

INTEGRACIÓN CONTINUA VS. ENTREGA CONTINUA VS. DESPLIEGUE CONTINUO



AUTOR

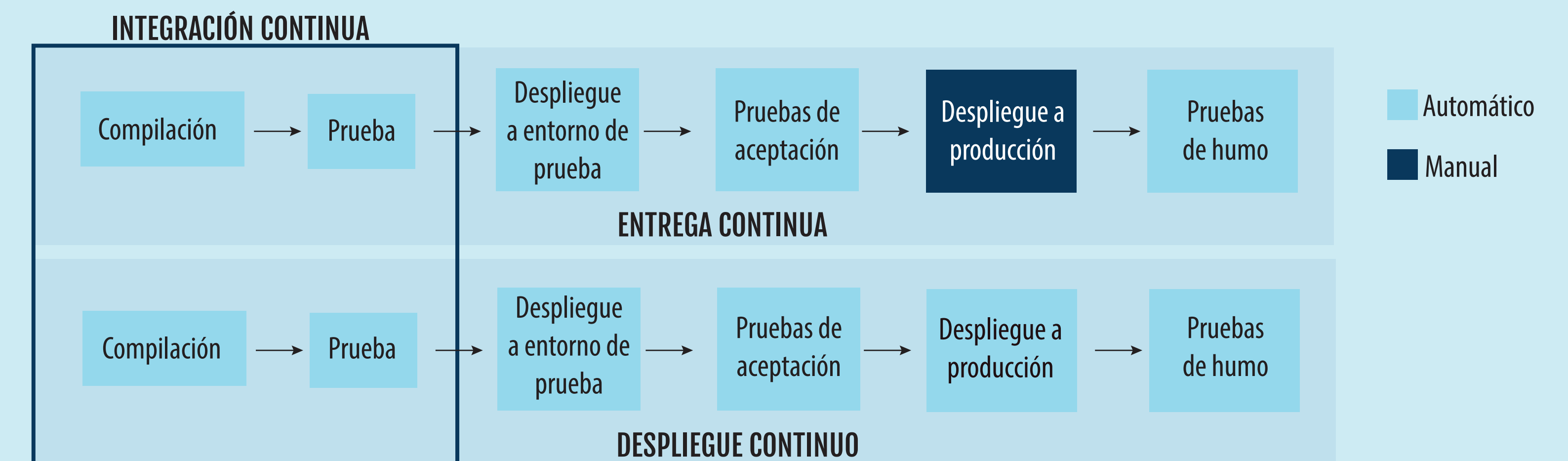
Sten Pittet - Atlassian

INTRODUCCIÓN

Si se habla de prácticas modernas de desarrollo, generalmente se mencionarán la **Integración Continua**, la **Entrega Continua**, el **Despliegue Continuo** y qué se requiere para usarlas.

El Despliegue Continuo es como la Entrega Continua, excepto que las versiones se producen automáticamente y, a su vez, estas dos técnicas son extensiones de la Integración continua.

Se sabe que la implementación de cada práctica tendrá un costo, pero sus beneficios lo compensan en gran medida.



INTEGRACIÓN CONTINUA

Los desarrolladores que practican la Integración Continua fusionan sus cambios a la rama principal **tan a menudo como sea posible**. Los cambios del desarrollador son validados al crear pruebas automatizadas que se ejecutan contra lo construido.

Al hacerlo, se evitan los problemas de integración que suelen presentarse cuando se espera hasta el día del *release* para fusionar todos cambios.

La Integración Continua pone énfasis en la automatización de las pruebas para verificar que la aplicación no se rompa cuando se integren nuevas actualizaciones en la rama principal.

ENTREGA CONTINUA

La Entrega Continua es una extensión de la Integración Continua que se asegura de poder presentar nuevos cambios a los clientes de forma rápida y sostenible.

Esto significa que, además de contar con pruebas automatizadas, también se ha automatizado el proceso del *release* y se puede desplegar la aplicación en cualquier momento haciendo clic en un botón.

Los despliegues deben ser automatizados. Si bien activar el despliegue sigue siendo manual, una vez que se inicia, no debería ser necesaria la intervención humana.

DESPLIEGUE CONTINUO

El Despliegue Continuo va un paso más allá de la Entrega Continua. Con esta práctica, cada cambio que pasa todas las etapas de la línea de producción es un *release* para los clientes. No hay intervención humana, y sólo una prueba fallida podrá evitar que se despliegue un nuevo cambio en producción.

El Despliegue Continuo es una excelente manera de acelerar el ciclo de retroalimentación con los clientes y quitar presión sobre el equipo, ya que ya no hay un día límite para el *release*. Los desarrolladores pueden enfocarse en construir software y ver su trabajo en funcionamiento minutos después de haber terminado de trabajar en él.

CONCLUSIÓN

A la hora de pensar en aplicar estas prácticas, se debe tener en cuenta que si recién se está empezando con un proyecto que no tiene usuarios, puede ser fácil desplegar cada actualización a la producción.

No obstante, si ya se cuenta con una aplicación con clientes, se debería comenzar por aplicar la Integración Continua y la Entrega Continua. Primero se empieza por implementar pruebas unitarias básicas automáticas y automatizar los despliegues lo antes posible. Una vez que se puedan realizar *releases* de software diariamente, será posible mirar hacia el Despliegue Continuo, aunque hay que asegurar que el resto de la organización también esté preparado para este cambio.



Costo

El equipo deberá escribir pruebas automatizadas para cada característica nueva, mejora o corrección de errores.



Beneficio

Se envían menos errores a producción a medida que las pruebas automáticas capturan de forma temprana las regresiones.



Costo

Se necesita una base sólida en la Integración Continua y el conjunto de pruebas necesita cubrir todo el código base que sea necesario.



Beneficio

Permite hacer *release* más a menudo, acelerando así el ciclo de retroalimentación con los clientes y de esta forma solucionar fácilmente problemas que se pudiesen presentar.



Costo

La cultura de prueba del equipo debe ser muy buena. La calidad del conjunto de pruebas determinará la calidad de los *release*.



Beneficio

Se puede desarrollar más rápido, ya que no hay necesidad de pausar el desarrollo por los *release*. El proceso de despliegue se activa automáticamente para cada cambio.

Ingeniería
de Software
4k2

UTN
Facultad
Regional
Córdoba

Grupo 4

Aicardi, Anabella
López G, Juan Manuel
Pignata, Camila
Rassow, Tiziana
Tresca, Angelina