

¡Manos a la obra!

### 1- Desafío

#### Consigna del desafío:

Elabora un documento o presentación breve (máximo 1 página o 2-3 diapositivas) en el que expliques de forma sencilla el proceso de contenerización de una aplicación monolítica utilizando Docker, basándote en los contenidos del manual "IMPLEMENTACIÓN MEDIANTE CONTENEDORES DE APLICACIÓN". Tu respuesta debe incluir:

- **Conceptos Básicos de Docker:**
  - Define los términos: Contenedor, Imagen, Dockerfile, Docker Engine, Docker Hub, Volúmenes, Red de Docker, Docker Compose y Capas de Imagen.
- **Implementación Práctica:**
  - Describe el proceso de instalación y configuración de Docker (utilizando Docker Desktop, por ejemplo).
  - Explica cómo crear una imagen Docker mediante un Dockerfile, resaltando buenas prácticas en su construcción.
  - Detalla cómo ejecutar un contenedor Docker, incluyendo aspectos básicos como el monitoreo de logs y el manejo del networking entre el host y el contenedor.
- **Despliegue en la Nube (Opcional):**
  - Menciona brevemente cómo se podría desplegar un contenedor en la nube utilizando AWS ECS (Servicio Elástico de Contenedores), si lo deseas.

### 2- ¿Dónde se lleva a cabo?

- **Entorno Local:** Docker Desktop para la práctica de contenerización.
- **En la Nube (Opcional):** AWS Academy para simular el despliegue en AWS ECS.

### 3- Tiempo de dedicación

- **Tiempo estimado:** 1 hora

### 4- Recursos

- Contenidos del manual "IMPLEMENTACIÓN MEDIANTE CONTENEDORES DE APLICACIÓN".
- Documentación oficial de Docker ([docs.docker.com](https://docs.docker.com)) y de AWS ECS para despliegues en la nube (opcional).

### 5- Plus

- **Opcional:** Agrega un diagrama sencillo (usando herramientas gratuitas como Lucidchart o Draw.io) que ilustre el flujo de creación, ejecución y despliegue de contenedores.

### 6- Condición

Esta práctica es para autoevaluación y no requiere entrega formal. Se recomienda compartir tus resultados con compañeros para fomentar el aprendizaje colaborativo.

### Resolución del Desafío: Implementación Mediante Contenedores de Aplicación

Esta resolución está diseñada para completarse en 1 hora. A continuación, se muestra un ejemplo de cómo estructurar tu respuesta:

#### 1. Introducción

##### Objetivo:

Explicar de forma sencilla el proceso de contenerización de una aplicación monolítica utilizando Docker, resaltando conceptos básicos y pasos prácticos para crear y ejecutar contenedores.

##### Contexto:

La contenerización permite empaquetar una aplicación junto con todas sus dependencias, garantizando portabilidad y consistencia entre entornos de desarrollo, pruebas y producción.

#### 2. Conceptos Básicos de Docker

- **Contenedor:**  
Una instancia aislada que ejecuta una aplicación y sus dependencias de forma consistente en cualquier entorno.
- **Imagen:**  
Una plantilla inmutable que define el entorno y la aplicación, a partir de la cual se crean contenedores.
- **Dockerfile:**  
Un archivo de texto con instrucciones para construir una imagen Docker de forma automatizada.
- **Docker Engine:**  
El motor que permite construir, ejecutar y gestionar contenedores.
- **Docker Hub:**  
Un repositorio central para almacenar y compartir imágenes Docker.

- **Volúmenes:**  
Permiten la persistencia de datos fuera del ciclo de vida de un contenedor.
- **Red de Docker:**  
Configuración que define cómo se comunican los contenedores entre sí y con el host.
- **Docker Compose:**  
Herramienta para definir y ejecutar aplicaciones multi-contenedor a través de un archivo YAML.
- **Capas de Imagen:**  
Cada instrucción en un Dockerfile crea una capa, permitiendo la reutilización y optimización de imágenes.

### 3. Implementación Práctica con Docker

#### Instalación y Configuración

- **Instalación:**  
Utiliza Docker Desktop (gratuito) para instalar Docker en tu equipo.
- **Verificación:**  
Ejecuta `docker --version` en la terminal para confirmar la instalación.

#### Creación de una Imagen Docker

- **Escribir un Dockerfile:**  
Crea un archivo `Dockerfile` que defina:
  - Una imagen base (por ejemplo, `FROM ubuntu:20.04`).
  - Instalación de dependencias necesarias.
  - Copia de la aplicación.

- Comando para iniciar la aplicación.
- **Construcción de la Imagen:**  
Ejecuta el comando `docker build -t nombre-imagen .` para crear la imagen.
- **Buenas Prácticas:**
  - Minimiza el número de capas combinando comandos.
  - Limpia archivos temporales para reducir el tamaño de la imagen.

### Ejecución de un Contenedor Docker

- **Iniciar un Contenedor:**  
Usa el comando `docker run -d -p 80:80 nombre-imagen` para ejecutar el contenedor y mapear puertos.
- **Monitoreo:**  
Emplea `docker logs <container-id>` para ver los logs y `docker ps` para listar contenedores en ejecución.
- **Networking:**  
Explica brevemente cómo se establece la comunicación entre el contenedor y el host.

### 4. Despliegue en la Nube (Opcional)

- **AWS ECS:**  
Menciona que, opcionalmente, se puede utilizar AWS ECS para desplegar contenedores en la nube sin gestionar la infraestructura subyacente.
- **Beneficios:**
  - Simplifica la escalabilidad y el manejo de contenedores en producción.

- Integración con otros servicios de AWS.

### 5. Conclusión

- **Resumen:**

La contenerización con Docker permite crear entornos consistentes y portables.

- **Importancia:**

Facilita el desarrollo, pruebas y despliegue, garantizando que la aplicación funcione de forma idéntica en todos los entornos.

- **Resultado Esperado:**

Un documento breve que explique claramente los conceptos, pasos prácticos para crear y ejecutar contenedores, y una mención (opcional) del despliegue en AWS ECS.

### **Recomendación:**

Complementa tu respuesta con un diagrama sencillo que ilustre el proceso de creación de una imagen, ejecución de un contenedor y, opcionalmente, el flujo hacia AWS ECS.