



**UNIVERSIDAD
DE ANTIOQUIA**

**DISEÑO E IMPLEMENTACIÓN DEL PROCESO
DE INTEGRACIÓN CONTINUA PARA
APLICACIONES DE LA UNIVERSIDAD DE
ANTIOQUIA**

Autor(es)

Cristian Fernando Ospina Ospina
Santiago Ruiz Velasquez

Universidad de Antioquia
Facultad de Ingeniería, Departamento de Ingeniería de
sistemas
Medellín, Colombia
2018



Diseño e implementación del proceso de integración continua para aplicaciones de la Universidad de
Antioquia

Cristian Fernando Ospina Ospina

Santiago Ruiz Velasquez

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:
Ingeniero de Sistemas

Asesores (a):

Luis Hernando Silva Florez, Ingeniero de Sistemas

Línea de Investigación:
Development Operations

Universidad de Antioquia
Facultad de Ingeniería, Departamento de ingeniería de sistemas
Medellín, Colombia
2018

Diseño e implementación del proceso de integración continua para aplicaciones de la Universidad de Antioquia

Cristian Fernando Ospina, Santiago Ruiz Velasquez
Departamento de ingeniería de sistemas, Universidad de Antioquia
Medellín, Colombia
cristianf.ospina@udea.edu.co
santiago.ruiz1@udea.edu.co

Resumen— Durante años en el ámbito del desarrollo de software siempre ha sido de preocupación la entrega de nuevas soluciones de una manera ágil y segura, en la búsqueda de una solución a esta necesidad fue surgiendo el término de entrega continua y con el paso del tiempo tomó mucha fuerza hasta hoy en día, que pasó a ser una práctica de desarrollo bastante adoptada en el medio, con la que se busca integrar un conjunto de técnicas con el fin de llevar de manera ágil los cambios en el código de una aplicación para pasar a producción de manera más segura.

La Universidad de Antioquia no es ajena a esta necesidad, por ende en el presente trabajo se busca entregar un marco de trabajo que permita implementar metodologías para hacer entregas permanentes de cambios al software, especialmente integración continua y control de los cambios tanto de aplicaciones móviles como web de manera ágil, encontrando que la implementación de dicho proceso asegura la calidad del software a la hora de pasar a producción y la automatización de este proceso.

Para esto se implementó un servidor con Jenkins el cual nos permite la integración con todo el conjunto de herramientas necesarias para la implementación de esta práctica de desarrollo y siguiendo el marco de trabajo desarrollado se configuraron 2 aplicaciones para adoptarlas en este proceso de integración continua.

Abstract— For years in the field of software development has always been of concern the delivery of new solutions in an agile and secure way, in the search for a solution to this need was emerging the term of continuous delivery and over time took much strength until today, which became a development practice quite adopted in the environment, which seeks to integrate a set of techniques in order to quickly implement changes in the code of an application to move to production more safely.

The University of Antioquia is no stranger to this need, therefore, in this work we seek to provide a framework that allows us to implement methodologies to make permanent deliveries of changes to the software, especially continuous integration and control of changes to both mobile and web applications in an agile manner, finding that the implementation of this process ensures the quality of the software when going into production and the automation of this process.

For this purpose, a server was implemented with Jenkins which allows us to integrate with all the tools necessary for the implementation of this development practice and following the framework developed, two applications were configured to be adopted in this continuous integration process.

I. INTRODUCCIÓN

En la actualidad del desarrollo de software ha surgido una nueva metodología llamada integración continua, tal como la define Martin Fowler, es una práctica de desarrollo en la que los miembros de un equipo integran su trabajo frecuentemente. Generalmente cada persona integra material al menos diariamente, dando lugar a múltiples integraciones por día. Cada integración es verificada por un build automatizado (incluyendo pruebas) para detectar errores de integración lo más rápido posible. [1]

Al encontrar que en la Universidad de Antioquia, tanto en el departamento de ingeniería de sistemas como en el área de gestión informática de la universidad, se están desarrollando aplicaciones tanto web como móviles y estos desarrollos no se hacen con un proceso de integración continua se encuentra una oportunidad de mejora que motiva el desarrollo de este trabajo, el cual se encuentra enfocado en el diseño e implementación de dicho proceso con la herramienta Jenkins, para las aplicaciones anteriormente mencionadas, encontrando las grandes ventajas que conlleva esta implementación facilitando así el proceso de desarrollo y asegurando productos finales con un grado de error mucho menor.

II. PLANTEAMIENTO DEL PROBLEMA

La Universidad de Antioquia cuenta con un área de gestión informática que tiene como misión proporcionar soluciones y servicios informáticos y de telecomunicaciones para el desarrollo adecuado de los procesos y la proyección de la Universidad. [2] Para lograr este objetivo el área de gestión informática tiene a cargo alrededor de 70 aplicaciones las cuales están en constante desarrollo y mantenimiento, el proceso de llevar las nuevas implementaciones desde el ambiente de desarrollo a producción actualmente es completamente manual pues se desarrollan con un proceso tradicional de desarrollo, como se puede ver en la Fig. 1, adicionalmente no se cuenta con un sistema de integración continua ni en el departamento de ingeniería de sistemas ni en el área de gestión informática de la Universidad de Antioquia por lo que surge la necesidad de implementar uno para que muchas de estas operaciones puedan realizarse automáticamente, pero

debido a la falta de conocimiento en este tema y al grueso de personal que debe atender los nuevos desarrollos y ajustes necesarios a las aplicaciones surge la necesidad de un apoyo externo al área de gestión informática el cual se encargue de desarrollar e implementar el proceso de integración continua tanto para aplicaciones móviles y web.

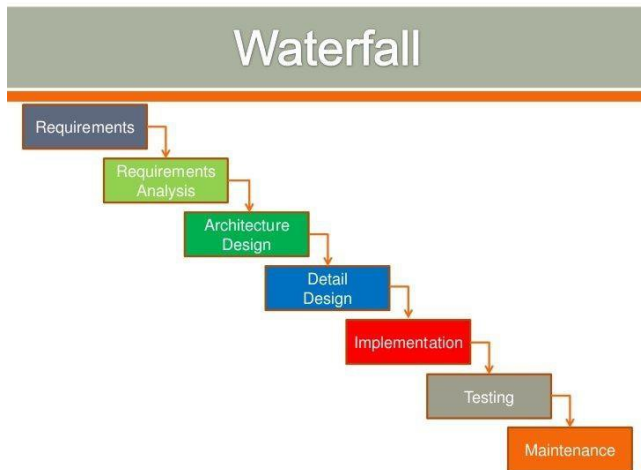


Fig. 1 Proceso de desarrollo tradicional

A. Objetivo general

Diseñar, desarrollar e implementar el proceso de integración continua para aplicaciones web y móviles del área de gestión informática el cual se encargue de llevar los nuevos desarrollos de las aplicaciones automáticamente a producción pasando por un proceso continuo de pruebas que asegure la calidad de las nuevas implementaciones usando un marco de trabajo ágil como Scrum.

B. Objetivos específicos

- Crear marco de referencia de integración continua para aplicaciones web y móviles para el área de gestión informática de la universidad.
- Montar un servidor el cual se encargará de ejecutar el proceso de integración continua de las aplicaciones.
- Implementar el proceso de integración continua en una aplicación web del área de gestión informática que lleve las nuevas implementaciones por un proceso de pruebas automatizadas hasta desplegarla en los servidores de producción.
- Implementar el proceso de integración continua para aplicaciones móviles llevándolas por un proceso de pruebas automatizadas hasta publicar las aplicaciones en la tienda de Google.
- Realizar la documentación necesaria para que en el área de gestión informática puedan continuar aplicando el proceso de integración continua a las aplicaciones del área.

III. INTEGRACIÓN CONTINUA

La integración continua es una práctica de desarrollo de software que busca llevar las actualizaciones de una aplicación desde ambiente de desarrollo a producción de manera automática pasando por unos controles como las pruebas automatizadas con

el fin de mejorar la calidad de la aplicación y evitar posibles errores o anomalías generados por los nuevos desarrollos.

Este proceso comienza cuando los desarrolladores combinan los cambios en el código en un repositorio central tras lo cual se ejecutan versiones y pruebas automatizadas, como lo son las pruebas unitarias, de integración, de servicios web, de seguridad, etc. Los objetivos clave de la integración continua consisten en encontrar y arreglar errores con mayor rapidez, mejorar la calidad del software y reducir el tiempo que se tarda en validar y publicar nuevas actualizaciones de software.

Con la integración continua, los desarrolladores envían los cambios de forma periódica a un repositorio compartido con un sistema de control de versiones como Git [3]. Antes de cada envío, los desarrolladores pueden elegir ejecutar pruebas de unidad local en el código como medida de verificación adicional antes de la integración. El servicio de integración continua detecta los envíos al repositorio compartido y crea y ejecuta de forma automática pruebas unitarias al nuevo código para detectar al instante cualquier error funcional o de integración. [4]

Ventajas:

- Mejora en la calidad del proceso: Con la integración continua se le da más visibilidad al proceso de desarrollo en general, a todos los pasos que se siguen desde que se empieza desarrollar una funcionalidad hasta que está en producción, así, todo el equipo sabe las fases por las que va pasando el código y el estado del software en cada momento (si compila, si pasa las pruebas, en qué entorno está cada versión, qué versión se está probando etc).
- Calidad en el producto: El proceso ayuda a detectar los errores en fases tempranas del desarrollo, para poder solucionarlos rápidamente, así se introducen varios tipos de pruebas y comprobaciones, minimizando los riesgos y haciendo que el software tenga menos bugs.
- Calidad: Al realizar integración continua el equipo acaba aprendiendo a hacer distintos tipos de pruebas (unitarias, de integración), mejores prácticas de programación, en general, a desarrollar código de mayor calidad.
- Disponibilidad constante de una versión para pruebas, demos o lanzamientos anticipados.
- Monitorización continua de las métricas de calidad del proyecto.

A grandes rasgos, el proceso de integración continua comienza cuando el desarrollador hace un commit sobre el repositorio del proyecto, luego el servidor de integración continua detecta el commit, hace un checkout y ejecuta las operaciones de compilación y de pruebas. Si se produjera un error, se genera una notificación al equipo de desarrollo el cual se encargará de corregir el error y luego actualizar el código en el repositorio y al pasar todas las pruebas se publican los reportes de estas, se crea una versión estable de la aplicación y la despliega en ambiente productivo.



Fig. 2 Ciclo de Integración continua

Algunas herramientas que se usan en el proceso de integración continua son: [5]

- Control de Versiones. Requisito fundamental que permite a un equipo trabajar de forma sincronizada. Cada uno trabaja en su parte y puede actualizar los cambios sin necesidad de que otros desarrolladores tengan que esperar por estos y viceversa. En nuestro caso usaremos Git.
- Sistema de construcción automatizado. Muchos equipos utilizan complejos entornos de desarrollo para implementar y compilar los proyectos. Esto en sí mismo no es malo, pero debemos poder construir el proyecto en cualquier máquina sin necesidad de dichos entornos.
- Commits diarios. Una vez que tenemos un control de versiones es recomendable que el almacenamiento de los cambios sea diario y continuo para así tener visibilidad sobre los cambios que se realizan en la aplicación.
- Build diario. También es recomendable tener una versión compilada y probada del sistema al final del día ya que esta nos servirá de backup ante cualquier situación.
- Pruebas unitarias. Nos sirven para comprobar que el proyecto funciona correctamente y que las nuevas funcionalidades no vayan a afectar la aplicación, estas pruebas deben ser automatizadas.
- Herramientas de construcción. Existen diferentes herramientas dependiendo del lenguaje de programación, en Java están Ant o Maven, en Ruby está Rake. En .NET ya existen diversas opciones en el Team Foundation Server.
- Herramientas de integración. Son herramientas que permiten realizar la compilación automática y ejecución de pruebas integradas al control de versiones. La herramienta que usaremos será Jenkins.

Como se dijo anteriormente el servidor de integración que usaremos será Jenkins [6], es un servidor de automatización de código abierto el cual gestiona y controla los procesos de entrega de software a lo largo de todo el ciclo de vida, incluidos la creación, el documento, la prueba, el paquete, la etapa, la implementación, el análisis de código estático y mucho más.

Debido a su facilidad de uso, la base de Jenkins son las tareas o jobs, donde en cada job configuraremos una funcionalidad que se realizará mientras esta tarea se ejecuta, como por ejemplo la construcción del artefacto, cuando en Jenkins tenemos pruebas automatizadas al detectar que el resultado no es el esperado o hay algún error, éste notificará a los desarrolladores de la situación para que lo solucionen.

Es posible configurar Jenkins para que se lance métricas de calidad, visualizar los resultados de dichas métricas, ver el resultado de los tests, generar y visualizar la documentación del proyecto o incluso pasar una versión estable del software al entorno de pruebas para ser probado y luego de estar certificado poder enviarlo al ambiente de producción, todas estas tareas se hacen de una manera muy amigable lo que facilita mucho el proceso de integración continua.

IV. METODOLOGÍA

Por las características del proyecto, se usó para su desarrollo una metodología ágil, específicamente la del framework Scrum.

Tal como lo describen en su sitio oficial, Scrum es un proceso de gestión y control que reduce la complejidad en el desarrollo de productos para satisfacer las necesidades de los clientes. [7]

A la hora de implementar el marco de trabajo se hicieron modificaciones a lo que propone Scrum originalmente, debido a que no teníamos un cliente o Product Owner (conocido en Scrum como P.O), decidimos modificar un poco las ceremonias propuestas por el framework, por lo cual solamente realizamos el desarrollo de los sprints.

Cada sprint se compuso de 3 fases: diseño, construcción y pruebas. En la fase de diseño definimos las tareas específicas que íbamos a realizar en cada sprint, descritas de una manera muy clara y con todas las especificaciones que luego nos iban a servir de insumo para determinar si se realizó de manera correcta o no. En la fase de construcción realizamos las tareas definidas para dicho sprint y a medida que se realizaron las tareas entramos en la fase de pruebas.

Los sprints tuvieron una duración de 1 mes y medio cada uno, en totalidad se realizaron 5 sprints durante los cuales se realizaron las siguientes tareas:

- Sprint 1: Instalación y configuración de la máquina virtual en el servidor Proxmox del laboratorio integrado de sistemas de la Universidad de Antioquia, esto con ayuda de los auxiliares de dicho laboratorio.
- Sprint 2: Instalación de Jenkins en la máquina virtual creada en el sprint 1, como también se realizó la configuración de este servidor de integración continua para poderlo integrar con las herramientas necesarias para tener todo el proceso de entrega continua de las aplicaciones de la universidad.
- Sprint 3: Configuración la herramienta de control de versiones de la Universidad de Antioquia (GitLab) y se integró para su funcionamiento con el servidor de Jenkins, esto con el fin de poder obtener el código fuente de las aplicaciones y poder realizar el proceso de integración continua.
- Sprint 4: Creación de los jobs en el servidor de Jenkins para realizar la compilación automática de las aplicaciones móviles.

- Sprint 5: Creación de los en el servidor de Jenkins para realizar la compilación automática de las aplicaciones web.
- Sprint 6: Se crearon los jobs para realizar las pruebas automatizadas sobre cada uno de los artefactos generados por los jobs de compilación para luego desplegarlos en la tienda de google o en el servidor de aplicaciones para las aplicaciones móviles y web respectivamente.

Durante el desarrollo del proyecto transversalmente a cada sprint, se fue realizando un proceso de documentación, del cual surgieron dos documentos importantes, manual técnico y manual de usuario, los cuales especifican el marco de referencia para la implementación de la integración continua lo cual da pie a que se pueda seguir implementando dicho proceso en futuras aplicaciones desarrolladas en el departamento de ingeniería de sistemas de la Universidad de Antioquia.

Cabe resaltar que durante el desarrollo del proyecto se presentaron diversos inconvenientes. Principalmente fueron inconvenientes más enfocados en la parte del hardware y las herramientas, uno de los problemas más destacables fue que debido a una descarga eléctrica presentada en la Universidad, se borró la máquina virtual que nos habían creado con el servidor Jenkins, lo cual nos retrasó un poco los tiempos de desarrollo, pero gracias a la diligente respuesta del profesor Hernando Silva y de los auxiliares del laboratorio integrado de sistemas se logró recuperar la información que se había perdido.

V. RESULTADOS Y ANÁLISIS

En este trabajo se ha llevado a cabo la implementación de un sistema de integración continua, específicamente la instalación de la herramienta Jenkins, con lo cual se ha creado el marco de referencia necesario para el área de gestión informática de la universidad para los desarrollos de aplicaciones que estos realicen para poner a disposición de la comunidad universitaria.

Se llevó a cabo el proceso de integración continua con una de las aplicaciones móviles desarrolladas por un grupo de estudiantes de la universidad y que hace parte del repertorio de aplicaciones que provee el área de gestión informática a la comunidad universitaria.

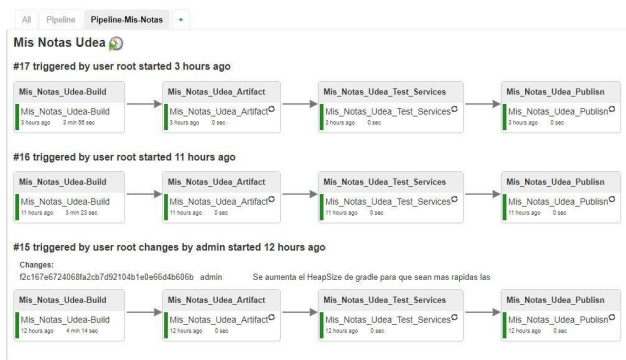


Fig. 3 Flujo de integración continua en Jenkins de una aplicación móvil

Adicionalmente se documenta la forma como se debe llevar a cabo la implementación del proceso de integración continua con una aplicación web, ejecutando las pruebas automatizadas y desplegando en un ambiente ya sea de pruebas o de producción.

Como se mencionó anteriormente se hicieron los documentos, con todos los pasos que se deben seguir, para que el área de gestión informática de la universidad pueda contar con una herramienta que les permita seguir implementando el proceso de integración continua en forma satisfactoria con las nuevas aplicaciones, tanto web como móviles, que se desarrollen dentro de la universidad.

VI. TRABAJO FUTURO

Inicialmente el proceso de implementación de integración continua se iniciará con las aplicaciones desarrolladas dentro del departamento de ingeniería de sistemas, procurando que cualquier proyecto integrador o trabajo de grado que implique el desarrollo de una aplicación ya sea móvil o web sea construido pensando en la implementación del marco de integración continua.

El objetivo a futuro sería que una vez se encuentre más maduro el proceso dentro del departamento, y se tenga una mejor adaptación a este, se pueda escalar al área de gestión informática de la Universidad de Antioquia, para que ya el proceso pase a un ámbito más global al nivel de la universidad y se pueda hacer una mejor capacitación a las personas encargadas del área de gestión gracias a la experiencia obtenida dentro del departamento de ingeniería de sistemas.

VII. CONCLUSIONES

Al evidenciar la importancia y relevancia del proceso de integración continua en el desarrollo de software, y encontrar que en la universidad no hay ningún proceso de este tipo y se desarrollan constantemente aplicaciones web y móviles, se ve la necesidad de realizar esta implementación.

Como se pudo apreciar en la descripción planteada en el marco teórico, una de las mayores ventajas en la integración continua y sobre todo al realizarla con Jenkins es el aseguramiento de la calidad a la hora de querer publicar el producto final elaborado por el grupo de desarrollo, pues es posible realizar la configuración del sistema de forma tal que al realizarse un commit al repositorio del proyecto se ejecuten un conjunto de pruebas dentro de la aplicación, de forma tal que si estas pruebas son exitosas se puedan realizar determinadas tareas, como por ejemplo la publicación de la aplicación en el ambiente de producción y todo esto de manera automática durante todo el proceso.

Gracias a la implementación realizada para el área de gestión informática de la Universidad de Antioquia será mucho más fácil la administración de las aplicaciones que tienen a cargo y de las que desarrollen en un futuro, pues se le facilita la labor a los desarrolladores, que con solamente configurar los jobs de Jenkins con las tareas que sean de ayuda para el objetivo propuesto, con un simple commit en los proyectos, o simplemente ejecutando una tarea desde la plataforma que brinda la herramienta, es posible reducir los tiempos de despliegue en el desarrollo y centralizar dicha labor, inclusive, se logra tener un reporte de cuando al ejecutar alguna tarea o al realizar algún commit, se detecta un error permite que sea visualizado a través de un correo electrónico de forma inmediata.

Anteriormente en el área de gestión dichas labores eran mucho más engorrosas, pues requerían que un desarrollador ejecutara manualmente las pruebas unitarias desarrolladas y adicionalmente generará los aplicativos para publicar en el ambiente ya sea de pruebas o producción por lo que con la implementación de la integración continua se logra una mejora en este proceso del área de gestión.

Uno de los resultados de este trabajo de grado, y que es de gran importancia, es la creación de un manual técnico en el cual se especifican los pasos que se deben seguir para la implementación e instalación de Jenkins en el servidor provisto por el programa de ingeniería de sistemas, además, en dicho manual también se especifica cómo pueden crearse más tareas en el Jenkins para suplir nuevas necesidades que surjan con el transcurso del tiempo.

REFERENCIAS

- [1] Martin Fowler. (2018). Continuous Integration. [online] Available at: <https://martinfowler.com/articles/continuousIntegration.html>
- [2] Universidad de Antioquia. (2018). Dirección de Planeación y Desarrollo Institucional. [online] Available at: http://www.udea.edu.co/wps/portal/udea/web/inicio/institucional/unidades-administrativas/direccion-planeacion-desarrollo-institucional/contenido/asmenuateral/acerca-direccion!/ut/p/z1/1ZRNb-lwElb_Chw4RnY-n19jIKZFlEAopCW5rIxtwN3ETHPDdVfXr4O6IUAtbLXisD555Hce2e94BhRgCQqB93yDFZcClzrOC_fbZIZi04rg-M5bBDCMU-g59sy7d13wdBD4QWSZoQPH0-GtD0PXCQM7mJrz2ATF1_LhvYlgOlvTyWlapTcj6-_yzwi6fPjJCiF4BAUoiFC12oK8lo3C5Y4yPIC4PY62smJ_9ly0iqsdOXg0gDvBKaaas7WFacFb1Wj79rgdQMobRgijvE7u_skVPa3DTyLKUvRMEkUlxjZEdvmJiV2LFmu4EE9YQbLyRpOjuWxNOQU4D13aQywyXYstwkEUNjCkxqGki5KxXkFD05s8ZAY76Wxwk7xW6exhFusJRGHvzRxNG7qnggy9wLPChf6trvBjHTpZAOEWngg8-waUy5vqZ3qfvSEww_6pve85-gEzIptJtcB4fw-vi3evi7avis-uak6F_xl8udYAegfz55aU19ZzouvRVgeV_MSj0xa0miZKNdgercHFwLlOwAsjwA6PAUMlairLmsq3_5pFH-115OJz_DK8aH9iur986_FelhVRpoolKeb_m8Rik7X/dz/d5/L2dBISevZ0FBIS9nQSEh/?uril e=wcm%3Apath%3A%2FPortalUdeA%2FasPortalUdeA%2FasHomeUdeA%2FasInstitucional%2FUnidades%20administrativas%2FDirecci3!b3n%20Desarrollo%20Institucional%2FContenido%2FasMenuLateral%2Facerca-direccion
- [3] GIT. (2018). [online] Available at: <https://git-scm.com/>
- [4] AWS Amazon. (2018). ¿Qué es la integración continua?. [online] Available at: <https://aws.amazon.com/es/devops/continuous-integration/>
- [5] Yeray Darias Camacho. (2017). Integración continua. [online] Available at: <https://emanchado.github.io/camino-mejor-programador/html/ch08.html>
- [6] Jenkins. (2018). [online] Available at: <https://jenkins.io>
- [7] Joel Francia. (2018). ¿Qué es Scrum?. [online] Available at: <https://www.scrum.org/resources/blog/que-es-scrum>