



Facultad Regional Córdoba

ISW-GRUPO 4

Integrantes:

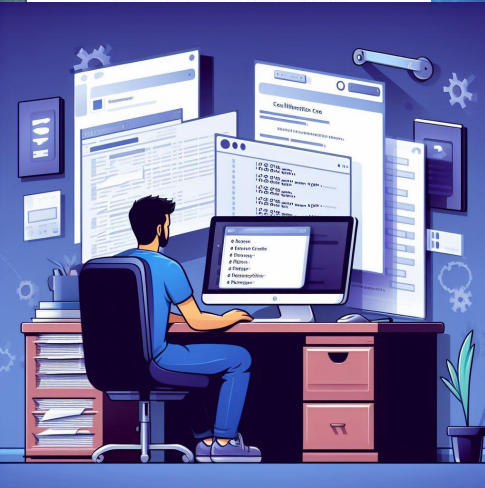
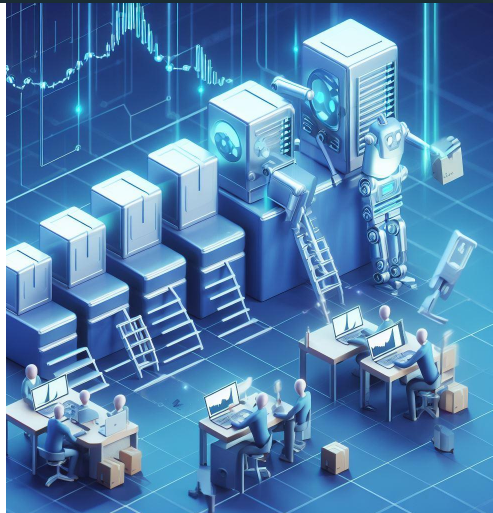
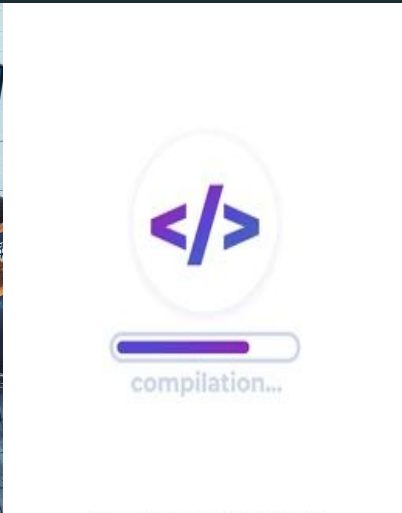
Yamili Yarbouh - 81841

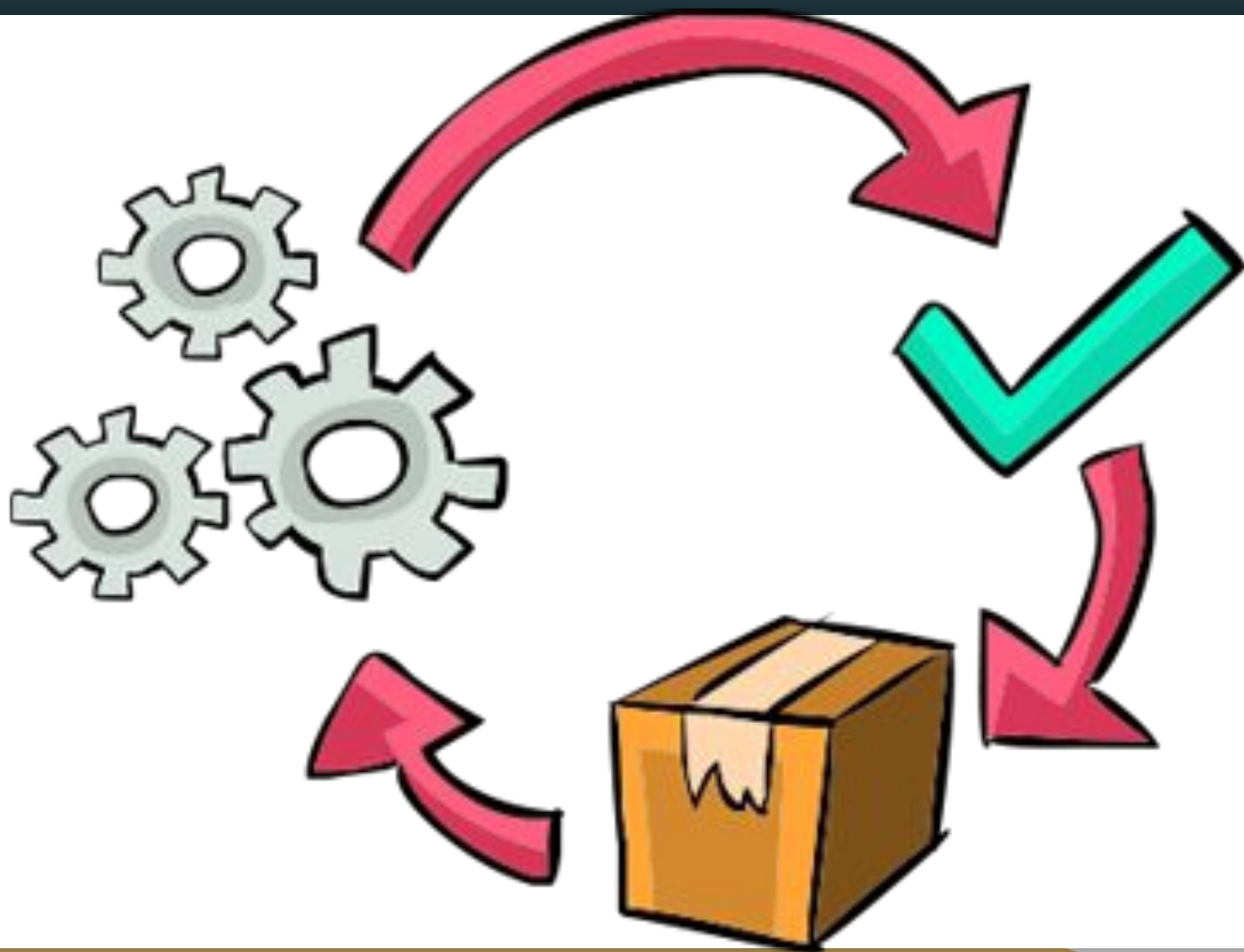
Daysi Camargo - 84200

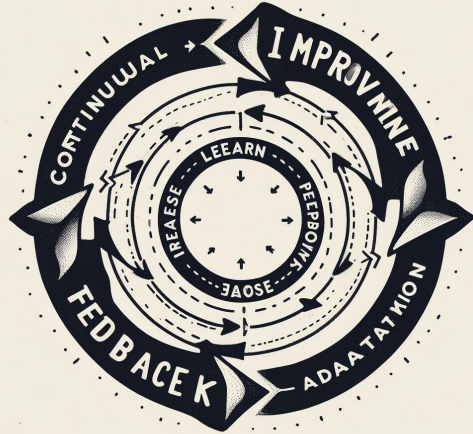
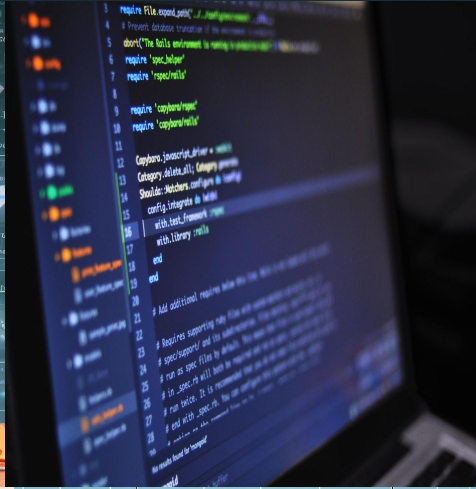
Sonzini Astudillo, Enrique José - 82508

Muñoz Jonathan 77915

Prácticas continuas - Entrega continua









Introducción

Hola, nosotros somos el grupo cuatro y les vamos a hablar sobre la entrega continua o continuous delivery, primero definiremos qué es la entrega continua, algunos requisitos claves que se tienen que cumplir para poder implementarla, sus fases y las ventajas y desventajas que tiene su implementación.

¿Qué es la entrega continua o continuous delivery?

La entrega continua consiste en la automatización del proceso de entrega del software, permitiendo que pueda ser implementado en producción de forma confiable y sencilla. De forma práctica se la puede entender como la entrega de actualizaciones de software a los usuarios o clientes de forma sólida y continua.

Para que la entrega continua sea exitosa es necesario cumplir con ciertos requisitos clave, estos son:

- Una base sólida de integración continua, esto quiere decir que el proceso de combinar el código de diferentes desarrolladores en un solo repositorio debe estar bien automatizado.
- Un entorno de pruebas que cubra una gran cantidad de código base, esto incluye el código nuevo, el código modificado y el código existente.
- Que se haya utilizado la misma aplicación de compilación a través de cada etapa del proceso, para de esta manera garantizar que las pruebas de etapas anteriores son confiables.
- Las implementaciones que se requieren deben ser automatizadas. Aunque se activa de forma manual, una vez iniciado el proceso no se necesita de la intervención humana.
- Los desarrolladores deben ser responsables de mantener el proceso de entrega automatizado y de solucionar cualquier problema que surja.
- Mantener todos los archivos de configuración en el control de versiones para ayudar a asegurar que las entregas sean coherentes y repetibles, es decir, no olvidarse de subir los ítems de configuración al repositorio para después no tener problemas a la hora de querer hacer la implementación.

Fases

Hay cinco fases que componen a la entrega continua. La primera es la de la implementación continua. ¿Cuántos de ustedes trabajaron en un mismo documento de google para completar los trabajos prácticos de ARE/AS? Eso es el equivalente de ARE para lo que llamamos **implementación continua**. Se trata de tener un repositorio en común con la versión del producto funcional más actual ¿Cuántos de ustedes usaban el corrector ortográfico de google para corregir sus errores de ortografía? Eso es el equivalente a la segunda fase: **pruebas continuas y automatizadas**. ¿Y cuántos usaban la opción de descargar como PDF que viene en google drive para poder subirlo a moodle sin problemas? A eso lo llamamos

automatización del despliegue. Es la tercera fase y busca automatizar ciertos aspectos de la entrega del producto para no tener que estar haciendo cosas relacionadas a la compatibilidad. ¿Cuántos vieron sus notas y cuántos se sacaron menos de ocho? La nota de los TPs se corresponde con los reportes de la cuarta fase, **mantenimiento y retroalimentación**. El sistema, ya en fase de despliegue, es revisado continuamente por algún sistema automatizado -el profe en nuestro caso- y en caso de encontrar errores se informa a los responsables por los medios adecuados. Para corregirlos, recurrimos a la quinta fase, **ciclos**. Agarramos el TP preexistente, NO lo eliminamos, cambiamos lo que hay que cambiar, agregamos lo que hay que agregar y subimos a moodle -el despliegue- la nueva versión. Lo mismo ocurre en el software, ya que no eliminamos el programa en sí, sino que manipulamos funciones específicas para largar la segunda versión.

Beneficios principales de la implementación

- **Lanzamientos más rápidos:** La entrega continua permite aplicar nuevas versiones de software de manera rápida y constante. Al automatizar gran parte del proceso de lanzamiento, se reducen los tiempos de espera.
- **Mayor calidad del software:** Fomenta la implementación de pruebas automatizadas exhaustivas. Esto garantiza una mayor calidad del software, ya que se pueden detectar y corregir errores de manera temprana en el ciclo de desarrollo, antes de que lleguen al entorno de producción.
- **Mejora continua:** La entrega continua tiene una orientación iterativa, lo que permite aprender de cada lanzamiento y retroalimentación recibida. Esto ofrece la oportunidad de mejorar continuamente el software y los procesos de desarrollo.
- **Mejora la productividad del equipo:** Se los libera de tareas manuales, permitiendo dedicar más tiempo a agregar nuevas funcionalidades que generen valor a la empresa.

Desventajas principales de la implementación

- **Costo y esfuerzo:** La implementación puede ser costosa y requerir un esfuerzo significativo. Las organizaciones deben estar dispuestas a invertir en la infraestructura y recursos necesarios para implementar CD de manera exitosa.
- **Falta de madurez de los procesos y herramientas:** Cuando se implementan por primera vez a menudo se detecta que hay procesos y herramientas que no están listos para la automatización. Esto puede conducir a problemas de calidad.
- **Necesidad de una cultura de cambio:** Requiere que los equipos de desarrollo y operaciones trabajen juntos de manera más estrecha. Además, se debe gestionar adecuadamente el estrés en los equipos que se puede generar por la presión de entregas frecuentes.

Cierre

Como conclusión podemos decir que la entrega continua ofrece una serie de ventajas significativas, como lanzamientos más rápidos, mayor calidad del software, mejora continua y mejora de la productividad del equipo. Sin embargo, su implementación puede ser costosa y requerir un esfuerzo significativo. Además, requiere una cultura de cambio y la madurez de los procesos y herramientas. Pero, si como organización estás buscando lanzar nuevas versiones de software de manera más rápida y eficiente, garantizar una mayor calidad del software, mejorar continuamente el software y los procesos de desarrollo y liberar a los equipos del desarrollo de tareas manuales. Entonces, la entrega continua es una opción que vale la pena considerar.

Gracias por su atención.

Bibliografía:

https://sentry.io/blog/la-entrega-continua-practicas-fases-y-beneficios/#Beneficios_principales

<https://www.ilimit.com/blog/integracion-continua-entrega-continua-despliegue-continuo/#entrega-continua>

<https://www.aplyca.com/blog/integracion-continua-y-entrega-continua-cicd#heading-2>

<https://recluit.com/que-es-continuous-integration-continuous-delivery/>

<https://www.jetbrains.com/es-es/teamcity/ci-cd-guide/continuous-integration-vs-delivery-vs-deployment/>