75 puntos)	Algunos contenedores desplegados y verificados correctamente. (2.5 puntos)	Intento de despliegue pero con graves errores o falta de verificación. (1.25 puntos)	No realizado. (0 puntos)
B5. Pruebas Básicas de la Página Web (5 puntos)	Todas las pruebas realizadas exitosamente con documentación completa. (5 puntos)	Mayoría de las pruebas realizadas correctamente con pequeñas omisiones. (3.75 puntos)	Algunas pruebas realizadas correctamente. (2.5 puntos)	Intento de pruebas pero con resultados inadecuados. (1.25 puntos)	No realizado. (0 puntos)
<b>Tareas avanzadas</b>					
A1. Personalización del Dockerfile (10 puntos)	Dockerfile personalizado con funcionalidades avanzadas y documentación detallada. (10 puntos)	Dockerfile personalizado con algunas funcionalidades avanzadas. (7.5 puntos)	Dockerfile ligeramente personalizado. (5 puntos)	Intento de personalización pero con deficiencias. (2.5 puntos)	No realizado. (0 puntos)
A2. Creación de contenedores con otros servidores web (5 puntos)	Uso adecuado de un servidor web alternativo con configuración avanzada. (5 puntos)	Uso de un servidor web alternativo con configuración básica. (3.75 puntos)	Uso de un servidor web alternativo pero con errores en la configuración. (2.5 puntos)	Intento de uso de otro servidor web pero con graves errores. (1.25 puntos)	No realizado. (0 puntos)

A3. Gestión Avanzada de Redes (5 puntos)	Implementación avanzada de redes con documentación detallada. (5 puntos)	Implementación adecuada de redes con algunas inexactitudes. (3.75 puntos)	Implementación básica de redes. (2.5 puntos)	Intento de implementación de redes pero con errores significativos. (1.25 puntos)	No realizado. (0 puntos)
A4. Automatización con Scripts (10 puntos)	Scripts para automatización bien implementados, ejecutados y documentados. (10 puntos)	Scripts para automatización con pequeños errores o falta de detalle. (7.5 puntos)	Uso básico de scripts para tareas simples. (5 puntos)	Intento de uso de scripts pero con resultados insatisfactorios. (2.5 puntos)	No realizado. (0 puntos)
A5. Monitoreo y Logging (10 puntos)	Implementación avanzada de herramientas de monitoreo y logging con documentación completa. (10 puntos)	Implementación adecuada de herramientas de monitoreo y logging. (7.5 puntos)	Uso básico de herramientas de monitoreo y logging. (5 puntos)	Intento de implementación pero con graves deficiencias. (2.5 puntos)	No realizado. (0 puntos)
<b>Documentación</b>					
Documentación de la Práctica (15 puntos)	Memoria detallada, bien estructurada, sin errores ortográficos. (15 puntos)	Memoria completa con algunos detalles menores faltantes o errores leves. (11.25 puntos)	Memoria completa pero con varias omisiones o errores ortográficos. (7.5 puntos)	Memoria incompleta, desorganizada o con numerosos errores. (3.75 puntos)	No realizada. (0 puntos)
Análisis y Justificación de Resultados de IA Generativa (25 puntos)	Análisis profundo y justificación detallada de las configuraciones y resultados de IA. (25 puntos)	Análisis adecuado con justificaciones superficiales o incompletas. (18.75 puntos)	Análisis básico con pocas justificaciones o reflexiones críticas. (12.5 puntos)	Intento de análisis pero con justificaciones inadecuadas. (6.25 puntos)	No realizado. (0 puntos)
Comentarios en Configuraciones (10 puntos)	Comentarios detallados y claros en todas las configuraciones. (10 puntos)	Comentarios adecuados en la mayoría de las configuraciones. (7.5 puntos)	Algunos comentarios útiles, pero falta de detalle o claridad. (5 puntos)	Comentarios escasos o poco claros. (2.5 puntos)	Sin comentarios en las configuraciones. (0 puntos)
<b>Penalizaciones</b>					
Requisitos de entrega	Cumple: 0 puntos		No cumple: -15 puntos		
Entrega en plazo fijado	En plazo: 0 puntos		Un poco tarde (horas): -10 puntos	Algo tarde (1 día): -15 puntos	Muy tarde (varios días): -20 puntos
Porcentaje de copia en Turnitin	1-10%: 0 puntos	11-20%: -15 puntos	21-30%: -20 puntos	31-40%: -25 puntos	Más del 40%: Suspenso automático