

# Introducción a Docker: Conceptos fundamentales y requisitos para su funcionamiento

# Introducción a Docker

En esta presentación exploraremos los **conceptos fundamentales** y los requisitos para el funcionamiento de Docker. ¡Prepárate para sumergirte en el mundo del contenedorización y la virtualización!

# ¿Qué es Docker?

Docker es una plataforma de **virtualización** que permite a los desarrolladores empaquetar, enviar y ejecutar aplicaciones en entornos aislados llamados *contenedores*. Estos contenedores son portátiles y consistentes en cualquier entorno.

Un contenedor es una unidad de software estandarizada que empaqueta código, bibliotecas, herramientas y configuraciones necesarias para ejecutar una aplicación. Los contenedores son aislados entre sí, lo que permite que se ejecuten de forma segura y eficiente en cualquier entorno.



# Beneficios de Docker

Los beneficios clave de Docker incluyen la **portabilidad**, la **eficiencia** en el uso de recursos y la **escalabilidad**. Con Docker, las aplicaciones pueden ejecutarse de manera predecible en cualquier entorno.



## Arquitectura de Docker

La arquitectura de Docker se compone de **clientes**, **servidores** y un **registro**. Los contenedores se ejecutan en un **motor de contenedores** que gestiona su creación y ejecución.



## Requisitos de Docker

Para utilizar Docker, es necesario disponer de un sistema operativo compatible, como **Linux** o **Windows**, y tener instalado el **motor de Docker**. Además, se requiere un **registro de Docker** para almacenar imágenes.



## Instalación de Docker

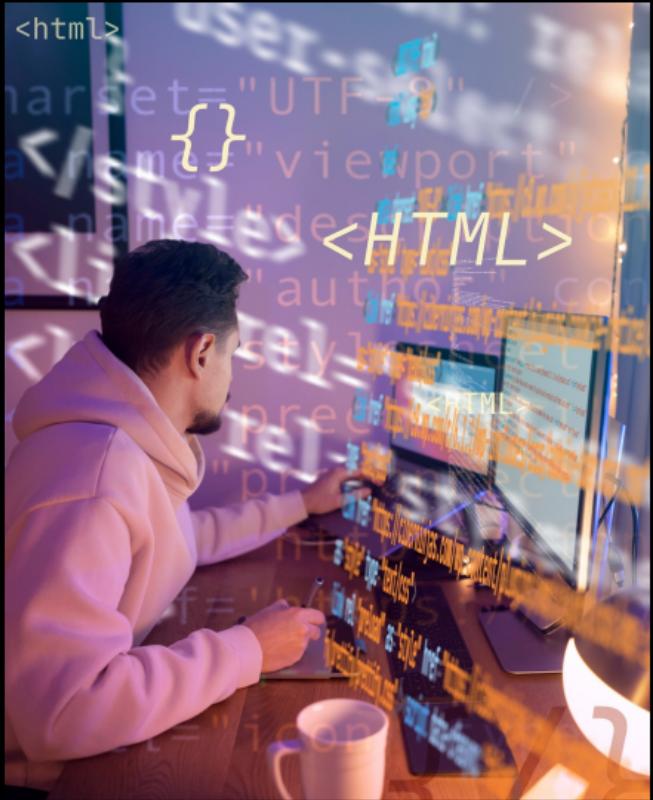
La instalación de Docker varía según el sistema operativo. En Linux, se puede instalar a través de un script. En Windows, se puede utilizar **Docker Desktop** para una instalación sencilla.

# Dockerfile

El **Dockerfile** es un archivo de texto que contiene los comandos necesarios para construir una imagen de contenedor.

Este archivo define el **entorno de ejecución** y las **dependencias** de la aplicación.





# Docker Compose

Docker Compose es una herramienta que permite definir y ejecutar aplicaciones **multi-contenedor**. Con un archivo **YAML**, se pueden configurar los servicios de la aplicación y su interacción.



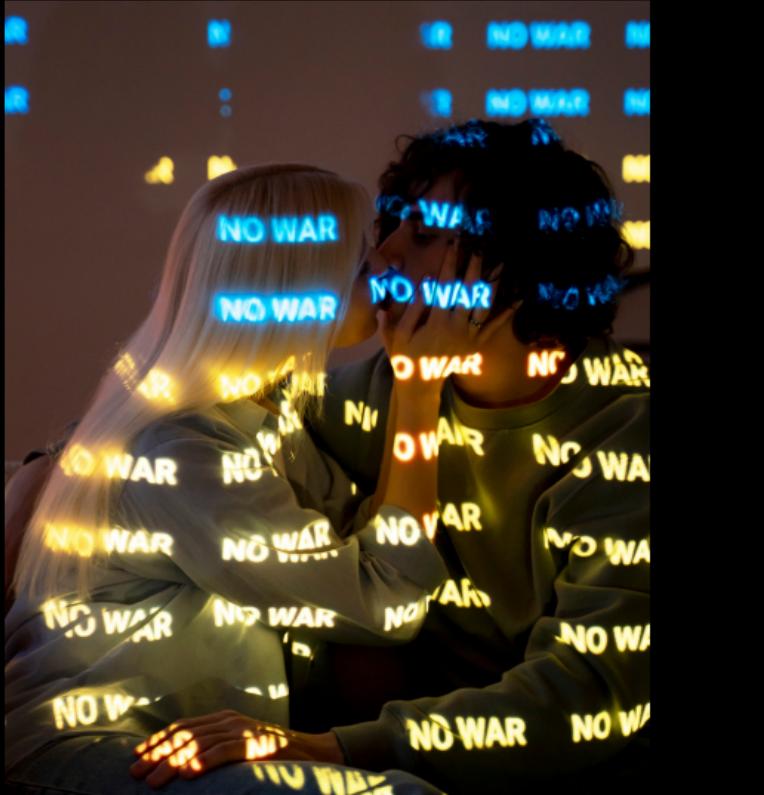
# Redes en Docker

Docker proporciona **diversos modos de red** para conectar contenedores entre sí y con el mundo exterior. Las redes pueden ser **punteadas, superpuestas o personalizadas** según las necesidades.

# Almacenamiento en Docker

El almacenamiento en Docker se gestiona a través de **volúmenes** y **puntos de montaje**. Los volúmenes permiten persistir datos más allá del ciclo de vida de un contenedor.





# Seguridad en Docker

La seguridad en Docker es fundamental. Se pueden aplicar **políticas de acceso y control de recursos** para proteger los contenedores. Además, es crucial mantener las **imágenes actualizadas** para evitar vulnerabilidades.

## Como funciona

- Un desarrollador crea una imagen Docker que contiene el código, las bibliotecas y las dependencias necesarias para ejecutar una aplicación.
- La imagen Docker se carga en un repositorio de imágenes, como Docker Hub.
- Un administrador de sistemas usa Docker para crear un contenedor a partir de una imagen Docker.
- El contenedor se ejecuta en un servidor.



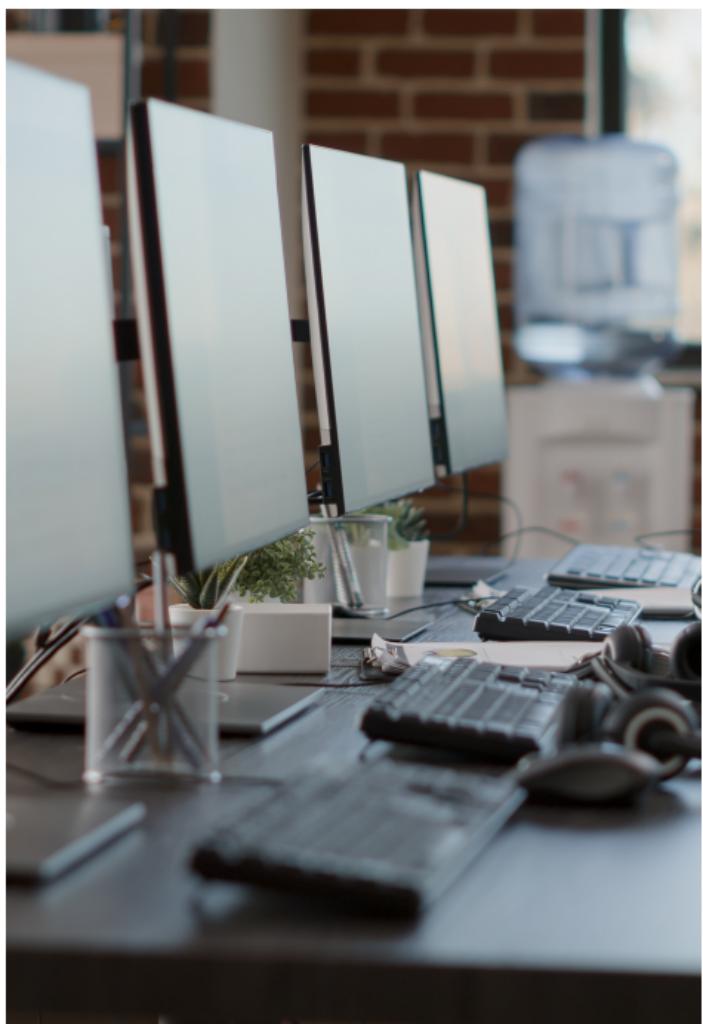
## Ventajas de usar docker

- Eficiencia: Los contenedores son más eficientes que las máquinas virtuales, ya que comparten el kernel del sistema operativo.
- Portabilidad: Los contenedores se pueden ejecutar en cualquier entorno que admita Docker.
- Seguridad: Los contenedores están aislados entre sí, lo que reduce el riesgo de que una aplicación infectada afecte a otras aplicaciones.
- Docker es una herramienta poderosa que puede ayudar a los desarrolladores y administradores de sistemas a crear, implementar y administrar aplicaciones de forma más eficiente y segura



## Como utilizarlo

- Despliegue de aplicaciones web: Docker se puede utilizar para desplegar aplicaciones web en un entorno de producción.
- Implementación de aplicaciones móviles: Docker se puede utilizar para implementar aplicaciones móviles en un entorno de prueba o producción.
- Despliegue de aplicaciones de datos: Docker se puede utilizar para desplegar aplicaciones de datos en un entorno de producción.
- Eficiencia: Los contenedores son más eficientes que las máquinas virtuales, ya que comparten el kernel del sistema operativo.



# Conclusiones

En esta presentación hemos explorado los conceptos fundamentales de Docker y los requisitos para su funcionamiento. Docker ofrece beneficios significativos en términos de **portabilidad, eficiencia y escalabilidad**.

# Thanks!

Do you have any questions?

joseesparza@uadec.edu.mx

4B ISC MATUTINO

Creado por Jose Luis Leija Esparza

