INTEGRACIÓN CONTINUA
ENTREGA CONTINUA
DESPLIEGUE CONTINUO

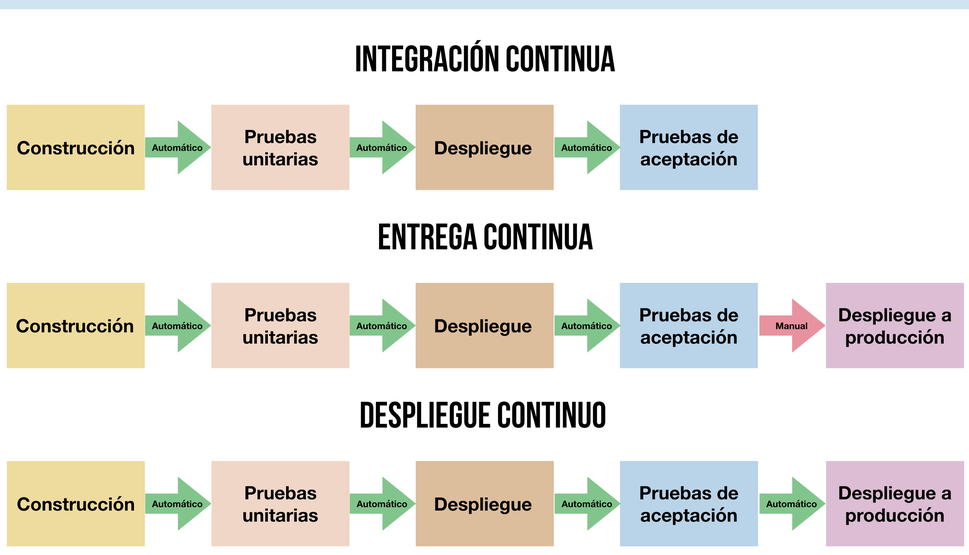
INGENIERÍA DE SOFTWARE

UNIVERSIDAD TECNOLÓGICA NACIONAL

GRUPO 3. ACOSTA. MUÑOZ. ROMERO. SANTI.

INTRODUCCIÓN

La **integración continua**, e**ntrega continua** y **despliegue continuo** son prácticas de desarrollo de software donde cada una pone énfasis en distintos aspectos de la construcción del software.
En la integración continua, se busca combinar los cambios del código en un repositorio frecuentemente, para lo cual se ejecutan versiones y pruebas automáticas.
En la entrega continua se crean, prueban y preparan automáticamente los cambios en el código de tal manera que puedan ser liberados en la fase de producción.
En el despliegue continuo cada cambio que pasa todas las etapas de su línea de producción se lanza a sus clientes, sin ningún tipo de intervención humana.
El **objetivo** de este trabajo es analizar cada una de estas prácticas y mostrar los beneficios que aportan su aplicación.



MÉTODOS

Se efectuó un análisis los siguientes informes:
- **“Continuous Integration vs. Continuous Delivery vs. Continuous Deployment”**, Sten Pittet.
- **“Integración Continua, Entrega Continua y Despliegue Continuo. ¿Qué, cuándo y cómo?”**, Francisco Quintero.
- **“Integración Continua (CI), Entrega Continua (CD) y Despliegue Continuo (CD)”**, Oscar Faura.

Y se realizó una comparación entre cada una de las técnicas citadas en estos mismos, obteniendo los requisitos necesarios para cada una y los beneficios que estas brindan.

RESULTADOS

Con base en lo analizado se lograron obtener los beneficios que aportan la implementación de cada una de estas prácticas, aunque se debe cumplir con ciertos requisitos para poder aplicarlas de forma correcta.
Cabe destacar que el despliegue continuo y la entrega continua son una extensión de la integración continua, es decir, que si se quiere aplicar alguna de las dos primeras, evidentemente se debe tener una base sólida proveniente de la última.

Práctica	Requisitos	Beneficios
Integración Continua	- Escribir pruebas automatizadas para cada cambio. - Se deben combinar los cambios de forma frecuente (una vez al día como mínimo). - Instalación y mantenimiento de un servidor de integración continua que monitoree los cambios.	- Disminuye los errores y evita que estos se manden a producción. - Desarrollo rápido de versiones. - Disminuye el costo de las pruebas. - Mejora el control de calidad. - Evita procesos manuales engorrosos. - Mejora visibilidad del estado del software. - Logra seguridad y confianza en el equipo de desarrollo.
Entrega Continua	- Las implementaciones requeridas deben ser automatizadas.	- Mayor velocidad y menor complejidad en el proceso de desarrollo. - Los errores son más fáciles de encontrar. - Entregas más seguidas, con retroalimentación inmediata. - Automatiza el proceso de publicación del software.
Despliegue Continuo	- Actualización continua de la documentación. - Pruebas de buena calidad.	- Los cambios se ejecutan de manera automática. - En cada versión se disminuyen los riesgos. - Flujo continuo y de buena calidad.

CONCLUSIÓN

Llevar a cabo la implementación de estas técnicas puede implicar ciertos costos, incluyendo la adaptación del equipo a un nuevo enfoque, pero en la praxis queda demostrado que cuando las mismas son integradas de manera correcta, se pueden alcanzar grandes beneficios, poniendo en manos del cliente, en el menor tiempo posible y con la mayor calidad, las nuevas funcionalidades implementadas y garantizando las funcionalidades anteriores.

Para más información: