**Temas que tenemos que abordar.**

**Contexto (Identificar la universidad, cátedra, grupo, etc.)**

**Tema abordado en la charla**

El tema abordado en la charla es "Despliegue Continuo (CD - Continuous Deployment)" en el contexto de la ingeniería de software. La charla explora en detalle en qué consiste el Despliegue Continuo, por qué es importante, cómo funciona, sus beneficios, desafíos y las herramientas esenciales para llevarlo a cabo con éxito. También destaca cómo el Despliegue Continuo está transformando la forma en que se desarrolla y entrega el software en la actualidad.

**Introducción cautivadora: Atrae la atención de la audiencia desde el principio con una anécdota, estadística impactante o pregunta provocadora.**

¿Alguna vez te has preguntado por qué las actualizaciones de tus aplicaciones favoritas parecen llegar tan lentamente? ¿O por qué a veces sientes que estás atrapado en una especie de bucle temporal donde el software nunca mejora? Bueno, hoy te sumergiremos en un emocionante viaje a través del mundo del Despliegue Continuo, donde los cambios de código se convierten en actualizaciones mágicas que llegan a tus dispositivos de forma casi instantánea. Así que, prepárate para descubrir cómo el futuro del desarrollo de software está literalmente a solo un clic de distancia.

Imagina por un momento que eres el dueño de una empresa de software. Tu equipo ha estado trabajando incansablemente en una nueva actualización de tu aplicación estrella. Pero, hay un problema: tienes que esperar días, semanas o incluso meses antes de que esa actualización llegue a tus valiosos clientes. ¿No sería genial poder acelerar ese proceso y llevar las mejoras directamente a tus usuarios de manera casi mágica? Bueno, esto no es magia, es el Despliegue Continuo, una estrategia revolucionaria en el mundo del desarrollo de software que nos permite hacer precisamente eso.

Imagina que el desarrollo de software es como tener una hoja de papel doblada. El Despliegue Continuo es como desdoblar esa hoja: en lugar de hacer cambios lentos y tediosos, con el Despliegue Continuo, los cambios se despliegan rápidamente, como desplegar una hoja para revelar instantáneamente una nueva sección. Esto significa que las actualizaciones llegan ágilmente a los usuarios, eliminando la espera y acelerando la entrega de innovaciones.

Esta analogía simplificada ilustra cómo el Despliegue Continuo agiliza el proceso de desarrollo de software, permitiendo cambios rápidos y eficientes, similar a desdoblar una hoja para mostrar nueva información.

**Una broma sobre el tema elegido para reforzar la atención**

Hablando de magia, el Despliegue Continuo es como el David Copperfield de la ingeniería de software. ¡Presto, cambio y mejora! En lugar de esperar y esperar como en un truco de magia anticuado, tus actualizaciones aparecen como por arte de magia en las manos de tus usuarios. ¡Y todo esto sin sombreros altos o conejos peludos! ¡Ah, la magia del código!

Y hablando de clics, el Despliegue Continuo es como el "ctrl+c" y "ctrl+v" de la ingeniería de software, ¡pero llevado a otro nivel! En lugar de copiar y pegar, es más bien como "ctrl+c" y "¡magia, aquí tienes tu actualización!". Olvida esperar a que las ruedas de desarrollo giren eternamente; con el Despliegue Continuo, tus usuarios obtienen lo último y lo mejor sin tener que esperar en una fila virtual. ¡Es como la versión tecnológica de un truco de magia bien ejecutado!

**Desarrollo: Divide tu mensaje en segmentos lógicos. Utiliza ejemplos, anécdotas personales y datos relevantes para respaldar tus puntos. Mantén cada segmento conciso y coherente.**

Segmento 1: Despliegue Continuo, la Revolución en el Desarrollo de Software

- Comencemos nuestro viaje en el mundo del Despliegue Continuo. Imagina que eres el responsable de una aplicación de mensajería, y tienes una nueva función brillante en mente. Antes del Despliegue Continuo, este proceso podía ser lento y tedioso, con largos plazos de espera. Pero, con esta estrategia, los cambios de código se publican automáticamente en el entorno de producción, eliminando ese desfase entre la programación y el valor al cliente. Esto significa que tu última idea genial puede llegar a los usuarios en un abrir y cerrar de ojos.

Segmento 2: ¿Qué Gana tu Empresa con el Despliegue Continuo?

- Entonces, ¿por qué debería importarte a ti o a tu empresa? Pues bien, agilizar el tiempo de comercialización es solo el comienzo. Esto se traduce en una mayor competitividad, ya que puedes responder más rápido a las necesidades cambiantes del mercado. La automatización de las pruebas de regresión reduce los costos y los errores. Los procesos de planificación y aprobación de publicaciones, que solían ser un dolor de cabeza, se vuelven obsoletos.

Segmento 3: La Evolución hacia la Automatización Total

- El Despliegue Continuo no es solo un paso en el proceso; es una evolución natural de la entrega continua. Al eliminar la aprobación manual, aceleramos aún más el proceso. Las actualizaciones se envían automáticamente una vez que pasan las pruebas, lo que garantiza un flujo constante de mejoras. Imagina una línea de ensamblaje en la que cada pieza de código nuevo se coloca en su lugar sin un operario humano. Eso es el Despliegue Continuo en acción.

Segmento 4: El Papel Fundamental de la Integración Continua (CI)

- Ahora, para que esta automatización funcione sin problemas, necesitamos un sistema eficaz de comunicación entre los desarrolladores. La Integración Continua, o CI, es el pegamento que une todo. En lugar de que los desarrolladores trabajen en copias individuales del código, todos fusionan sus cambios en un repositorio al menos una vez al día. Esto garantiza que los cambios sigan siendo compatibles con la rama principal. De esta manera, evitamos problemas de integración y aseguramos una transición sin problemas.

Segmento 5: Las Herramientas Esenciales del Despliegue Continuo

- Finalmente, para llevar a cabo esta revolución, necesitas las herramientas adecuadas. El control de versiones, las revisiones de código, la Integración Continua, la gestión de configuración y la automatización de releases son elementos esenciales para un flujo de trabajo eficaz. Estas herramientas no solo mejoran la colaboración, sino que también minimizan los errores en el proceso de entrega.

Segmento 6: La Supervisión de Infraestructuras, la cereza del postre

- Para redondear todo el proceso, la supervisión de infraestructuras es clave. Permite visualizar el rendimiento de las aplicaciones, asegurándote de que los cambios tengan un impacto positivo. Es como tener una ventana mágica para ver cómo tus actualizaciones afectan al mundo real.

En resumen, el Despliegue Continuo es una revolución en el desarrollo de software que elimina la espera y pone las actualizaciones en manos de tus usuarios más rápido que nunca. Con la Integración Continua y las herramientas adecuadas, el futuro del desarrollo de software está listo para ser más rápido, eficiente y mágico que nunca antes.

**Conclusiones: Resuma el mensaje central y brinda una llamada a la reflexión para que la audiencia se lleve algo valioso.**

En este emocionante viaje a través del Despliegue Continuo, hemos descubierto una estrategia que está transformando la forma en que desarrollamos software. Hemos visto cómo el Despliegue Continuo automatiza la entrega de actualizaciones, acorta los plazos de desarrollo y mejora la calidad del software que llega a los usuarios.

Pero más allá de la eficiencia y la velocidad, lo más valioso es la capacidad de adaptación. El Despliegue Continuo nos permite responder rápidamente a las cambiantes demandas del mercado y a las necesidades de nuestros usuarios. Nos libera de la espera y nos permite enfocarnos en la innovación.

Así que la próxima vez que veas una actualización de tu aplicación favorita, recuerda que detrás de esa pequeña notificación hay un mundo de automatización, colaboración y mejora continua. Y la pregunta que te dejo es: ¿Estás listo para abrazar esta revolución en tu desarrollo de software? ¿Estás listo para liberar tu creatividad y hacer que tus ideas lleguen más rápido que nunca a tus usuarios? La respuesta está en tus manos, y el Despliegue Continuo es la varita mágica que puedes usar para hacerlo posible. ¡El futuro del desarrollo de software está en constante evolución, y tú eres parte de él!

**Referencias y Agradecimientos**

**Info que encontre**

El despliegue continuo es una estrategia de desarrollo de software en la que los cambios de código de una aplicación se publican automáticamente en el entorno de producción. Esta automatización se basa en una serie de pruebas predefinidas. Una vez que las nuevas actualizaciones pasan esas pruebas, el sistema envía las actualizaciones directamente a los usuarios del software.

El despliegue continuo ofrece varias ventajas para las empresas que quieren escalar su portfolio de aplicaciones y TI. En primer lugar, agiliza el tiempo de comercialización al eliminar el desfase entre la programación y el valor al cliente, que suele ser de días, semanas o incluso meses.

Para lograrlo, se deben automatizar las pruebas de regresión, de modo que se eliminen los altos costes de las pruebas de regresión manual. Los sistemas que las organizaciones han puesto en marcha para gestionar grandes paquetes de cambios de producción, incluidas las reuniones de planificación y aprobación de publicaciones, también se pueden eliminar en la mayoría de los cambios.

El despliegue continuo amplía un poco más la automatización y elimina la necesidad de intervención manual. Las pruebas y los desarrolladores se consideran lo suficientemente fiables como para que no sea necesario aprobar la publicación en producción. Si se pasan las pruebas, se considera que el código nuevo está aprobado y se inicia el despliegue en producción.

El despliegue continuo es el resultado natural de una entrega continua bien realizada. Al final, la aprobación manual aporta un valor mínimo o nulo, y solo ralentiza el proceso. Llegados a ese punto, se suprime, y la entrega continua pasa a ser un despliegue continuo.

Para que funcione la automatización de los procesos de despliegue, todos los desarrolladores que trabajan en un proyecto necesitan una forma eficaz de comunicar los cambios que se producen. Y eso es lo que posibilita la integración continua.

Normalmente, cuando se trabaja en el mismo proyecto de desarrollo de software, los desarrolladores trabajan en copias individuales de una rama maestra del código. Sin embargo, se pueden producir problemas y errores de funcionalidad una vez que los desarrolladores fusionan los cambios en la base de código principal, especialmente cuando los desarrolladores trabajan de forma independiente entre sí. Cuanto más tiempo trabajen de forma independiente, mayor será el riesgo.

Con la CI, todo el mundo fusiona sus cambios de código en un repositorio al menos una vez al día. A medida que se producen actualizaciones, se ejecutan pruebas de compilación automatizadas para asegurarse de que los cambios siguen siendo compatibles con la rama maestra. De este modo, se evitan errores y se detectan los problemas de integración lo más rápido posible.

Herramientas de despliegue continuo

Para desarrollar y desplegar continuamente mejoras de software de alta calidad, los desarrolladores deben utilizar las herramientas adecuadas para [crear prácticas eficaces de DevOps](https://www.ibm.com/garage/method/practices/deliver/tool_continuous_delivery). Hacerlo no solo garantiza una comunicación eficaz entre los departamentos de desarrollo y operaciones, sino que también minimiza o elimina los errores en el conducto de entrega de software.

Estas son algunas de las herramientas más cruciales utilizadas en un flujo de trabajo de despliegue continuo:

**Control de versiones:** el control de versiones facilita la integración continua mediante el seguimiento de las revisiones de los activos de un proyecto determinado. También conocido como control de «origen» o «revisión», el control de versiones ayuda a mejorar la visibilidad de las actualizaciones y los cambios de un proyecto, además de facilitar la colaboración de los equipos independientemente de dónde y cuándo trabajen.

**Revisión de código:** la «revisión de código» es el proceso de utilizar herramientas para probar el código fuente actual, así de simple. Las revisiones de código permiten mejorar la integridad del software al encontrar errores en la codificación y ayudan a los desarrolladores a resolver estos problemas antes de desplegar las actualizaciones.

**Integración continua (CI):** la CI es un componente indispensable del despliegue continuo y desempeña un papel muy importante en la minimización de los obstáculos para el desarrollo cuando varios desarrolladores trabajan en el mismo proyecto. Existe una gran variedad de herramientas de CI propietarias y de código abierto, y cada una de ellas atiende las complejidades exclusivas de los despliegues de software empresarial.

**Gestión de configuración:** la gestión de la configuración es la estrategia y la disciplina de asegurarse de que todo el software y el hardware mantienen un estado coherente. Esto incluye la configuración y la automatización adecuadas de todos los [servidores](https://www.ibm.com/es-es/topics/cloud-server), el [almacenamiento](https://www.ibm.com/es-es/topics/cloud-storage), la [red](https://www.ibm.com/es-es/topics/networking) y el software.

**Automatización de releases:** la automatización de los releases de aplicaciones (o la orquestación de releases de aplicaciones) es muy importante al automatizar todas las actividades necesarias para impulsar el despliegue continuo. Las herramientas de orquestación conectan los procesos entre sí para garantizar que los desarrolladores sigan todos los pasos necesarios antes de enviar nuevos cambios a producción. Estas herramientas trabajan estrechamente con los procesos de gestión de configuración para garantizar que todos los entornos del proyecto se suministren correctamente y puedan ofrecer su máximo nivel de rendimiento.

**Supervisión de infraestructuras:** al utilizar un modelo de despliegue continuo, es importante poder visualizar los datos que residen en los entornos de prueba. Las herramientas de supervisión de la infraestructura le ayudan a analizar el rendimiento de las aplicaciones para ver si los cambios realizados tienen un impacto positivo o negativo.

<https://www.ibm.com/es-es/topics/continuous-deployment>

<https://www.youtube.com/watch?v=zMdEjQW-NgI&list=WL&index=2&t=168s>

<https://www.youtube.com/watch?v=Odmv3is3Ibc&list=WL&index=3>