



**UTT**

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

**GOBIERNO DE BAJA CALIFORNIA**

## **TEMA**

Liberación y despliegue continuo de software (CD)

## **PRESENTADO POR**

Perez Bello Vanory Esperanza

## **GRUPO**

10° B

## **MATERIA**

Software Development Process

## **PROFESOR**

Ray Brunett Parra Galaviz

Tijuana, Baja California, 24 Enero del 2025

## **Liberación y despliegue continuo de software (CD)**

El CD es una práctica en el desarrollo de software que automatiza la entrega de nuevas versiones de una aplicación, asegurando que los cambios realizados (como nuevas funcionalidades o correcciones) puedan ser implementados de forma rápida, fiable y segura en los entornos de producción.

*Aspectos clave:*

### **1. Automatización del proceso:**

- El CD utiliza pipelines automatizados para compilar, probar y desplegar código.
- Permite detectar y solucionar errores antes de llegar a producción.

### **2. Liberación continua:**

- Se enfoca en la preparación de versiones listas para su despliegue.
- Garantiza que cada cambio aprobado esté disponible para ser liberado en cualquier momento.

### **3. Despliegue continuo:**

- Extiende la liberación continua al ejecutar automáticamente los cambios aprobados en el entorno de producción.
- Implica monitoreo constante para detectar problemas después del despliegue.

### **4. Beneficios:**

- **Velocidad:** Reduce el tiempo entre el desarrollo y el uso real en producción.
- **Calidad:** Asegura que el software esté probado y funcional antes de llegar a los usuarios.

- **Colaboración:** Mejora la integración entre desarrolladores, testers y operaciones (DevOps).
- **Confianza:** Minimiza riesgos gracias a despliegues pequeños y frecuentes.

#### 5. Herramientas comunes:

- Jenkins, GitLab CI/CD, CircleCI, Travis CI, y AWS CodePipeline.

#### ***Ciclo típico de CD:***

1. **Integración continua (CI):** Comprobación del código y pruebas automáticas.
2. **Pruebas y validación:** Asegurar calidad y compatibilidad.
3. **Liberación:** Código listo para producción.
4. **Despliegue:** Implementación automatizada en servidores o entornos de producción.

#### ***Consideraciones:***

- Requiere un enfoque bien definido en seguridad y monitoreo.
- Las pruebas automatizadas son esenciales para garantizar la estabilidad del sistema.
- La cultura organizacional juega un papel crucial en el éxito del CD, apoyada por prácticas DevOps.

Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley.

Kim, G., Debois, P., Willis, J., & Humble, J. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations*. IT Revolution Press.

Bass, L., Weber, I., & Zhu, L. (2015). *DevOps: A Software Architect's Perspective*. Addison-Wesley.

Fowler, M. (2013). *Continuous Integration and Continuous Delivery*. Recuperado de [<https://martinfowler.com/bliki/ContinuousDelivery.html>]

Microsoft Azure. (2023). *What is Continuous Delivery?*. Recuperado de [<https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-continuous-delivery/>]