

# Explorando la Contenedorización: Proyecto Práctico con Docker

Ingeniería | Ingeniería de sistemas | Aprendizaje Basado en Proyectos

## Descripción

Este plan de clase está diseñado para que los estudiantes universitarios de Ingeniería de Sistemas comprendan y apliquen los conceptos fundamentales de la contenedorización utilizando Docker. A través de un enfoque de Aprendizaje Basado en Proyectos, los estudiantes desarrollarán un proyecto práctico que les permitirá experimentar cómo empaquetar, distribuir y ejecutar aplicaciones en contenedores, facilitando la portabilidad y escalabilidad de software. La relevancia de Docker en la industria tecnológica actual es indiscutible, ya que permite optimizar procesos de desarrollo, pruebas y despliegue, habilidades que aumentan la empleabilidad y preparación profesional de los estudiantes. Además, este conocimiento conecta con su vida real al mostrar cómo se gestionan aplicaciones modernas en la nube y entornos empresariales, promoviendo el trabajo colaborativo y la autonomía en la resolución de problemas tecnológicos complejos.

## Objetivos de Aprendizaje

- Explicar los conceptos básicos y ventajas de la contenedorización con Docker.
- Configurar y usar Docker para crear contenedores que ejecuten aplicaciones simples.
- Diseñar y construir un proyecto colaborativo que integre contenedores Docker para resolver un problema específico.
- Analizar y evaluar el impacto de la contenedorización en los procesos de desarrollo y despliegue de software.

## Recursos Necesarios

- Computadoras con sistema operativo compatible y acceso a internet (1 por estudiante o pareja).
- Instalación previa de Docker Desktop o Docker Engine configurado.
- Editor de código (Visual Studio Code, Sublime Text o similar).
- Material impreso con guía rápida de comandos básicos de Docker (1 por estudiante).
- Proyector multimedia para presentación inicial y demostraciones.
- Repositorio digital compartido (GitHub Classroom o similar) para entrega y seguimiento del proyecto.
- Documentación oficial de Docker accesible en línea.

## Requisitos Previos

- Conocimientos básicos de sistemas operativos y línea de comandos.
- Familiaridad con conceptos de virtualización y administración de entornos de desarrollo.

- Experiencia previa en programación básica en algún lenguaje (ej. Python, JavaScript o Java).
- Habilidades para trabajar colaborativamente en equipos pequeños.

## Actividades

### Fase de Inicio

#### Tiempo estimado:

20 minutos

#### Propósito de la sesión:

**Docente:** Explica que la sesión busca introducir a los estudiantes en la contenedorización con Docker, una tecnología clave para el desarrollo y despliegue de aplicaciones modernas, y enfatiza la importancia de esta habilidad en la industria actual.

#### Activación de conocimientos previos:

**Docente:** Inicia con la pregunta detonadora: "*¿Cómo creen que se podría empaquetar una aplicación para que funcione igual en cualquier computadora sin importar el sistema operativo?*" Solicita a los estudiantes que reflexionen y compartan ideas brevemente en plenaria.

**Estudiantes:** Participan comentando experiencias previas con instalaciones de software o problemas de compatibilidad.

#### Motivación y enganche:

**Docente:** Presenta un dato curioso: "*Empresas como Spotify, Netflix y PayPal utilizan Docker para acelerar sus procesos de desarrollo y garantizar que sus aplicaciones funcionen de forma confiable en cualquier entorno.*" Muestra un breve video (3 minutos) que ilustra casos reales de uso de Docker en la industria.

**Estudiantes:** Observan el video y plantean preguntas iniciales.

#### Contextualización:

**Docente:** Relaciona cómo la contenedorización puede facilitar la vida profesional de los estudiantes al permitirles entregar proyectos que funcionen sin errores en ambientes diferentes y mejorar la colaboración en equipo.

**Estudiantes:** Comprenden la conexión entre la teoría y su futura práctica profesional.

---

### Fase de Desarrollo

#### Tiempo estimado:

75 minutos

#### Presentación del contenido:

**Docente:** Introduce brevemente los conceptos clave de Docker: imagen, contenedor, Dockerfile y comandos básicos, apoyándose en un esquema visual proyectado. En lugar de una clase magistral extensa, plantea un proyecto real: desarrollar una aplicación web simple y contenerizarla.

### Actividad 1: Configuración y exploración básica de Docker

- **Objetivo:** Explicar los conceptos básicos y ventajas de la contenedorización con Docker.
- **Instrucciones:**
  - **Docente:** Indica a los estudiantes que en parejas abrirán la terminal y ejecutarán comandos para listar contenedores, imágenes y crear un contenedor básico usando una imagen oficial (ej. hello-world).
  - Proporciona la guía rápida impresa con comandos.
- **Organización:** Parejas
- **Producto:** Captura de pantalla o breve informe de los comandos ejecutados y resultados obtenidos.
- **Tiempo:** 25 minutos
- **Rol docente:** Circula entre parejas, responde dudas, formula preguntas como: "¿Qué observan cuando ejecutan el contenedor hello-world? ¿Cómo creen que funciona la imagen detrás de esto?"

### Actividad 2: Construcción y despliegue de una aplicación contenida

- **Objetivo:** Configurar y usar Docker para crear contenedores que ejecuten aplicaciones simples.
- **Instrucciones:**
  - **Docente:** Presenta un pequeño proyecto: una aplicación web estática con HTML y servidor simple (ejemplo con Python Flask o Node.js). Explica cómo crear un Dockerfile para esta aplicación.
  - Los estudiantes, en grupos de 3-4, desarrollan o adaptan el código y crean el Dockerfile siguiendo pasos guiados en el repositorio compartido.
- **Organización:** Grupos de 3-4 estudiantes
- **Producto:** Dockerfile funcional y contenedor que ejecuta la aplicación localmente.
- **Tiempo:** 35 minutos
- **Rol docente:** Supervisar, orientar la escritura del Dockerfile, preguntar: "¿Qué instrucciones usan y por qué? ¿Cómo aseguran que la aplicación se ejecute correctamente dentro del contenedor?"

### Actividad 3: Presentación y discusión del proyecto

- **Objetivo:** Analizar y evaluar el impacto de la contenedorización en los procesos de desarrollo y despliegue de software.
- **Instrucciones:**
  - **Docente:** Cada grupo presenta brevemente su contenedor y explica el proceso, retos y beneficios observados.
  - Se promueve una discusión grupal con preguntas como: "¿Cómo cambiaría este proceso sin Docker? ¿Qué ventajas encontraron?"

- **Organización:** Plenaria
- **Producto:** Presentación oral y discusión grupal.
- **Tiempo:** 15 minutos
- **Rol docente:** Facilitar la discusión, sintetizar ideas clave y conectar con objetivos.

### Diferenciación:

- **Estudiantes que terminan antes:** Se les invita a explorar comandos avanzados de Docker (ej. docker-compose) o a investigar casos de uso reales para compartir brevemente.
- **Estudiantes con más dificultades:** Se ofrece apoyo individualizado o en mini-grupos, con ejemplos más simples y pasos guiados adicionales.

### Transiciones:

El docente conecta la exploración inicial con la construcción del proyecto práctico, y luego integra ambas experiencias en la presentación y discusión, facilitando una comprensión integral.

---

## Fase de Cierre

### Tiempo estimado:

25 minutos

### Síntesis:

**Docente:** Propone una actividad de mapa mental colectivo en la pizarra o digital, donde los estudiantes aportan conceptos, ventajas y retos de Docker y la contenedorización.

**Estudiantes:** Participan en la construcción del mapa, recapitulando lo aprendido.

### Reflexión metacognitiva:

- ¿Qué habilidades nuevas desarrollaron hoy relacionadas con Docker y la contenedorización?
- ¿Cómo creen que podrían aplicar esta tecnología en futuros proyectos o en su vida profesional?
- ¿Qué dificultades encontraron y cómo las superaron durante el desarrollo del proyecto?

**Docente:** Solicita que los estudiantes respondan por escrito o en voz alta, promoviendo la autoevaluación.

### Retroalimentación:

**Docente:** Proporciona retroalimentación inmediata destacando logros y áreas de mejora, comentando aspectos técnicos y de colaboración observados durante las actividades.

### Transferencia:

**Docente:** Explica que este aprendizaje es base para trabajar con tecnologías de orquestación de contenedores como Kubernetes y para proyectos profesionales en ambientes cloud y DevOps.

## Tarea o reto:

**Docente:** Propone como tarea opcional que los estudiantes creen un contenedor para una aplicación propia o analicen un caso real de uso de Docker en una empresa y preparen un breve informe para la próxima clase.

## Evaluación

### Tipo de evaluación:

- Diagnóstica: En la fase de inicio, a través de la pregunta detonadora para identificar conocimientos previos.
- Formativa: Durante la fase de desarrollo, mediante observación directa, interacción con estudiantes y revisión de productos parciales (comandos ejecutados, Dockerfile creado, presentación grupal).
- Sumativa: Al cierre de la sesión, evaluando el mapa mental colectivo, respuestas de reflexión y entregables del proyecto.

### Criterios de evaluación:

- Capacidad para explicar y aplicar los conceptos básicos de Docker (Objetivo 1).
- Correcta configuración y despliegue de contenedores funcionales (Objetivo 2).
- Participación activa y colaboración en el proyecto grupal (Objetivo 3).
- Análisis crítico sobre el impacto de la contenedorización en desarrollo de software (Objetivo 4).

### Instrumentos sugeridos:

- Lista de cotejo para seguimiento de actividades prácticas.
- Rúbrica para evaluar el proyecto Dockerfile y presentación.
- Observación directa y registro anecdótico para participación y reflexión.
- Autoevaluación escrita al final de la sesión.

### Evidencias de aprendizaje:

- Capturas de pantalla o informes de comandos Docker ejecutados.
- Dockerfile funcional y contenedor en ejecución.
- Presentación grupal y aportaciones en discusión.
- Mapa mental colectivo y respuestas reflexivas.