

Evaluación Final

Containerización y Orquestación de Contenedores

Objetivo general

Aplicar los conocimientos adquiridos sobre **contenedores, redes, volúmenes, servicios, orquestación y despliegue en clústeres**, mediante la creación, containerización y despliegue de una aplicación práctica de forma local y dentro de un cluser **Docker Swarm** y **Kubernetes**.

Objetivos específicos

1. Diseñar una arquitectura de aplicación modular y escalable.
2. Crear imágenes personalizadas utilizando **Dockerfiles** optimizados.
3. Gestionar redes, volúmenes y variables de entorno en entornos Docker.
4. Implementar un **docker-compose.yml** funcional para desarrollo local.
5. Desplegar la aplicación en un **clúster Docker Swarm**, usando stacks y servicios replicados.
6. Adaptar la misma solución al entorno **Kubernetes**, aplicando manifestos YAML (Replicaset, Deployments, Services, Load Balancer, etc.).
7. Documentar el proceso de implementación y validar el funcionamiento del despliegue.

Requerimientos del proyecto

El estudiante deberá desarrollar o utilizar una **aplicación multi-servicio** (de libre elección, o una de las siguientes sugerencias):

- Agenda de contactos (frontend + API + base de datos)
- Aplicación de tareas pendientes (To-Do List)
- Sistema de comentarios (frontend, API y almacenamiento)
- Servicio de registro de usuarios (autenticación + base de datos)
-

Requerimientos técnicos mínimos

1. Arquitectura

- Mínimo **5 a 7 servicios** diferenciados:
 - Frontend
 - Backend/API
 - Base de datos
- Uso de **variables de entorno** para configuración.
- Comunicación interna mediante **red personalizada** de Docker/Kubernetes.

2. Containerización

- Cada servicio debe contar con su **Dockerfile** propio.
- Las imágenes deben construirse con **etiquetas versionadas (tags)**.
- Usar buenas prácticas:
 - Imágenes base ligeras (Alpine preferentemente)
 - `.dockerignore` configurado
 - No exponer credenciales en los Dockerfiles

3. Despliegue local (docker-compose)

- Crear un archivo `docker-compose.yml` que permita:
 - Levantar toda la solución con `docker-compose`
 - Configurar volúmenes persistentes
 - Configurar redes personalizadas
 - Exponer el frontend al puerto 80 o 8080 del host
 - Incluir variables de entorno y dependencias entre servicios
 - Incluir los entry-points en los servicios que sean necesarios (Frontend, Backend, Base de Datos)

4. Despliegue en Docker Swarm

- Convertir el `docker-compose.yml` en un stack (`stack-deploy.yml`)
- Iniciar un clúster Swarm
- Desplegar con el stack elaborado
- Incluir:
 - Réplicas de servicios (escalabilidad)
 - Versionamiento
 - Configs/Secrets

5. Despliegue en Kubernetes

- Crear los manifiestos YAML necesarios:
 - `replicaset.yml`
 - `deployment.yml`
 - `service.yml`
 - `loadbalancer.yml`
 - `secret.yml` (si aplica)

- Hacer uso de versionamiento (2 versiones como mínimo)
- Desplegar y exponer la solución completa en la herramienta de clusterización asignada.
- Verificar el correcto funcionamiento

Entrega y documentación

El alumno deberá:

1. Subir la imagen del producto software a un repositorio público.
2. **Dockerfiles, Docker-compose y manifiestos YAML** utilizados.
3. **Documento técnico (PDF o README.md)** con:
 - Descripción de la aplicación
 - Arquitectura general (diagrama incluido)
 - Instrucciones paso a paso para desplegar en la solución de forma local como en un entorno cluster (Docker Compose, Swarm y Kubernetes respectivamente)
 - Capturas de pantalla o salidas de verificación que demuestren el desarrollo, implementación de la solución solicitada.

Criterios de evaluación

Criterio	Descripción	Ponderación
Arquitectura y diseño	Modularidad, separación de servicios, uso de buenas prácticas	10%
Containerización	Correcto uso de Dockerfiles, tags, y optimización	25%
Despliegue en Docker Compose/Swarm	Correcta ejecución, escalabilidad y persistencia	20%
Despliegue en Kubernetes	Manifiestos correctos, pods y servicios funcionales	20%
Documentación	Claridad, detalle, diagramas y guía reproducible	25%