

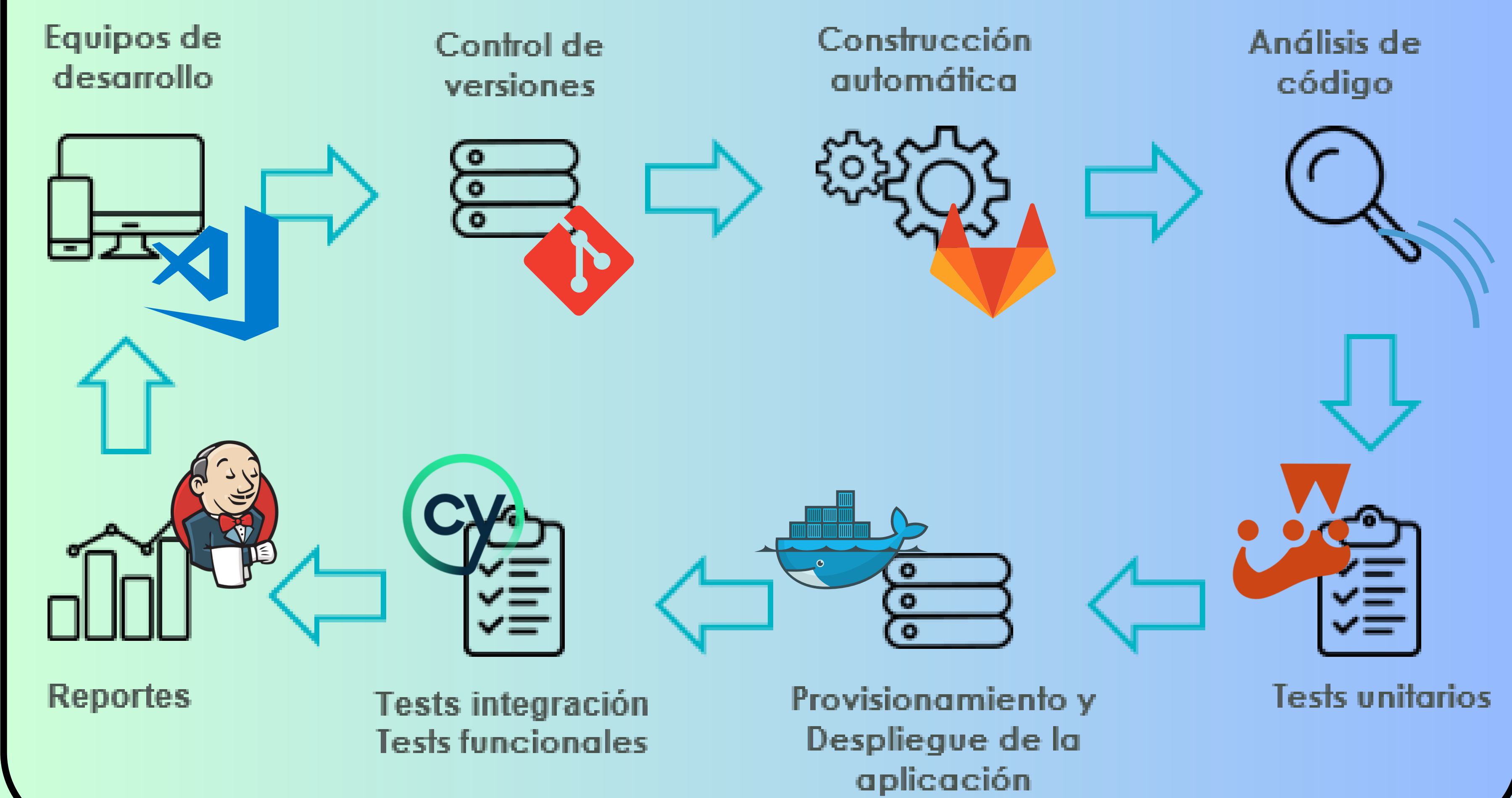
INTEGRACIÓN CONTINUA

Introducción

El primer paso para entregar un software consistente y de alta calidad es la Integración Continua.

Es una práctica esencial que permite a los equipos de desarrollo integrar y probar su código frecuentemente. Este enfoque no solo mejora la calidad del software, sino que también acelera el ciclo de entrega, reduciendo el tiempo y el esfuerzo necesarios para detectar y corregir errores.

Materiales y métodos



Características

Utiliza TDD (Test-Driven Development) como técnica de desarrollo.

Fusiona continuamente las actualizaciones del código fuente de todos los desarrolladores de un equipo en una línea principal compartida.

Es una técnica automatizada.

Se traduce en código de mejor calidad.

El servidor de CI ayuda a reforzar el hábito de testear minuciosamente el código antes de enviarlo.

Conclusiones

La Integración Continua ofrece ventajas significativas en el desarrollo de software, incluyendo mayor eficiencia, calidad y colaboración. Al automatizar pruebas y detectar errores tempranamente, permite entregas más rápidas y fiables. Además, fomenta una cultura de colaboración y mejora continua. Es una práctica esencial en el entorno actual de desarrollo ágil.

Resultados

Mejora la calidad del código

Deteccion de los errores mas rapida y facil

Reduce tareas repetitivas y manuales

Puede crear versiones de pruebas en cualquier momento

Completa viabilidad del proyecto

Mayor confianza y seguridad del equipo de trabajo

Referencias

Continuous Delivery de Jez Humble y David Farley.
GitHub

Documentación GitLab CI/CD

Rossel Sander. Continuous Integration, Delivery and Deployment