
Introducción a CI/CD

Despliegue de Aplicaciones Web

Curso 2025/2026

Índice general

- 1 ¿Qué es CI/CD? 1**

- 2 Conceptos básicos de CI/CD 3**
 - 2.1 Pipeline 3
 - 2.2 Integración Continua (CI) 3
 - 2.3 Entrega Continua (CD: *Continuous Delivery*) 3
 - 2.4 Despliegue Continuo (CD: *Continuous Deployment*) 4
 - 2.5 Artefactos 4
 - 2.6 Entornos 4

- 3 Referencias 5**

- 4 Licencia 6**

1 ¿Qué es CI/CD?

CI/CD son las siglas de **Integración Continua** (*Continuous Integration*) y **Entrega/Despliegue Continuo** (*Continuous Delivery/Deployment*).

Podemos decir que CI/CD es un conjunto de prácticas que permiten a los equipos de desarrollo de software **automatizar tareas** como la integración del código, realización de tests y el despliegue de las aplicaciones de un manera rápida y segura.

- **CI (Integración Continua):** Consiste en integrar en un repositorio los nuevos cambios que se realizan sobre el código fuente de una aplicación. Para que el nuevo código se pueda integrar con el código existente, tiene que pasar una serie de **test** (unitarios, de integración, etc.) para verificar que el código no tiene errores.
- **CD (Entrega/Despliegue Continuo):** Una vez que se ha verificado que el nuevo código no tiene errores, el proceso de CD se encarga de **automatizar el proceso de entrega y despliegue de la aplicación** en los entornos que se definan, que pueden ser de prueba, preproducción o producción

En el proceso de entrega se realiza la creación de artefactos (imágenes de contenedores, .zip, .jar, .war, etc) y en el proceso de despliegue se instala la aplicación en los servidores.

CI/CD Pipeline Explained

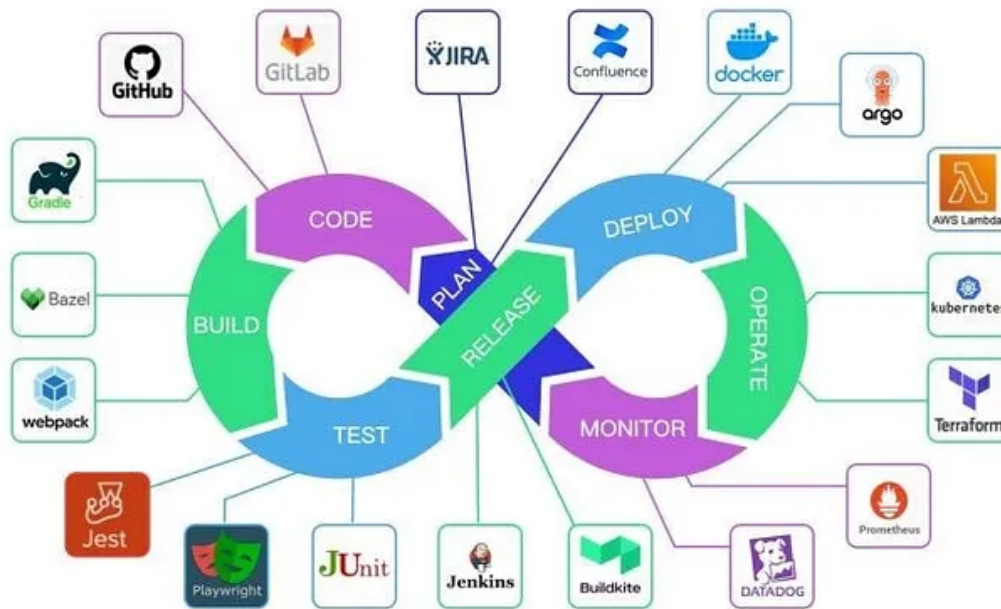


Imagen: [ByteByteGo](#).

2 Conceptos básicos de CI/CD

2.1. Pipeline

Un *pipeline* es una secuencia automatizada de pasos que se ejecutan para construir, probar y desplegar software. Cada paso en el *pipeline* se llama etapa (*stage*), y puede incluir tareas como:

- Compilación del código.
- Ejecución de pruebas (unitarias, de integración, etc).
- Construcción de artefactos (Ejemplo: una imagen de Docker).
- Despliegue en diferentes entornos (pruebas, preproducción, producción).

Los pipelines se suelen definir en archivos [YAML](#).

En [GitHub Actions](#), a los *pipelines* se les llama *workflows*.

2.2. Integración Continua (CI)

Consiste en integrar los nuevos cambios que se realizan de manera frecuente sobre el código fuente de una aplicación en un repositorio compartido.

Un proceso típico de CI puede ser:

- Un desarrollador hace un push de su código al repositorio.
- Se ejecutan pruebas automáticas (unitarias, de integración, etc.).
- Si las pruebas pasan, el código se integra en la rama principal.

Herramientas de CI:

- [Jenkins](#)
- [GitHub Actions](#)
- [GitLab CI](#)
- [CircleCI](#)
- [Travis CI](#)

2.3. Entrega Continua (CD: *Continuous Delivery*)

Consiste en automatizar la entrega del código a entornos de prueba o preproducción.

Un proceso típico de Entrega Continua puede ser:

- Una vez que el código pasa las pruebas de CI, se crea un artefacto que puede ser una imagen Docker que contiene el código de la aplicación.
- El artefacto se despliega en un entorno de pruebas o preproducción.
- El equipo puede decidir cuándo desplegar a producción de forma manual.

2.4. Despliegue Continuo (CD: *Continuous Deployment*)

Consiste en automatizar el despliegue a producción después de que el código pasa todos los tests de CI.

A diferencia con la Entrega Continua, en este caso, el código se despliega automáticamente en producción sin que el equipo tenga que hacer nada.

2.5. Artefactos

Es lo que se crea a partir del código fuente, pueden ser archivos binarios, paquetes o imágenes Docker. Los artefactos se almacenan en repositorios para su posterior uso en despliegues.

2.6. Entornos

Son los diferentes lugares donde se despliega la aplicación. Los más comunes son:

- **Desarrollo:** Donde los desarrolladores trabajan en nuevas funcionalidades.
- **Pruebas:** Donde se ejecutan pruebas automatizadas y manuales.
- **Preproducción:** Un entorno similar a producción para pruebas finales.
- **Producción:** El entorno donde los usuarios finales interactúan con la aplicación.

3 Referencias

- [CI/CD](#). Wikipedia.
- [¿Qué es la CI/CD?](#). GitLab.
- [CI/CD in 5 minutes](#). ByteByteGo.
- [Jenkins](#)
- [GitHub Actions](#)
- [GitLab CI](#)
- [CircleCI](#)
- [Travis CI](#)

4 Licencia

Esta página forma parte del curso Despliegue de Aplicaciones Web de José Juan Sánchez Hernández y su contenido se distribuye bajo una licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.