

La Integración Continua se refiere a la práctica de automatizar tareas de modo que se ejecuten automáticamente cuando se produce un evento, por ejemplo, una nueva versión de código en nuestro repositorio. Ejemplos de las tareas que pueden ser automatizadas son: compilación de componentes, ejecución de pruebas unitarias, ejecución de pruebas de integración, ejecución de pruebas de aceptación, obtención de métricas de calidad de código...

Algunos de los objetivos que persigue la integración continua son:

- Facilitar la integración entre distintos los distintos componentes de nuestra aplicación.
- Automatizar la construcción de nuestra aplicación.
- Automatizar la ejecución de tests en la construcción de nuestra aplicación.
- Desplegar las cambios de nuestro código tan frecuentemente como sea posible.
- Probar en una réplica del entorno de producción.
- Permitir que todo el mundo pueda probar la última versión de nuestra aplicación de una manera sencilla.

La integración continua aporta aporta innumerables ventajas en el objetivo de conseguir un software de alta calidad, algunas de las cuales son:

- Detectar rápidamente posibles conflictos entre desarrolladores.
- Garantizar el correcto funcionamiento de nuestra aplicación antes de desplegar una nueva versión.
- Agiliza la corrección de los errores detectados.
- Permite resolver problemas de integración a lo largo de todo el proceso de desarrollo del software y no sólo al final del mismo, evitando situaciones caóticas previas a la fecha de entrega.

Aunque el concepto de integración continua es muy amplio, en este curso lo vamos a delimitar a:

- **Docker Build:** proceso de construcción de una imagen de docker.
- **Testing con Docker:** como utilizar Docker para facilitar la automatización del testeado de nuestras aplicaciones.