

INTEGRACIÓN CONTINUA



Cátedra de Ingeniería de Software

Curso 4K2 - Grupo 1

Barrionuevo, G Floreano, M

Carranza, E Screpnik, J

D'Agostino, J Pellis, L

INTRODUCCIÓN

Las prácticas continuas basadas en un enfoque ágil permiten a las organizaciones lanzar con mayor frecuencia y de forma fiable nuevas funcionalidades y productos

RESULTADOS

ANTES

El código se integraba en una fase de integración del ciclo de vida del software. Esta fase después que partes del equipo trabajaran semanas, meses, años en partes distintas de la aplicación. Esto se lo denominó **EN CASCADA**

HOY

Gracias a las metodologías ágiles, la integración se hizo frecuente y los desarrolladores se integraron con la mayor frecuencia posible, generalmente poco después de que se completa una unidad. Esto se hace en el repositorio de código fuente compartido

La integración frecuente se volvió automatizada y continua, lo que generó la necesidad de algún tipo de verificación antes de que se integre el nuevo código Aquí es donde entra en juego la práctica de: **INTEGRACIÓN CONTINUA**

BENEFICIOS



Mejora la productividad de desarrollo



Permite que se encuentre y arregle los errores con mayor rapidez.



Permite que se entreguen las actualizaciones con mayor rapidez.

MÉTODOS Y MATERIALES

Las herramientas que permiten realizar la Integración Continua, entre otras son:



JENKINS



CODESHIP



TRAVIS CI

PASOS DE IMPLEMENTACIÓN CON DOCKER

- 1- Escribir los tests o pruebas unitarias.
- 2- Configurar la imagen de docker que se va a usar para correr las pruebas
- 3- Correr el contenedor y ejecutar las pruebas unitarias en el contenedor
- 4- Correr las pruebas unitarias en el servidor de integración continua



CONCLUSIÓN

Es posible y altamente recomendable invertir tiempo en mejorar la calidad del software que se construye, teniendo en cuenta que la integración continua es un cambio de paradigma, el cual requiere la aceptación de todos los miembros del equipo, puesto que la integración continua es una práctica y no una herramienta en si.

BIBLIOGRAFÍA

<https://developers.redhat.com/blog/2017/09/06/continuous-integration-a-typical-process/>

<https://www.redhat.com/es/topics/devops/what-is-ci-cd>

<https://medium.com/@juanelojga/continuous-integration-pipeline-docker-eacd3b65955a>

<https://aws.amazon.com/es/devops/continuous-integration/>

[1] Martin Flower, Paul Julius (continuous Integration on Improved Software Quality & Reduced Risk), PAUL M. DUVALL with STIVE MATYAS, ANDREW CLOVER