

Ejercicio Práctico 2: Implementación de Docker, Testing y CI/CD en el Proyecto Final

Contexto:

En este ejercicio, los estudiantes deberán aplicar los conceptos de DevOps en su proyecto desarrollado a lo largo de la especialización. Deberán integrar Docker para la ejecución del proyecto, implementar pruebas automatizadas en al menos un componente, configurar un pipeline de CI/CD con GitHub Actions y gestionar ramas en la nube para mantener un flujo de desarrollo ordenado.

Duración de la evaluación: 3 horas

Requisitos y Puntos de Evaluación

1. Integración de Docker en el Proyecto (2 puntos)

- Crear un `Dockerfile` funcional para el proyecto.
- Configurar `docker-compose.yml` si el proyecto requiere múltiples servicios.
- Asegurar que el contenedor se ejecute correctamente y facilite la portabilidad del proyecto.

2. Implementación de Pruebas Automatizadas (1.5 puntos)

- Crear pruebas unitarias o de integración para al menos un componente clave del proyecto.
- Utilizar herramientas de testing apropiadas como Jest, Mocha, PyTest o JUnit.
- Asegurar que las pruebas se ejecuten correctamente y generen reportes.

3. Configuración de CI/CD con GitHub Actions (2 puntos)

- Implementar un pipeline en GitHub Actions que incluya:
 - Instalación de dependencias.
 - Ejecución de pruebas automatizadas.
 - Construcción y verificación del contenedor Docker.
- Asegurar que el pipeline se ejecute en cada push o pull request a la rama principal.

4. Gestión de Ramas en la Nube (1.5 puntos)

- Crear y gestionar al menos dos ramas en GitHub o GitLab:
 - Una rama principal (`main` o `master`).
 - Una rama secundaria (`develop` o `feature-branch`) para pruebas y cambios.
- Implementar buenas prácticas en el flujo de trabajo, como pull requests y revisiones de código antes de fusionar cambios.

Entrega

Los estudiantes deberán proporcionar una de las siguientes opciones:

- **Opción 1:** Un enlace al repositorio de GitHub con:
 - `Dockerfile` y `docker-compose.yml`.
 - Código de pruebas y resultados.
 - Configuración del pipeline en GitHub Actions (`.github/workflows/ci.yml`).
 - Historial de ramas y pull requests.
- **Opción 2:** Un archivo ZIP con:
 - El código fuente del proyecto con las pruebas y la configuración de Docker.
 - El archivo de configuración de GitHub Actions.
 - Un documento detallando el flujo de trabajo y la gestión de ramas.