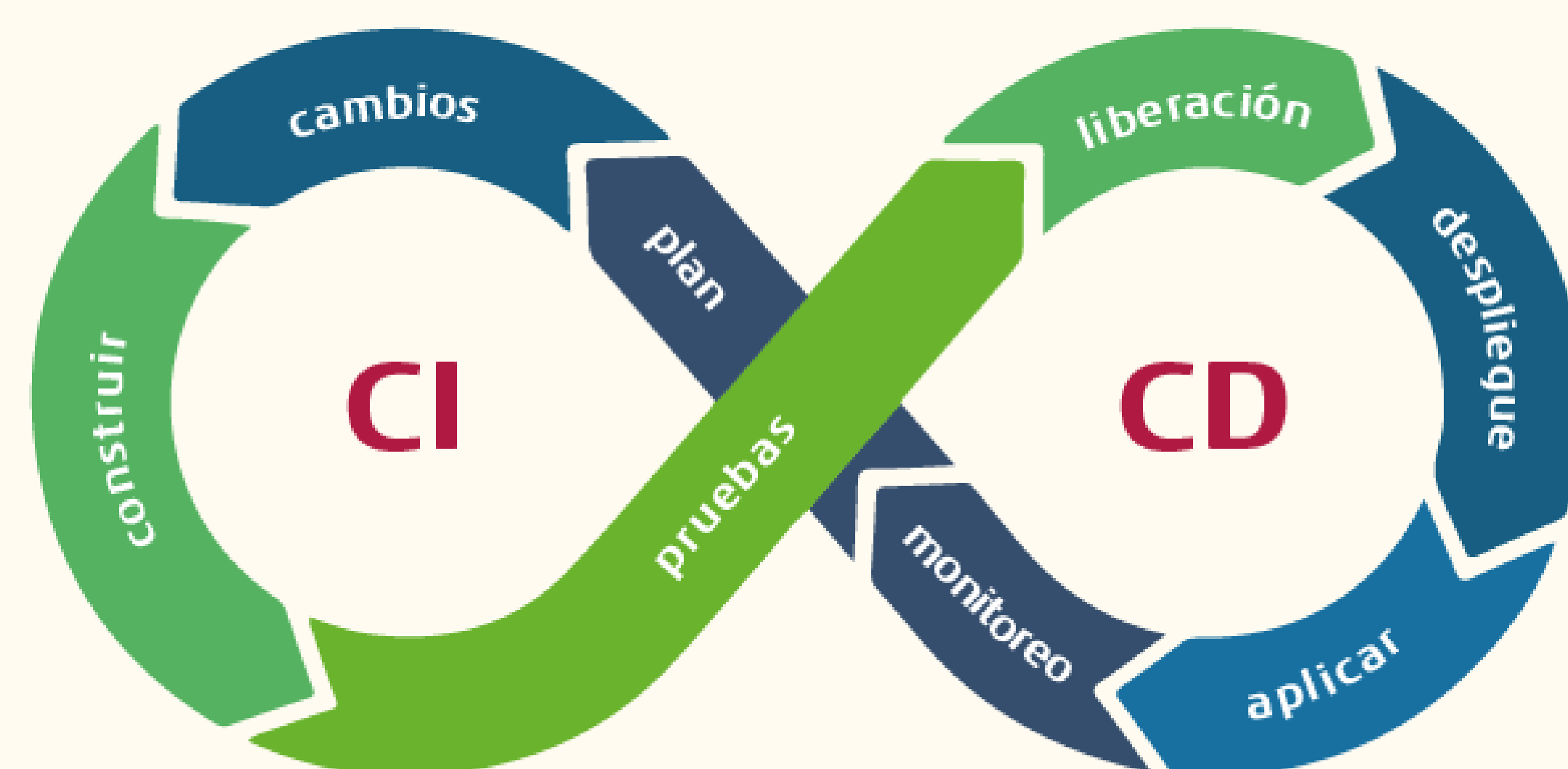








Integración Continua en el Desarrollo de Software

Introducción

La **Integración Continua (IC)** es una práctica de desarrollo de software que permite a los equipos integrar sus cambios de código frecuentemente en un repositorio compartido. Esto ayuda a detectar errores temprano, mejorar la calidad del software y agilizar el proceso de entrega.

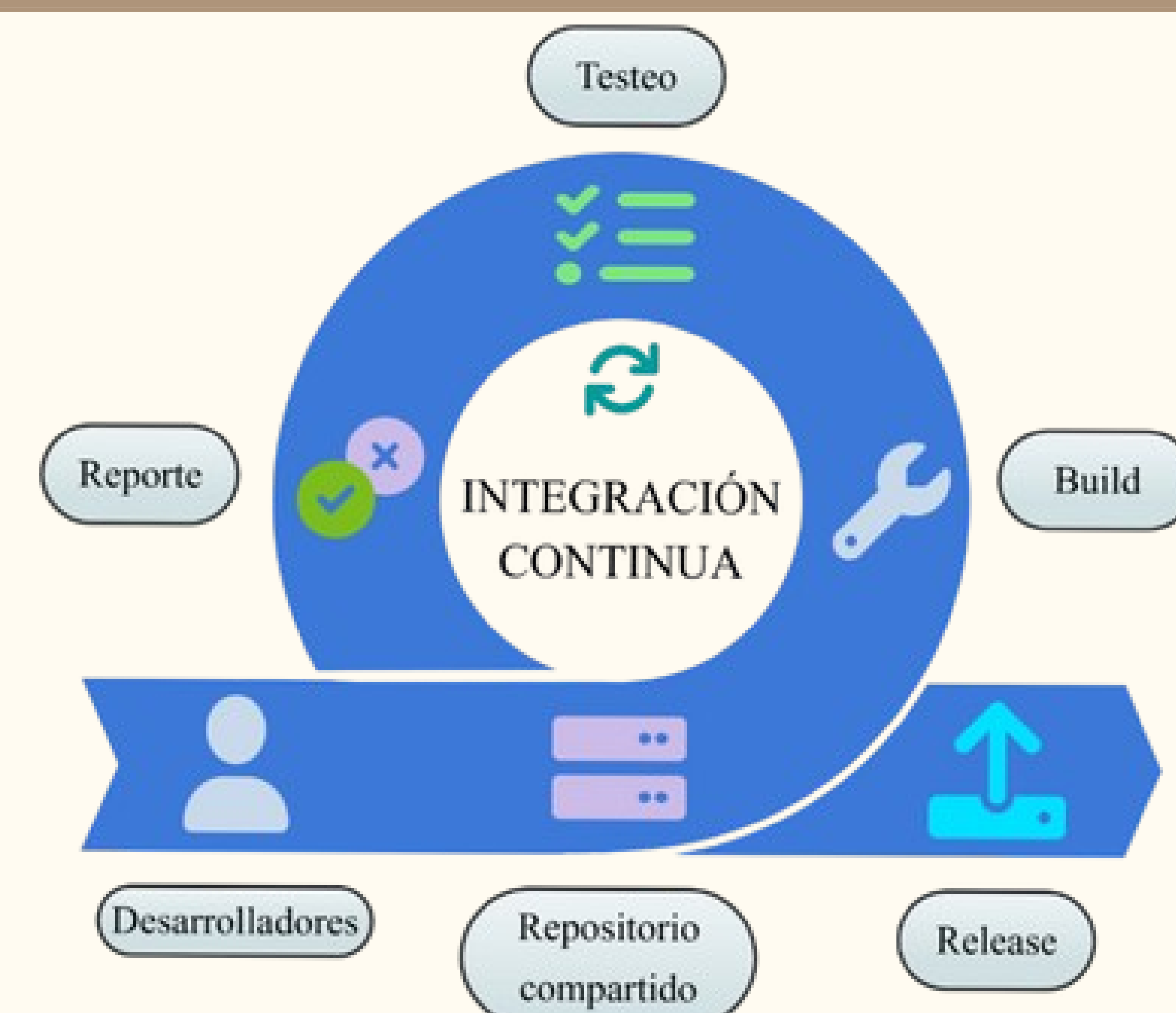


Herramientas

-  • Bitbucket Pipelines
-  • Jenkins
-  • AWS CodePipeline
-  • CircleCI
-  • Azure Pipelines
-  • GitLab

¿Cómo aplicar la Integración Continua?

1. Escribir pruebas para las partes críticas del código base.
2. Obtener un servicio de IC para la ejecución de pruebas automáticamente
3. Integrar a diario los cambios.
4. Corregir la compilación en cuanto se rompa.
5. Escribir pruebas para cada nueva historia que se implemente.



Pilares

- Control de Versiones
- Despliegues Automatizados
- Automatización de Compilaciones
- Pruebas Automatizadas



Resultados

- Permite el escalado
- Mejora el ciclo de retroalimentación
- Mejora la comunicación
- Detección temprana de errores



Conclusión

La **integración continua** es esencial en el desarrollo de software, ya que permite detectar errores rápidamente y mejorar la calidad y eficiencia. Mediante el uso de control de versiones y pruebas automatizadas, IC optimiza el flujo de trabajo, reduce tiempos de entrega y facilita la adaptación a cambios, generando un valor constante.

Referencias



<https://www.atlassian.com/es/continuous-delivery/continuous-integration>
<https://www.ibm.com/es-es/topics/continuous-integration>
<https://aws.amazon.com/es/devops/continuous-integration/>