



DEVOPS - INGENIERÍA DE SOFTWARE I

Mora Axel, Ventura Gino

Definición de DevOps

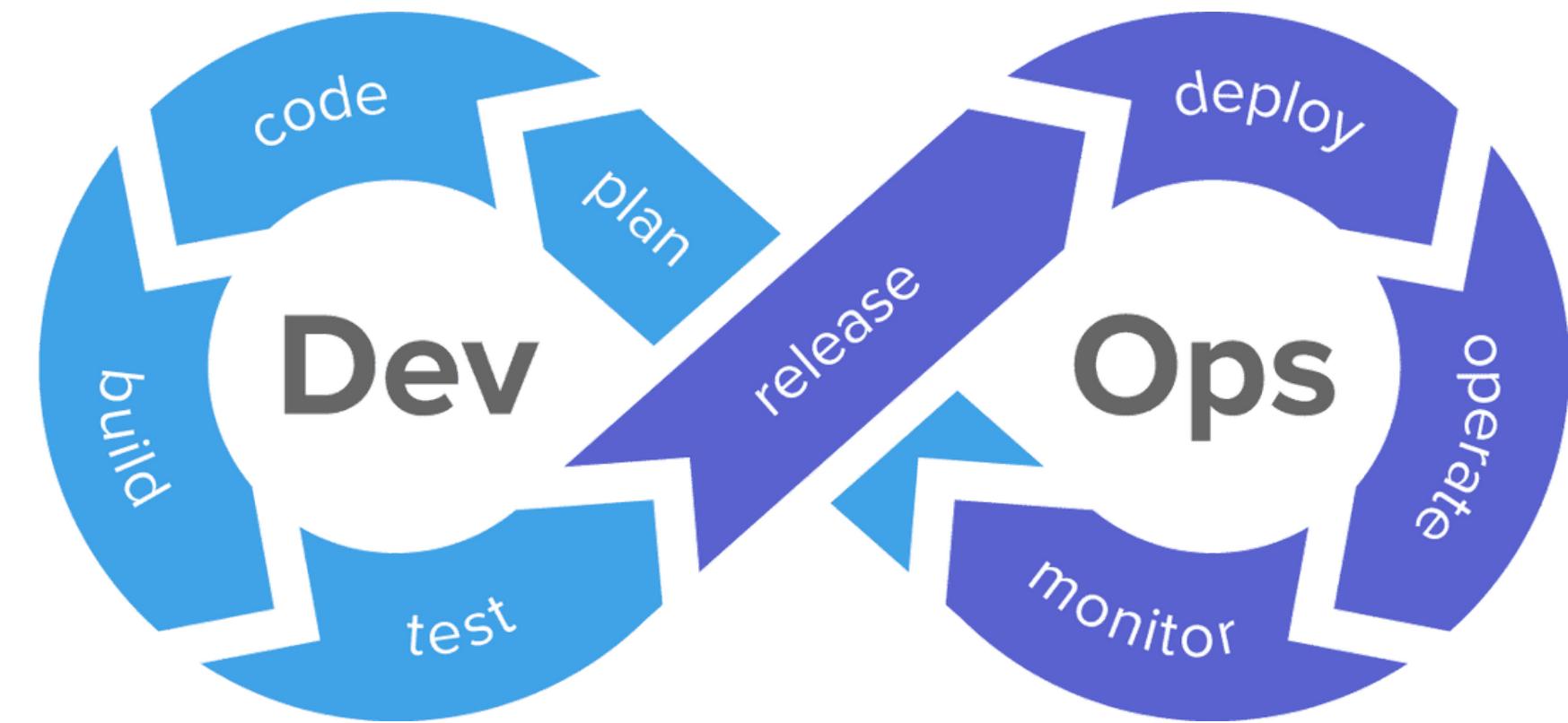
Es una combinación de los términos ingleses development (desarrollo) y operations (operaciones).

Designa la unión de personas, procesos y tecnología para ofrecer valor a los clientes de forma constante.



Significado para los equipos

Permite que los roles que antes estaban aislados (desarrollo, operaciones de TI, ingeniería de la calidad y seguridad) se coordinen y colaboren para ofrecer productos mejores y más confiables.



Historia de DevOps



Historia de DevOps

El origen de DevOps nos lleva al año 2007 en Bélgica cuando un consultor freelance llamado **Patrick Debois** experimenta los conflictos e inconvenientes de la rivalidad entre los equipos de operaciones y desarrollo en la organización para la cual trabajaba.

Un año después, en una conferencia tiene un encuentro con Andrew Shafer, donde coinciden en la problemática y deciden crear un grupo en Google llamado Agile System Administrators, para que otras personas puedan compartir sus opiniones al respecto.

Para 2009 John Allspaw y Paul Hammond dan a conocer "10 deploys per day" que proponía enfocar esfuerzos en la integración de los departamentos de operaciones y desarrollo. Esta propuesta llama la atención de Debois, quien decide llevarlo a cabo y adoptar el término **DevOps (Dev and Ops Cooperation)**.

En qué consiste DevOps



Al adoptar una cultura de DevOps, los equipos adquieren la capacidad de responder mejor a las necesidades de los clientes, aumentar la confianza en las aplicaciones que crean y alcanzar los objetivos empresariales en menos tiempo.

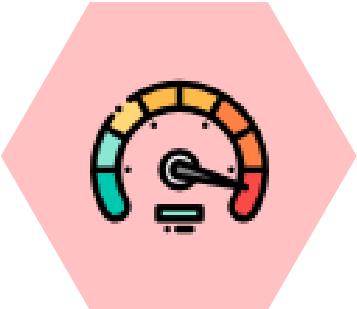
Bajo un modelo de DevOps, los equipos de desarrollo y operaciones ya no están “aislados”. A veces, los dos equipos se fusionan en uno solo.

En algunos casos, los equipos de control de calidad y de seguridad también se integran con el equipo de desarrollo y de operaciones e intervienen durante todo el ciclo de vida de la aplicación.



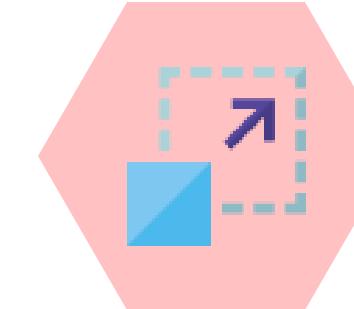
Ventajas de DevOps

Los equipos que adoptan la cultura DevOps mejoran el rendimiento y crean productos de más calidad en menos tiempo, lo que aumenta la satisfacción de los clientes. Esta mejora de la colaboración y la productividad es fundamental también para alcanzar objetivos de negocio como estos:



Velocidad

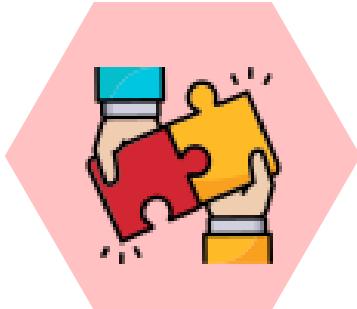
Permite una mejor y más rápida entrega de productos, como también, resolución de problemas en menos tiempo y con menor complejidad.



Escalabilidad

Permite la operación y administración de la infraestructura y los procesos de desarrollo a escala. La automatización ayuda a administrar sistemas complejos o cambiantes de forma eficaz con menos riesgo.

Colaboración Mejorada



Los desarrolladores y equipos de operaciones colaboran estrechamente, comparten muchas responsabilidades y combinan sus flujos de trabajo. Así se reducen las ineficacias y se ahorra tiempo.



Seguridad

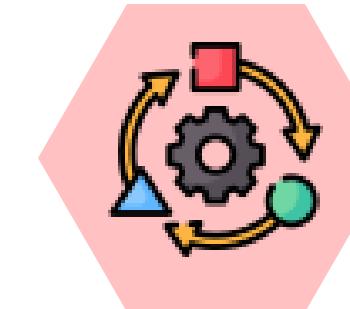
La automatización de los procedimientos de control contra amenazas y vulnerabilidades bajo metodologías DevOps ofrece una mayor seguridad de las aplicaciones desarrolladas.

Confiabilidad

Garantiza la calidad de las actualizaciones de la aplicación y los cambios en la infraestructura a fin de poder realizar entregas más rápidas de forma confiable mientras ofrece una experiencia positiva a los usuarios finales.

Utilizando prácticas como la integración continua y la entrega continua sirven para comprobar que cada cambio es funcional y seguro.

Las prácticas de monitoreo y registro ayudan a mantenerse informado del desempeño en tiempo real.



Adaptación

La entrega continua permite una mayor flexibilidad y adaptación al entorno que los sistemas de desarrollos tradicionales en los que el tiempo de despliegue es mayor.

Prácticas de DevOps



Prácticas de DevOps

01

Desarrollo Continuo

Abarca las fases de planificación y codificación del ciclo de DevOps. Puede incluir también mecanismos de control de versiones.

02

Pruebas Continuas

Incorpora pruebas de código automatizadas y programadas con antelación que se realizan a medida que el código de aplicación se está creando o actualizando. Gracias a estas pruebas, el código pasa antes a la fase de producción.

03

Integración Continua

Se combinan herramientas de gestión de configuración con otras herramientas de pruebas y desarrollo para saber qué cantidad del código que se está creando está listo para pasar a producción.

Prácticas de DevOps

04

Entrega Continua

Automatiza el lanzamiento de código nuevo o modificado a la fase de producción.

05

Supervisión Continua

Implica la supervisión continua del código en la fase de producción y la infraestructura que la sustenta. A través de un bucle de retroalimentación en el que se notifican errores o problemas, este podría volver a la fase de desarrollo.

06

Aprendizaje Continuo

Se acepta el fracaso y se responde rápido a los errores, y se incorpora lo que se aprende a los procesos, de modo que la mejora es continua, Aumenta la satisfacción del cliente y la capacidad de adaptación al mercado de forma constante.

¿Quién utiliza DevOps?



¿Quién utiliza DevOps?



Desde las empresas hasta las start-ups en fase inicial, DevOps se está adoptando en las empresas de IT de todo el mundo. Estas son algunas de las organizaciones que han implementado DevOps:

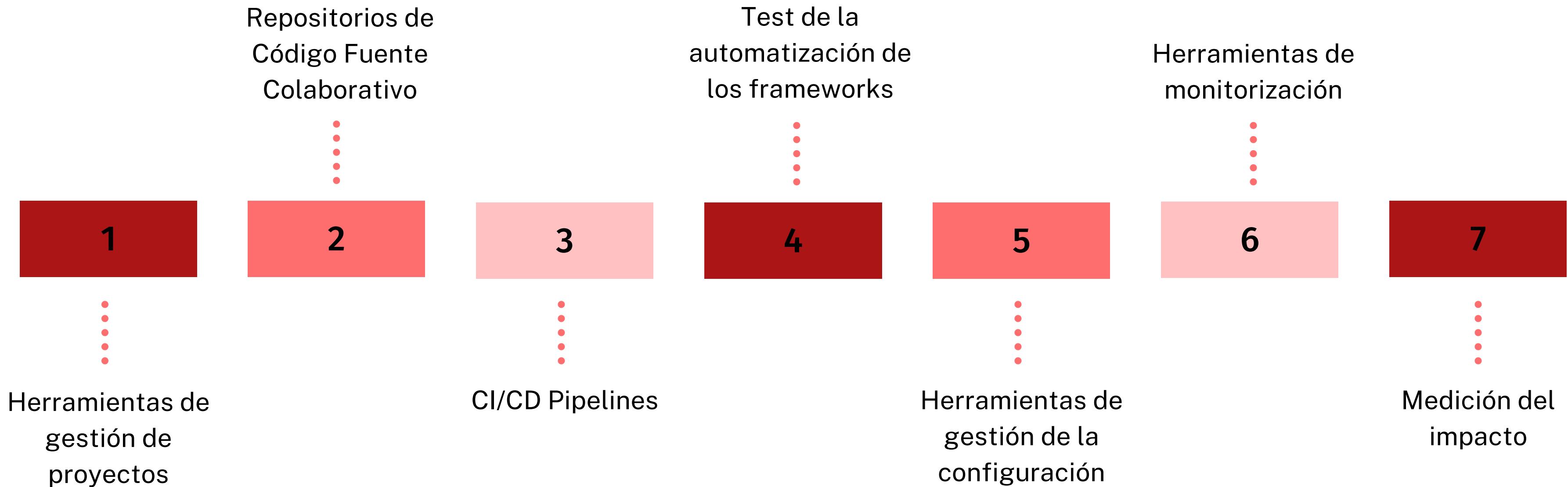
- Empresas como Amazon, Facebook, Netflix, etc.
- Empresas de medios de comunicación como Sony Pictures.
- Organizaciones gubernamentales como la Oficina de Comercio de EE.UU.
- Pequeñas y medianas empresas.

Herramientas DevOps

Una parte fundamental de la aplicación de este enfoque son las herramientas DevOps, utilizadas para agilizar cada aplicación del proceso, desde el planteamiento inicial hasta la entrega del software.



Herramientas DevOps



Herramientas DevOps

Herramientas de gestión de proyectos

Se encargan de la administración y seguimiento de las tareas del proyecto. Por ejemplo: GitHub Issues y Jira.

Repositorios de código fuente colaborativo

Gracias a estos entornos, los desarrolladores pueden trabajar en la misma base de código. Los repositorios de código abierto más destacados son GitHub y GitLab.

CI/CD pipelines

Estas son herramientas para la integración continua, es decir, la automatización de todo el ciclo de vida de un desarrollo. Por ejemplo: Jenkins y ArgoCD.

Test de automatización de los frameworks

Son herramientas de prueba continua que graban y reproducen la funcionalidad de la aplicación. Las mas usadas son Selenium, Appium o Serenity.

Herramientas de gestión de la configuración

Son encargadas de configurar y gestionar la infraestructura como código facilitando la labor de los ingenieros en la ejecución de un script. Las de código abierto más usadas son Ansible (Red Hat), Chef, Puppet y Terraform.

Herramientas de monitorización

Su objetivo es recopilar datos en tiempo real y detectar posibles problemas que puedan afectar al rendimiento de las aplicaciones y la experiencia del usuario. Algunas de estas herramientas de código abierto son Datadog, Nagios, Prometheus y Splunk.

Medición del impacto

Estas se implementan después del lanzamiento del producto y su objetivo es registrar el comportamiento y satisfacción de los usuarios, mediante mapas de calor, encuestas o anotación de incidencias.

Conclusión



DevOps trata de eliminar los grupos aislados y hacer que el ciclo de vida del desarrollo sea más rápido, altamente automatizado y colaborativo, ya sea entre departamentos o dentro de un mismo equipo.