Sin terminar (Alfa)

Integración Continua

¿Que es la Integración Continua?

Para poder hablar de sus ventajas, lo primero es conocer en qué consiste. La integración continua es el nombre que se le da a la automatización de las labores de compilación, test y análisis estático del código. Esto se puede corregir de muchas maneras.

Podemos llamar integración continua a todo lo que hay entre un script que periódicamente ejecuta el trabajo y un servicio online que lo ejecute.

¿Cómo funciona?

Independientemente del sistema que utilicemos para iniciar, el proceso como mínimo seguirá el siguiente camino :

Codigo fuente --> Control de Versiones --> Build Automatico --> Pruebas Automatizadas --> Resultado y Reportes--> Código...

1.Descarga el código fuente desde el repositorio de control de versiones
2.Compilara el código según sea necesario
3.Realizará las pruebas unitarias y/o de integración
4.Publicará los resultados de modo que sean accesibles

(Dependiendo de lo integrado que esté el servicio con el repositorio de código, podemos verlo directamente en el historial
de commits. Por ejemplo en el caso de GitHub)

¿Cuando debería ejecutarla?

Lo ideal, es ejecutarla siempre que se añaden cambios al repositorio principal. Pero eso tiene un coste.

Compilar cada cambio que hacemos nos da una cierta seguridad, pero puede hacer que nuestro sistema de CI trabaje más de

la cuenta y suponga un coste adicional. En cambio, aplicará a todos los pull request nos garantiza que todos los cambios

sobre la línea principal de trabajo van a funcionar bien. Aquí se trata de encontrar un equilibrio entre coste y beneficio.

Si nuestro proyecto es Open Source, existen varios servicios de integración continua online gratuitos como (Travis CI, App

Veyor, Azure Pipelines, Cicle CI)

¿Y qué me aporta?

- Detectar rápidamente los posibles errores de compilación de nuestro código
- Detectar funcionamientos anómalos
- Mejorar la calidad
- Nos permite compilar/testear nuestro código en diferentes plataformas

En conclusión

Con el gran abanico de posibilidades que existen hoy en día para poder añadir integracion continua a nuestro código, no hacerlo no es una opción. Da igual si es un proyecto grande o pequeño, da igual si eres un único desarrollador o un equipo. Existen incluso algunas opciones gratuitas que nos permiten hacerlo.

¿Quién no ha oído nunca "en mi máquina funcionaba"? La integración continua es una solución muy eficaz para evitar este tipo de problemas y poder aportar valor con nuestro software sin tener que perder las horas y horas en compilar nuestro proyecto para varias plataformas, ejecutar pruebas, analizadores de código.

Aplicaciones de Integración

Aplicación a utilizar : Jenkins (Mejor Valorada)

Aplicaciones similares a conocer : CircleCI, Travis CI, Bitrise, Visual Studio App Center, TeamCity, GoCD, Bamboo, GitLab CI,

Jenkins : Es un servidor de Integración Continuam, open-source, escrito en java, el cual nos permite automatizar todos los procesos de integración y entrega mediante tareas,tareas que son extremadamente fáciles de crear y programar. Con Las tareas podemos monitorear cambios sobre un control de versiones, compilar código, ejecutar pruebas, vaya, podemos realizar todos los pasos necesarios para que el software esté listo para su puesta en producción.

Algo interesante a mencionar, es que si en algún momento alguna tarea falla, Jenkins podrá notificar al equipo de desarrollo al product Manager o cualquier responsable del error, de tal forma que se tomen cartas en el asunto y se llegue a una solución lo más pronto posible.

Con Jenkins podemos obtener métricas sobre el software, cantidad de pruebas exitosas, cobertura de código documentación, tiempo de compilación , etc.

Contenido Extra

Integración continua es un tema el cual se ha vuelto muy popular en los últimos años, principalmente, por el impulso que ha

tenido por parte de grandes empresas, tales como Netflix, Uber, entre otros; Las cuales basan su desarrollo en un enfoque

agilista.

Contenido Extra

Al nosotros hablar de integración continua comúnmente, también hablaremos de entrega continua y despliegue continuo.

En esta ocasión estaremos explicando cada uno de estos temas y veremos las ventajas de su implementación.

En muchas ocasiones como desarrolladores estaremos haciendo deploy de nuestras aplicaciones de forma manual; Por ejemplo,

si hablamos de un sitio web, el primer paso es subir todos los cambios a nuestra rama principal, posteriormente ejecutar

todas las pruebas unitarias, si las pruebas son exitosas se procede a compilar el proyecto y subir todos los cambios ha

producción, ejecutar las migraciones y listo, los clientes ya podrán hacer uso de los nuevos features en el sitio web, claro

siempre y cuando ninguno de los pasos anteriores hayan fallado.

Contenido Extra

Todos estos pasos a menudo son tediosos e involucran una cierta cantidad de tiempo, tiempo en el que regularmente el

desarrollador no tiene mucho que hacer. Para evitar estos problemas surge la integración, entrega y despliegue continuo.

·Integración Continua

·Entrega Continua

·Despliegue Continuo

Contenido Extra

Entrega Continua : no es más que una extensión de integración continua. Al realizar un commit sobre la rama principal se

ejecutarán (de manera automática)todos los pasos necesarios para hacer que el proyecto se libere a producción. Se compila

el código, se ejecutan pruebas unitarias, se ejecutan migraciones. Todo esto en un ambiente que no es producción, es decir

entrega continua permite que nuestro proyecto esté listo para que el usuario final haga uso de él, sin embargo, despliegue

a producción no se realiza. Este pequeño y último paso (realizar el deploy)será ejecutado de forma manual, esto con la

finalidad de tener un control sobre cuándo liberar el producto. Habrá ocasiones en las cuales será necesario postergar la

liberación del software, ya sea por estrategias de marketing, petición del cliente, etc...

Contenido Extra

La gran ventaja de Entrega Continua es que nos permite reducir la cantidad de errores al momento de realizar el deploy de la aplicación, puesto que ya todos los pasos para la liberación fueron ejecutados anteriormente.

Podemos concluir que el objetivo de entrega continua no es colocar el producto en producción, sino más bien que el software este disponible para su puesta en produccion en cualquier momento.

Despliegue Continuo : Si en entrega Continua nos quedamos a un solo paso para que el usuario final utilice el producto, con despliegue continuo todo el ciclo de entrega se completa.

Utilizando despliegue continuo al realizar un commit sobre la rama principal, se ejecutan todos los pasos necesarios (de forma automática) para el despliegue del proyecto a producción. Esto se traduce en que el usuario final estará utilizando un producto en constante actualización.

Para implementarlo será necesario haber implementado de forma correcta Integración Continua.