

Actividad 4: Planificación del Pipeline de Integración y Entrega Continua (CI/CD)

1. Herramientas seleccionadas

- Repositorio de código: GitHub
GitHub será la plataforma a la que se subirá todo el código, y el control de versiones se hará con Git.
- Contenedorización: Docker
Se utilizará docker para empaquetar la aplicación en contenedores.
- Gestión de dependencias: Maven/Gradle
Automatiza compilaciones y gestión de librerías.
- Orquestación: Kubernetes
Kubernetes administra los contenedores de producción.
- Pruebas unitarias: JUnit
Automatiza las pruebas de unidad del MVP.
- Análisis de calidad: SonarQube
Detecta errores en el código como bugs, vulnerabilidades y estilo.

2. Etapas del Pipeline CI/CD

- Desarrollo y Control de Versiones:
Acción: Los desarrolladores subirán el código a un repositorio Git con ramas de desarrollo (develop) y producción (main).
- Integración Continua (CI):
 - a. Inicio: Se activa con cada push o Pull Request en GitHub.
 - b. Compilación: Se construye el proyecto con Maven/Gradle.
 - c. Análisis de código: Se revisan bugs, vulnerabilidades y estilo.
 - d. Pruebas:
 - Unitarias (JUnit): Validan las funcionalidades individuales de la MVP.
 - Integración (RestAssured): Verifican APIs yUI
 - e. Generación de artefacto: Se almacena el JAR/WAR en un repositorio seguro.
 - f. Aviso: Si el código falla en alguna parte se alerta a los desarrolladores

3. Construcción de la Aplicación

- Generación de contenedores Docker
Se genera una imagen Docker con la versión del código y se almacena en un Docker Registry (DockerHub)

4. Despliegue en Entorno de Staging

- Automatización con Docker Compose/Kubernetes
 - a) Se despliega la aplicación en un entorno de staging para pruebas de aceptación.
 - b) Pruebas end-to-end (Cypress, Selenium).
 - c) Pruebas de carga y seguridad.

5. Aprobación Manual para Producción

- Si las pruebas en staging son exitosas, un revisor aprueba el despliegue a producción.

6. Despliegue en Producción (CD)

- El objetivo de esta parte es automatizar el despliegue del software en entornos de staging y producción, asegurando la estabilidad y la rapidez
 - a. Inicio: Se activa con una compilación exitosa en CI o con un merge a main.
 - b. Gestión de artefactos: Se descarga el JAR/WAR.
 - c. Construcción de contenedor: Se generará una imagen Docker del servicio actualizado.
 - d. Despliegue:
 - Staging: Se despliega automáticamente para realizar pruebas funcionales.
 - Producción: Se realizará bajo aprobación manual para evitar fallos masivos.
 - e. Pruebas post-despliegue: Se ejecutan pruebas de sanity check y monitoreo.
 - f. Monitoreo y rollback: Si se detectan fallos, se activa una reversión automática a la versión estable anterior.