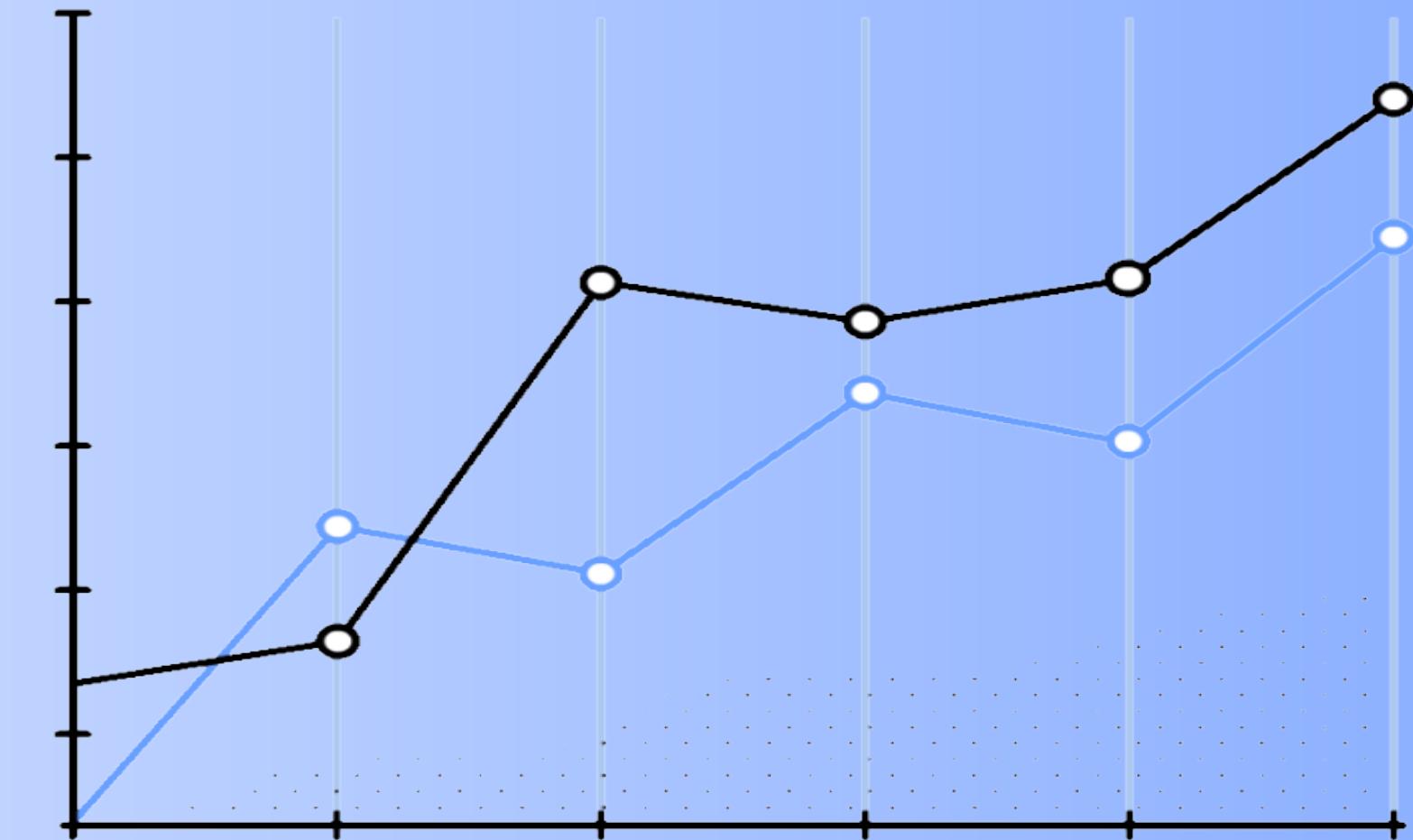


ACTIVIDAD 1

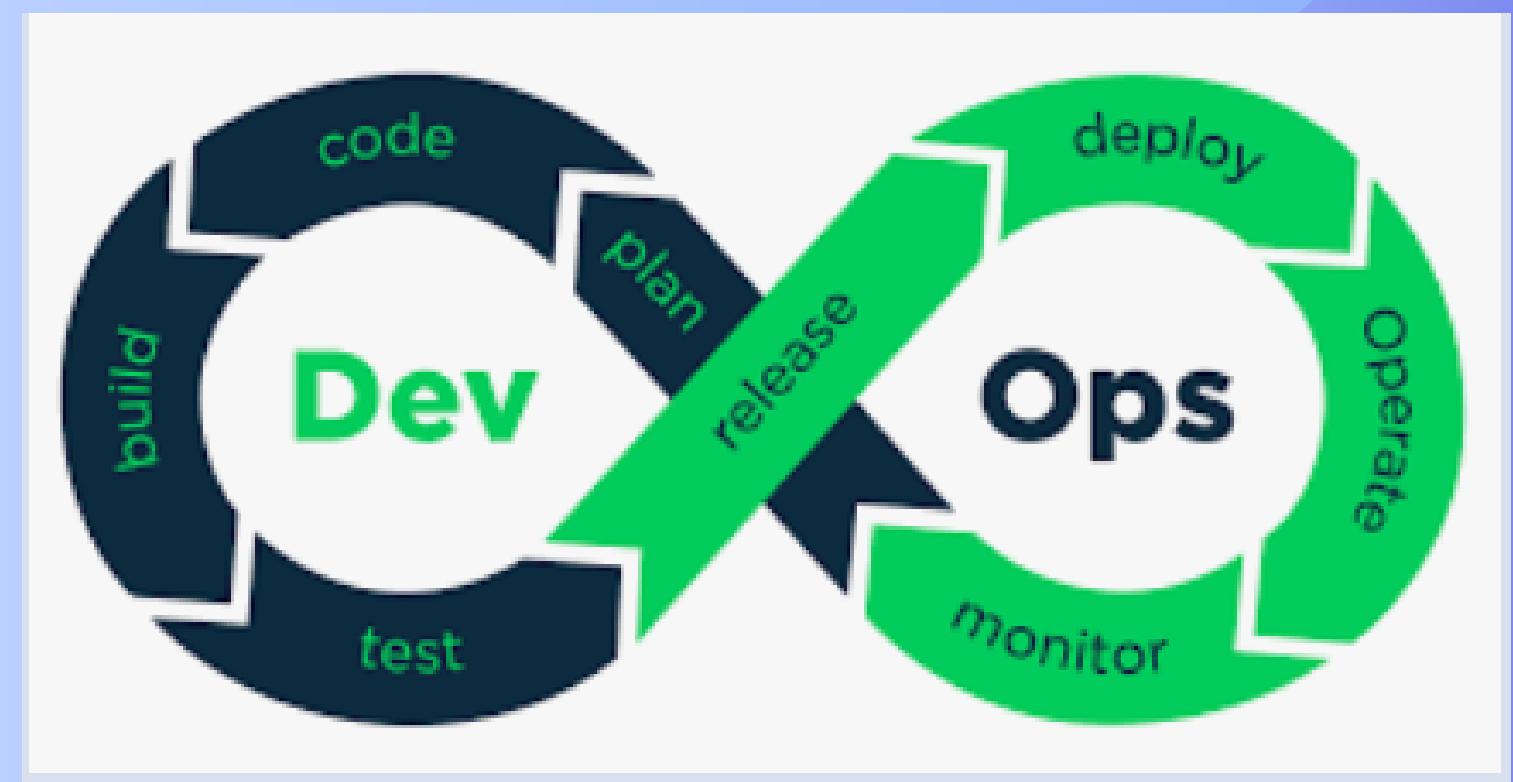
Bianca Merchan Torres
Jesus Diego Santa Cruz Basilio
Daren Herrera Romo



INTRODUCCION A DEVOPS

¿QUÉ ES DEVOPS?

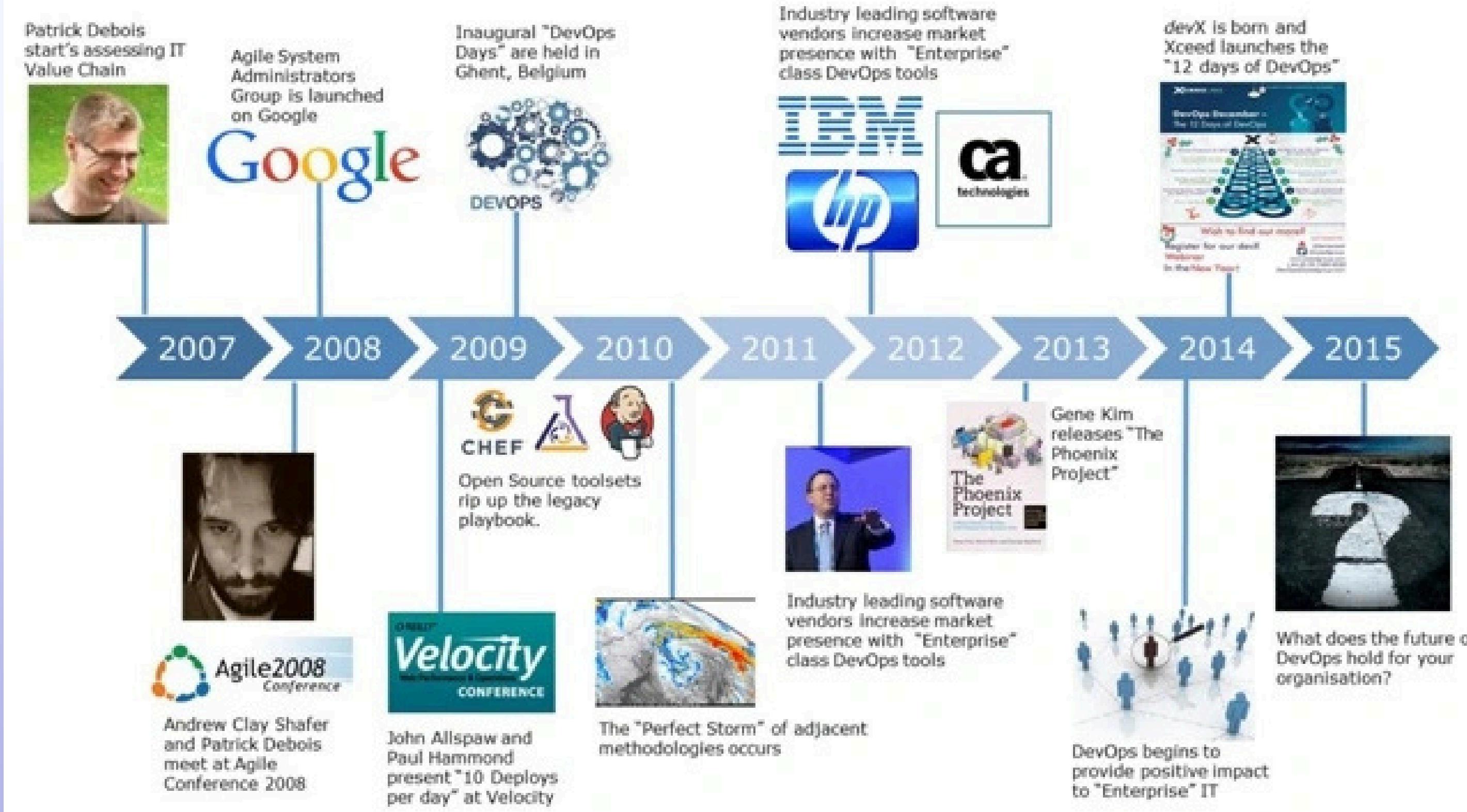
DevOps es una metodología que busca unir los equipos de desarrollo, aseguramiento de calidad (QA) y operaciones en un mismo proceso colaborativo para reducir el tiempo de entrega, mejorar la calidad y responder de manera más ágil a los cambios y necesidades del negocio.



HISTORIA Y ANTECEDENTES DE DEVOPS

DevOps apareció como una respuesta a los modelos recurrentes de desarrollo de software, donde los equipos de desarrollo y operaciones trabajaban de manera individual, lo que generaba ciclos largos de desarrollo y problemas de comunicación. Con DevOps se dio mas importancia a la colaboración, acortando el tiempo entre la creación de una idea y su implementación.

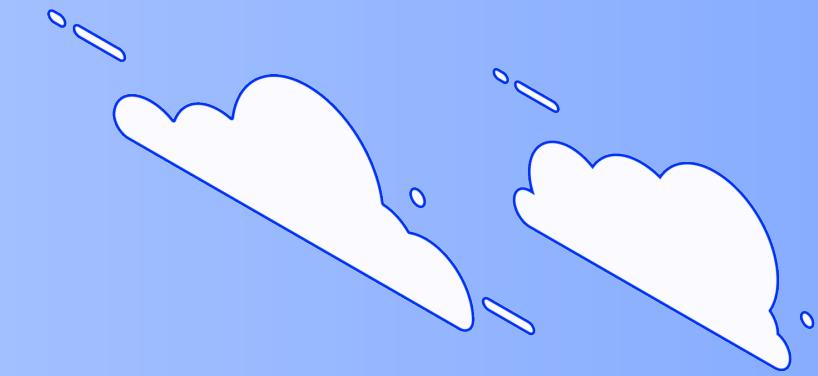
DevOps Timeline



DIFERENCIAS ENTRE LOS EQUIPOS DE DESARROLLO Y OPERACIONES EN EL PASADO

En los modelos tradicionales, los equipos de desarrollo, aseguramiento de calidad (QA) y operaciones trabajaban de forma separada. El equipo de desarrollo diseñaba y creaba el software, el equipo de QA realizaba pruebas exhaustivas, y el equipo de operaciones se encargaba del despliegue y mantenimiento. Este enfoque generaba una división entre roles y generaba problemas de comunicación, lo que alargaba los ciclos de desarrollo y complicaba la generación de respuesta ante cambios y errores.





PRINCIPIOS FUNDAMENTALES DE DEVOPS

- **Enfoque en el cliente:** La mejora en la entrega de software, con el cliente como el centro del proceso.
- **Equipos autónomos y multifuncionales:** Equipos que combinan disciplinas (desarrollo, QA, operaciones) para permitir un flujo de trabajo más ágil y flexible.
- **Mejora continua:** Un enfoque en la evaluación constante de procesos y resultados, tanto en el ámbito técnico como cultural, para realizar ajustes y optimizaciones.
- **Automatización:** Implementación de herramientas y procesos que mecanizan tareas frecuentes (como integración y entrega continua), disminuyendo la intervención manual y aumentando la eficiencia.

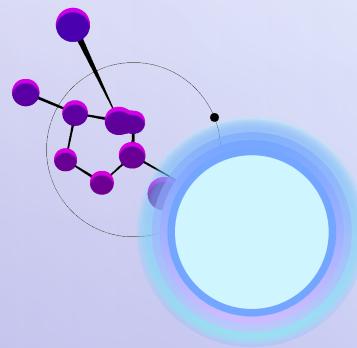


QUE NO ES DEV OPS

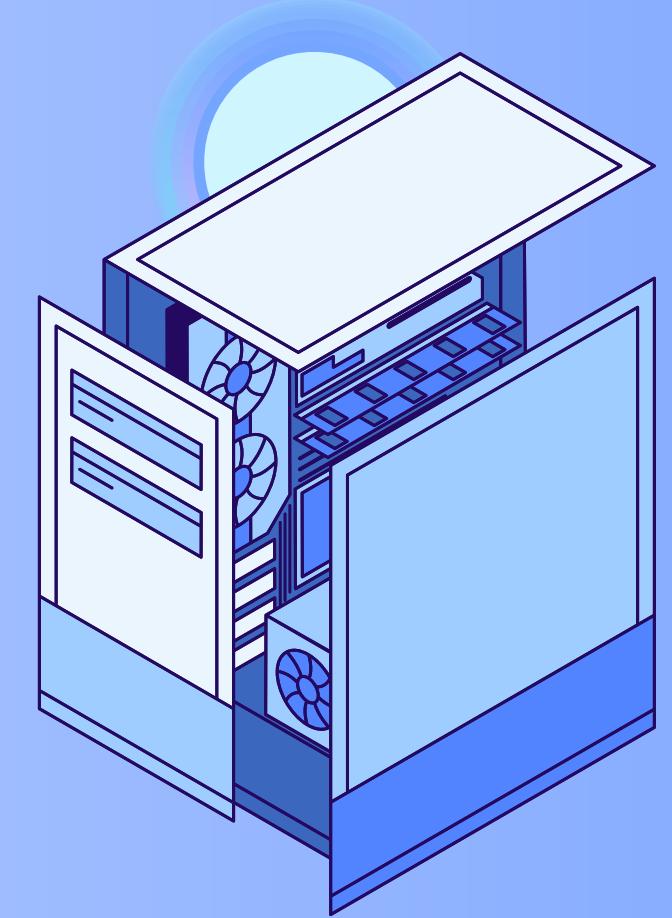
DevOps no es solo un conjunto de herramientas, individuos o procesos aislados, sino de una transformación cultural que involucra a toda la organización. DevOps no es una solución mágica ni un proceso automático; implica un cambio profundo en la forma de trabajar, comunicar y colaborar entre los equipos.

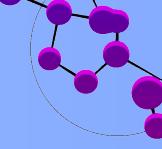


¿POR QUÉ SURGIÓ LA NECESIDAD DE DEVOPS?

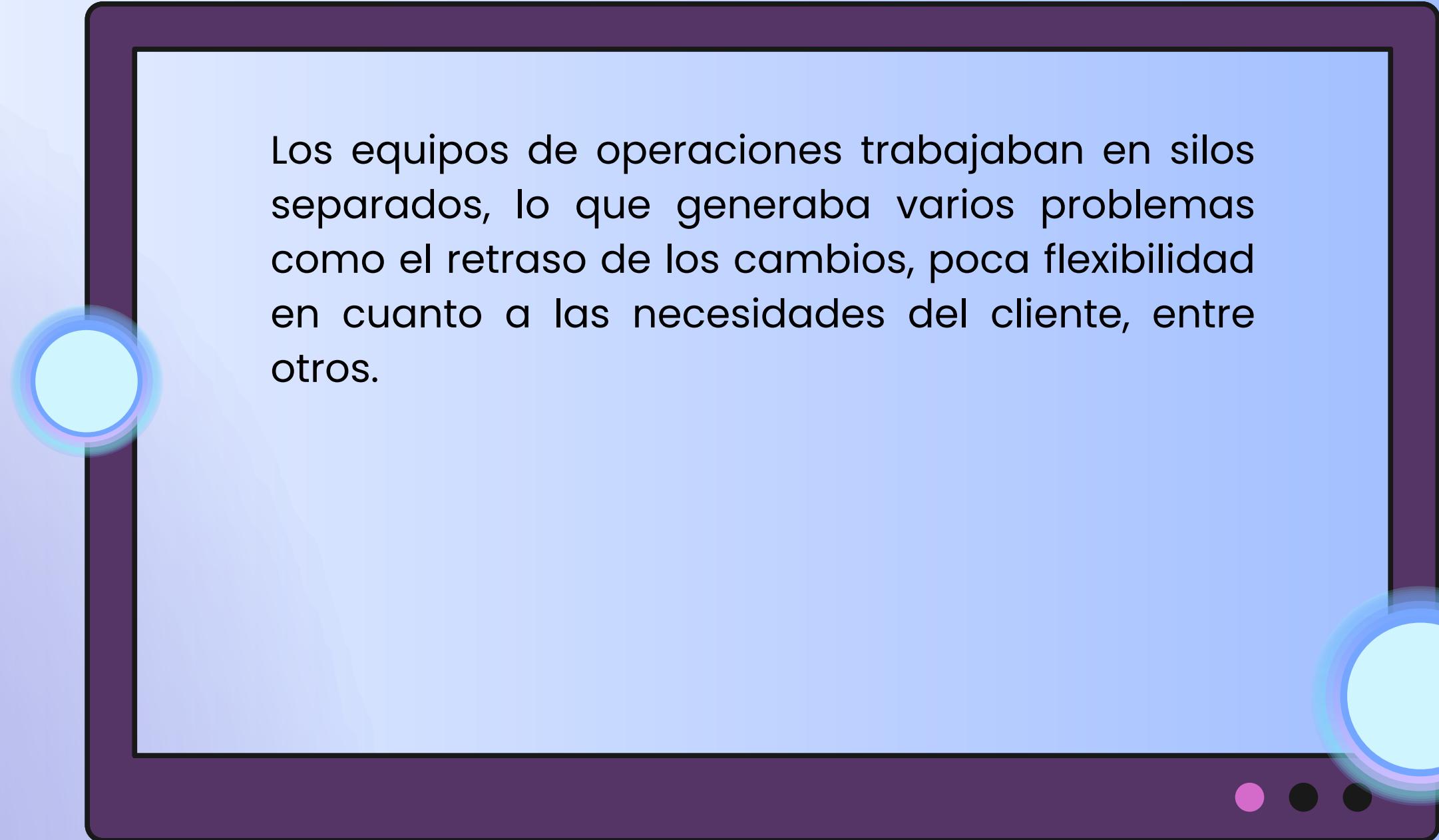


Surgió en base a la necesidad de mejorar los modelos tradicionales que une a las distintas áreas del desarrollo de software, permitiendo acortar ciclos de entrega y mejorar la calidad del producto.





¿CÓMO LA FALTA DE COMUNICACIÓN Y COORDINACIÓN LLEVÓ A LA CREACIÓN DE DEVOPS?



Los equipos de operaciones trabajaban en silos separados, lo que generaba varios problemas como el retraso de los cambios, poca flexibilidad en cuanto a las necesidades del cliente, entre otros.





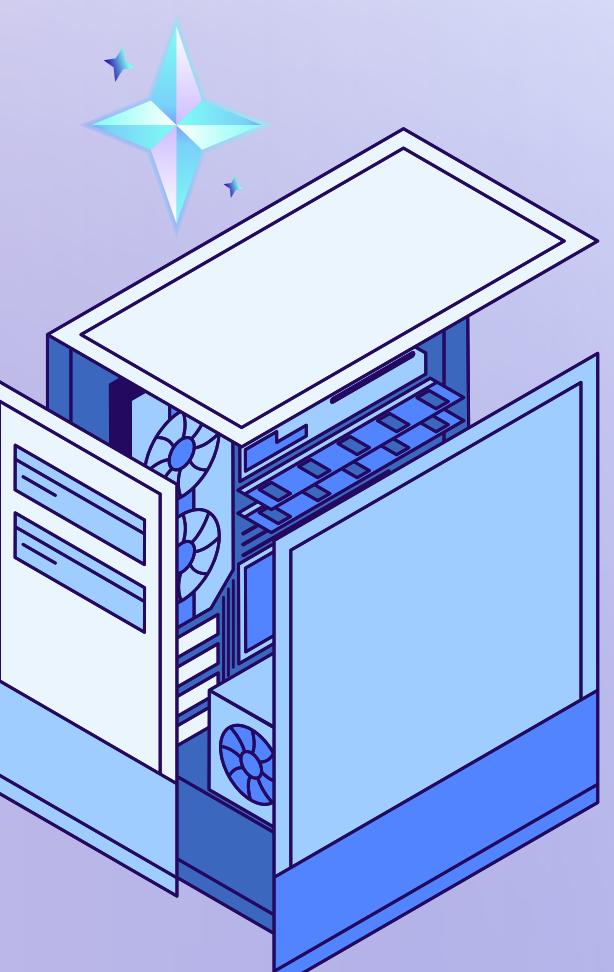
¿CÓMO EL PRINCIPIO DE MEJORA CONTINUA IMPACTA TANTO EN LOS ASPECTOS TÉCNICOS COMO EN LOS CULTURALES DE UNA ORGANIZACIÓN?

A nivel técnico, implica la optimización constante de pipelines de integración y entrega continua, la automatización de tareas y la resolución de problemas rápidamente. A nivel cultural, significa fomentar una mentalidad de aprendizaje y adaptación, donde cada miembro del equipo se siente responsable de la calidad y mejora del proceso general.

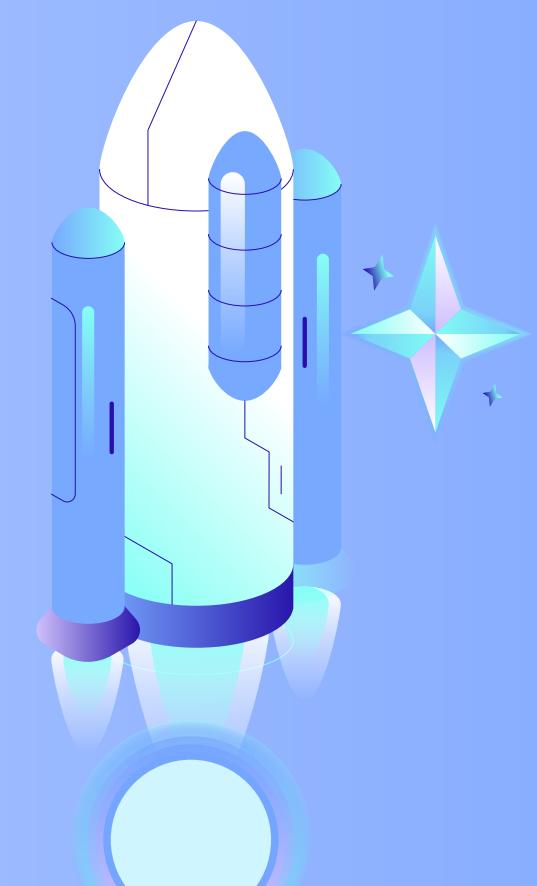
¿QUÉ SIGNIFICA QUE DEVOPS NO SE TRATA SOLO DE HERRAMIENTAS, INDIVIDUOS O PROCESOS?

Esto significa que DevOps no debe verse únicamente como el desarrollo y uso de nuevas tecnologías, ni como la implementación de ciertos procesos. DevOps es una transformación organizacional que busca cambiar la forma en que los equipos interactúan y colaboran. Implica un cambio cultural en cómo se gestionan las responsabilidades, se comunican los equipos y se toman decisiones de forma conjunta. No se trata solo de usar las herramientas adecuadas, sino de trabajar juntos y con un enfoque en la mejora continua.

¿CÓMO CONTRIBUYEN LOS EQUIPOS AUTÓNOMOS Y MULTIFUNCIONALES A UNA IMPLEMENTACIÓN EXITOSA DE DEVOPS?



En un entorno DevOps, los equipos multifuncionales reúnen expertos en desarrollo, pruebas, seguridad y operaciones dentro de un mismo grupo, lo que les permite trabajar de manera conjunta desde la planificación hasta la implementación y mantenimiento del software.



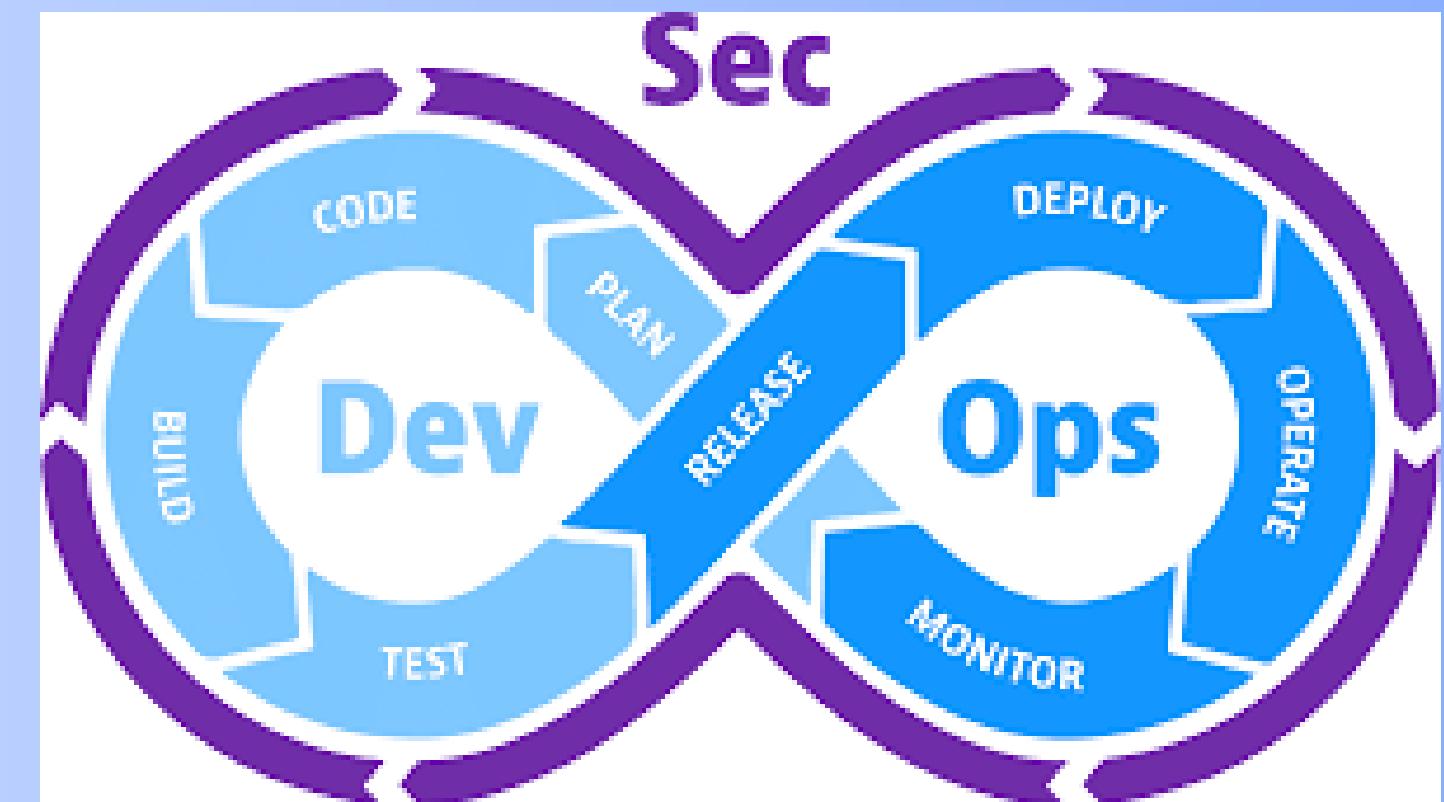
Esta autonomía facilita la toma de decisiones y agiliza el proceso de entrega de software, ya que los equipos pueden implementar cambios sin depender de aprobaciones externas o procesos secuenciales extensos. Además, al compartir la responsabilidad del código en producción, todos los miembros del equipo están más comprometidos con la calidad y estabilidad del sistema, lo que reduce la posibilidad de errores y fallos en producción. La colaboración constante también permite una retroalimentación más rápida, lo que facilita la mejora continua y la optimización del flujo de trabajo.



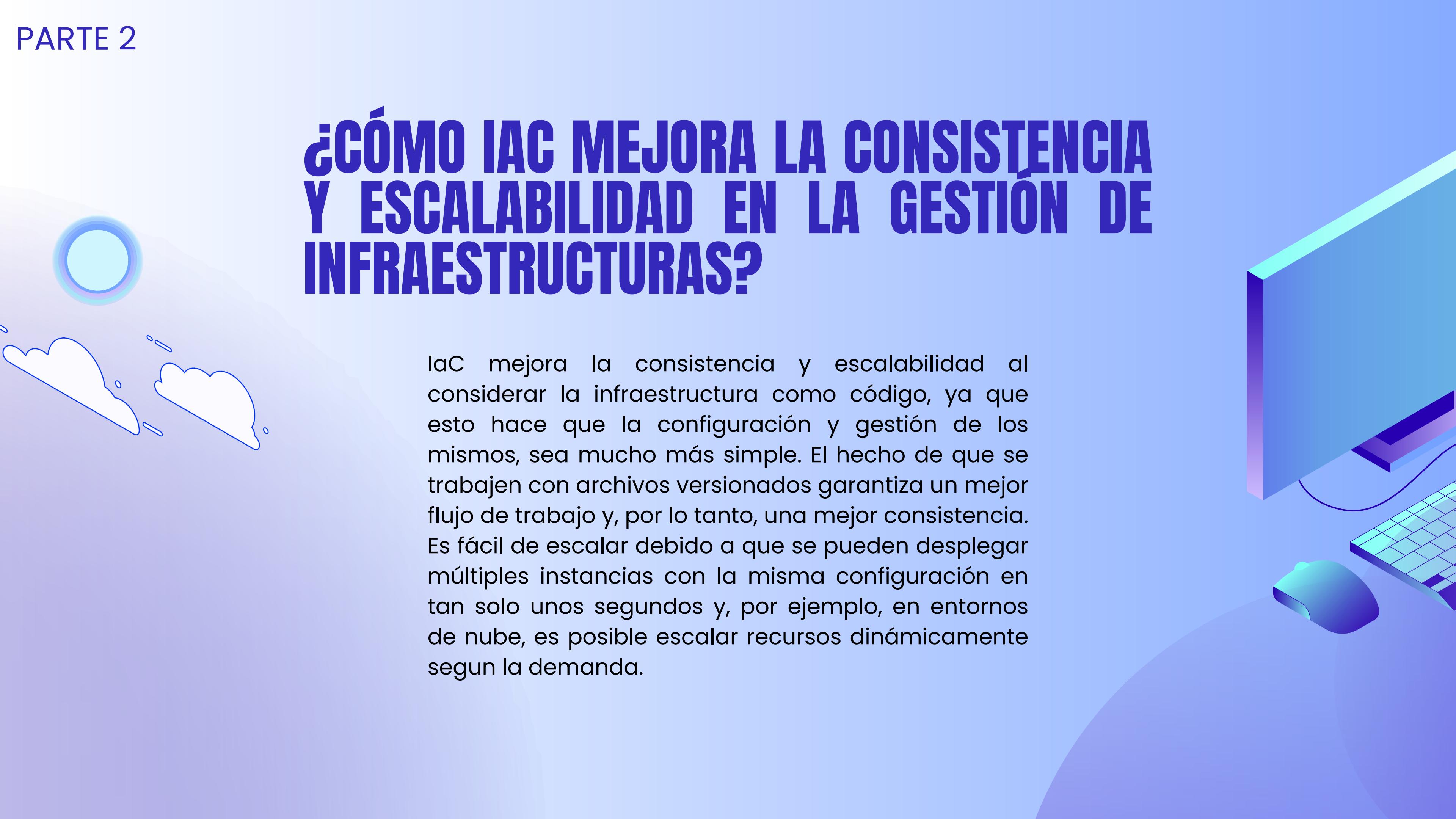
INFRAESTRUCTURA COMO CÓDIGO Y DEVSECOPS

¿QUÉ SIGNIFICA "DESPLAZAR A LA IZQUIERDA" EN EL CONTEXTO DE DEVSECOPS Y POR QUÉ ES IMPORTANTE?

Antes, cuando se trabajaba con la metodología tradicional, la seguridad era el último paso en el proceso de desarrollo. Por ello, DevSecOps incorpora un nuevo concepto: “Desplazar a la izquierda”. Esto consiste en integrar la seguridad en la fase temprana de desarrollo. Con esto, se pueden identificar y corregir problemas tempranamente para reducir costos y minimizar riesgos.



¿CÓMO IAC MEJORA LA CONSISTENCIA Y ESCALABILIDAD EN LA GESTIÓN DE INFRAESTRUCTURAS?



IaC mejora la consistencia y escalabilidad al considerar la infraestructura como código, ya que esto hace que la configuración y gestión de los mismos, sea mucho más simple. El hecho de que se trabajen con archivos versionados garantiza un mejor flujo de trabajo y, por lo tanto, una mejor consistencia. Es fácil de escalar debido a que se pueden desplegar múltiples instancias con la misma configuración en tan solo unos segundos y, por ejemplo, en entornos de nube, es posible escalar recursos dinámicamente según la demanda.



¿CUÁL ES LA DIFERENCIA ENTRE MONITOREO Y OBSERVABILIDAD? ¿POR QUÉ ES CRUCIAL LA OBSERVABILIDAD EN SISTEMAS COMPLEJOS?

El monitoreo es uno de los pilares de la observabilidad. El monitoreo es el proceso de reunir y analizar datos sobre las métricas de un sistema. La observabilidad va mucho más allá del monitoreo, esta permite entender problemas complejos. No solo detecta fallos, sino que también ayuda a saber el porqué de los mismos. Es importante en los sistemas complejos ya que estos, actualmente, se caracterizan muchas veces por ser impredecibles y dinámicos. Con la observabilidad podemos detectar y entender problemas más rápido, tener un mejor rendimiento y una mayor resiliencia en estos sistemas.



¿CÓMO PUEDE LA EXPERIENCIA DEL DESARROLLADOR IMPACTAR EL ÉXITO DE DEVOPS EN UNA ORGANIZACIÓN?

La experiencia hace a las personas más veloces y eficientes al realizar tareas. Los desarrolladores, en su caso, debido a ello, tienen flujos de trabajo mejor optimizados, podiendo desarrollar y desplegar código más rápido. Además, la familiarización con las herramientas con la que se suele trabajar en el enfoque DevOps suele ser crucial para un mejor desenvolvimiento en los equipos.



CÓMO INNERSOURCE PUEDE AYUDAR A REDUCIR SILOS DENTRO DE UNA ORGANIZACIÓN

InnerSource es la aplicación de principios del desarrollo open source dentro de una organización. En lugar de que los equipos trabajen de manera aislada en sus propios repositorios y procesos, InnerSource permite que cualquier empleado contribuya a los proyectos internos de la empresa. Con InnerSource, los proyectos internos son accesibles para otros equipos, lo que evita duplicar esfuerzos y fomenta la colaboración abierta. La documentación y las prácticas compartidas permiten que más personas comprendan y mejoren los sistemas existentes.



¿QUÉ ROL JUEGA LA INGENIERÍA DE PLATAFORMAS EN MEJORAR LA EFICIENCIA Y LA EXPERIENCIA DEL DESARROLLADOR?

La ingeniería de plataformas es una práctica creada a partir de conceptos de DevOps. Busca mejorar la seguridad, cumplimiento, costos y entre otros aspectos relacionados a cada equipo de desarrollo. Las plataformas que los desarrolladores usan, tienen la capacidad de eliminar todo el proceso de desarrollo con experiencias de desarrollador altamente optimizadas y operaciones simplificadas.

- Ayuda a los desarrolladores a ser independientes (kits de inicio, complementos IDE)
- Encapsula prácticas comunes en bloques reutilizables
- Proporcionar consejos y comentarios anticipados sobre problemas
- Simplificación de las operaciones mediante la administración de la infraestructura y herramientas relacionadas

