

# Planificación del Pipeline de Integración y Entrega Continua (CI/CD)

## 1. Introducción

Para mejorar el desarrollo del MVP de **EduTech IA**, es importante automatizar el proceso de integración y entrega del código. Esto se logra con un pipeline de **Integración y Entrega Continua (CI/CD)**, que permite detectar errores rápidamente y hacer despliegues eficientes.

## 2. Flujo de Trabajo del Pipeline

El pipeline de CI/CD seguirá estos pasos:

### 2.1. Etapas del Pipeline

#### 1. Commit y Control de Versiones

- Los desarrolladores suben cambios al repositorio de **GitHub**.

#### 2. Pruebas Automáticas

- Se ejecutan pruebas para asegurar que el código funciona correctamente.

#### 3. Integración Continua

- Se verifica que los cambios no generen errores en el sistema.

#### 4. Construcción del Proyecto

- Se compila el código y se generan los archivos necesarios para su ejecución.

#### 5. Despliegue Automático

- Si todo está correcto, el código se envía al servidor para que esté disponible para los usuarios.

### 2.2. Herramientas Utilizadas

- **GitHub Actions:** Automatiza las pruebas y la integración del código.
- **Docker:** Permite empaquetar la aplicación y desplegarla fácilmente.
- **Heroku / AWS:** Servicios en la nube para alojar la aplicación.

## 3. Beneficios del Pipeline CI/CD

- **Automatización:** Reduce el trabajo manual y acelera el desarrollo.
- **Menos errores:** Las pruebas automáticas evitan fallos en producción.
- **Entrega rápida:** Permite lanzar nuevas versiones de forma continua y segura.

#### 4. Conclusión

Implementar un pipeline de CI/CD mejora la calidad del desarrollo, haciendo que los cambios sean seguros y rápidos. Con herramientas como **GitHub Actions** y **Docker**, garantizamos un proceso eficiente y estable para el proyecto **EduTech IA**.