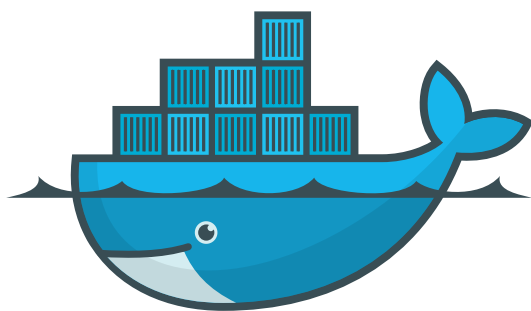


# INTEGRACIÓN CONTINUA

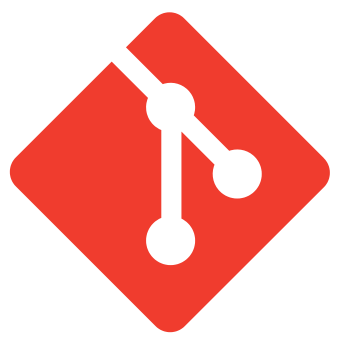
Universidad Tecnológica Nacional – Facultad Regional Córdoba  
Becerra, Cociglio, Córdoba, Esteban, Miranda, Páez, Wedemeyer, Zabala

## INTRODUCCIÓN

La integración continua (CI) automatiza la integración de cambios de código de varios desarrolladores en un proyecto de software, permitiendo detectar errores rápidamente y mantener una base de código estable.



## MATERIALES Y MÉTODOS



Para implementar CI se utilizan herramientas como **Jenkins** para automatizar pruebas y despliegue, **Git** o **Bitbucket** para control de versiones, **SonarQube** para el análisis estático de código y **Docker** para entornos de prueba estandarizados.

Cada cambio activa un pipeline que compila, prueba, analiza y despliega el código en entornos de prueba.

## RESULTADOS

- **Detección temprana de errores**

Las pruebas automáticas permiten corregir problemas desde las primeras fases, mejorando la estabilidad del código.



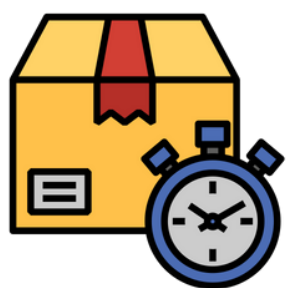
- **Mejor colaboración de equipo**

La CI facilita la integración continua entre desarrolladores, reduciendo conflictos y mejorando la comunicación.



- **Ciclo de desarrollo acelerado**

Los procesos de compilación y pruebas automáticas permiten una entrega más rápida de nuevas funcionalidades.



- **Reducción de riesgos**

La base de código se mantiene confiable, minimizando problemas graves en producción y garantizando cambios seguros.



## CONCLUSIÓN

La integración continua mejora la calidad y velocidad del desarrollo, detectando errores temprano y reduciendo tiempos de entrega. Fomenta una colaboración ágil y un entorno confiable, beneficiando a desarrolladores y usuarios finales.

## REFERENCIAS

**Atlassian.** Integración Continua. **Solving Ad Hoc.** La Integración Continua y cómo aplicarla en la empresa.  
**IBM.** Continuous Integration.  
**Javier Garzás.** Beneficios de la Integración Continua.