

DEVOPS

**SOFTWARE AVANZADO – VACACIONES
DICIEMBRE 2020**

DEVOPS SEGÚN WIKIPEDIA

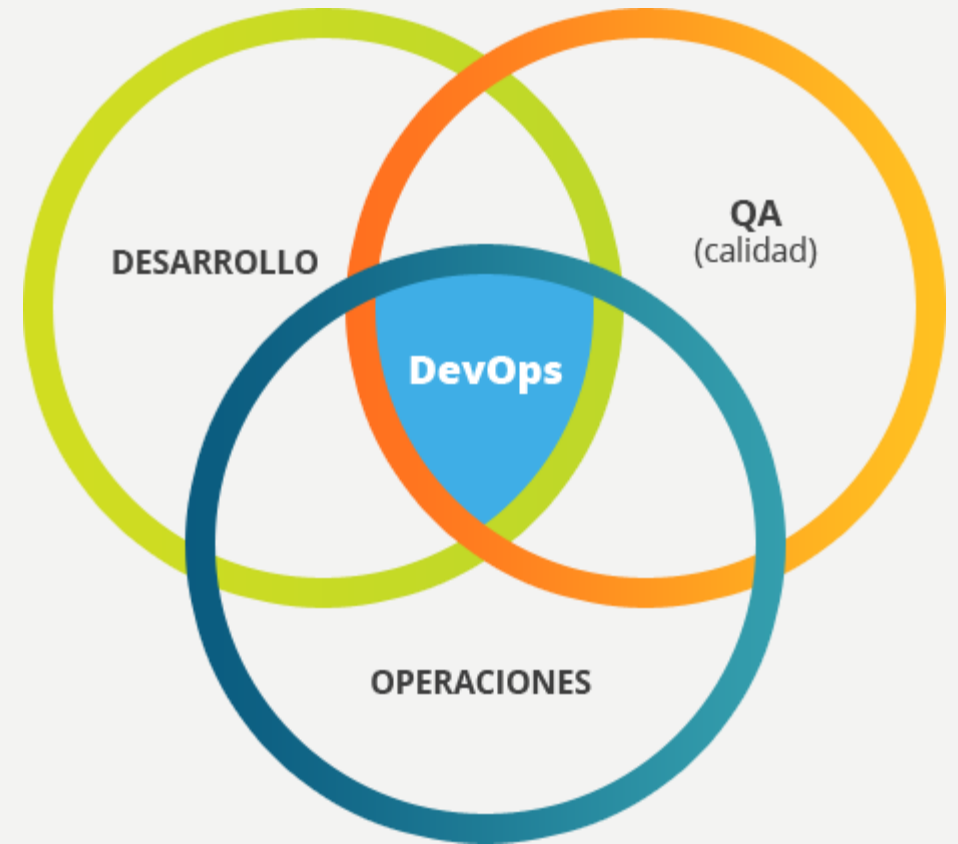
DevOps es un acrónimo inglés de development (desarrollo) y operations (operaciones), que se refiere a una metodología de desarrollo de software que se centra en la comunicación, colaboración e integración entre desarrolladores de software y los profesionales de sistemas en las tecnologías de la información (IT).

DevOps es una respuesta a la interdependencia del desarrollo de software y las operaciones IT. Su objetivo es ayudar a una organización a producir productos y servicios software más rápidamente, de mejor calidad y a un coste menor.

Las empresas con entregas (releases) muy frecuentes podrían requerir conocimientos de DevOps. Flickr desarrolló un sistema DevOps para cumplir un requisito de negocio de diez despliegues diarios. A este tipo de sistemas se les conoce como despliegue continuo (continuous deployment) o entrega continua (continuous delivery), y suelen estar asociados a metodologías lean startup. Grupos de trabajo, asociaciones profesionales y blogs usan el término desde 2009.

IDEAS CLAVES SOBRE DEVOPS

- DevOps es una metodología para creación de software.
- DevOps se basa en la integración entre desarrolladores software y administradores de sistemas.
- DevOps permite fabricar software más rápidamente, con mayor calidad, menor coste y una altísima frecuencia de releases.



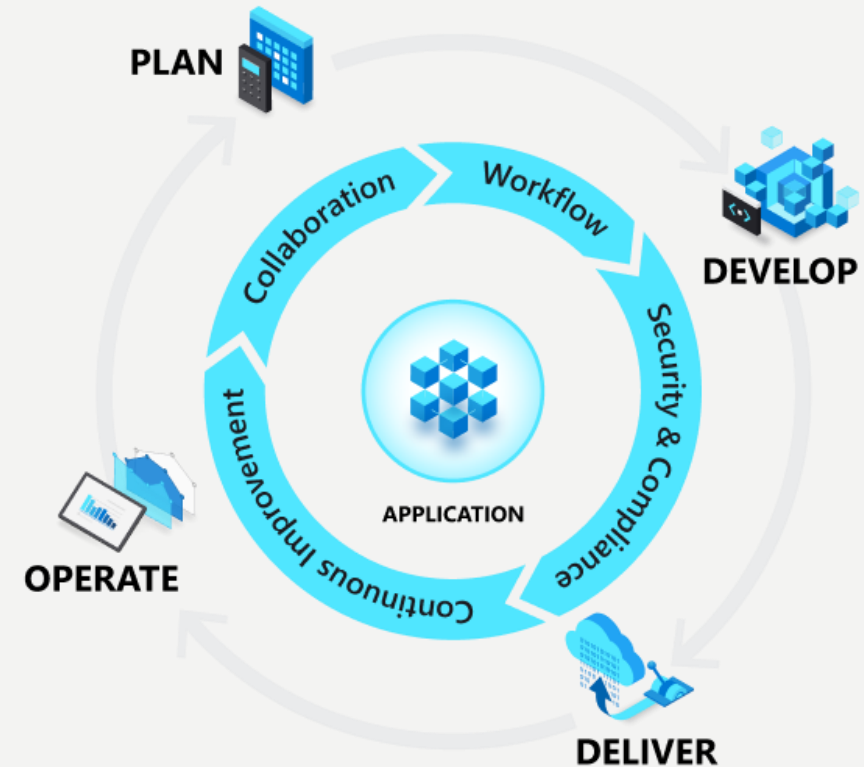
¿ES DEVOPS UNA CULTURA?

DevOps no es en sí una cultura, pero sí requiere de un fuerte cambio cultural y organizativo para su implementación. Un cambio cultural hacia la colaboración, la comunicación, y en último término la completa integración entre las antiguas áreas, en lo habitual rabiosamente estancas, de desarrollo y sistemas.

Este cambio cultural es tan complicado de conseguir en algunas organizaciones, que son muchos los que lo identifican directamente con DevOps, pero recordemos: DevOps es una metodología de desarrollo software, y un cambio de cultura no es en sí mismo una forma de desarrollar software.

DEVOPS Y EL CICLO DE VIDA DE LAS APLICACIONES

DevOps influye en el ciclo de vida de las aplicaciones a lo largo de las fases de planeamiento, desarrollo, entrega y uso. Cada fase depende de las demás y las fases no son específicas de un rol. En una auténtica cultura de DevOps, todos los roles están implicados de algún modo en todas las fases.



FASE DE PLANEACIÓN

En la fase de planeamiento, los equipos de DevOps conciben, definen y describen las características y la funcionalidad de las aplicaciones y los sistemas que van a crear. Realizan un seguimiento del progreso tanto de forma general como de forma pormenorizada, desde tareas de un único producto hasta tareas que abarcan carteras de numerosos productos.

La creación de registros de trabajo pendiente, el seguimiento de los errores, la administración del desarrollo de software ágil con Scrum, el uso de paneles Kanban y la visualización del progreso son algunas de las formas en las que los equipos de DevOps planean con agilidad y visibilidad.

FASE DE DESARROLLO

La fase de desarrollo incluye todos los aspectos de la programación como lo son escritura, pruebas, revisión e integración del código por parte de los miembros del equipo y la compilación de ese código en artefactos de compilación que se pueden implementar en varios entornos.

Los equipos de DevOps buscan innovar con rapidez sin sacrificar la calidad, la estabilidad ni la productividad. Para ello, utilizan herramientas muy productivas, automatizan los pasos cotidianos y manuales, e iteran el código en pequeños incrementos mediante pruebas automáticas e integración continua.

FASE DE ENTREGA

La entrega es el proceso de implementar aplicaciones en entornos de producción de un modo constante y confiable. La fase de entrega incluye también la implementación y la configuración de la infraestructura básica totalmente gobernada que constituye esos entornos.

En la fase de entrega, los equipos definen un proceso de administración de versiones con fases de aprobación manual claras. También establecen puertas automáticas que mueven las aplicaciones de una fase a otra hasta que están disponibles para los clientes.

La automatización de estos procesos hace que estén controlados y sean escalables y repetibles. De este modo, los equipos que practican DevOps pueden realizar entregas con facilidad, confianza y tranquilidad.

FASE DE FUNCIONAMIENTO

La fase de uso implica mantener y supervisar las aplicaciones, así como solucionar los posibles problemas, en los entornos de producción. Al adoptar prácticas de DevOps, los equipos trabajan para asegurar la confiabilidad, la alta disponibilidad y el objetivo de ningún tiempo de inactividad del sistema, al tiempo que refuerzan la seguridad y la gobernanza. Los equipos de DevOps buscan identificar los problemas antes de que afecten a la experiencia del cliente y mitigarlos rápidamente a medida que surgen. El mantenimiento de esta vigilancia requiere una telemetría muy completa, alertas que permitan tomar medidas y visibilidad total de las aplicaciones y del sistema subyacente.

PRÁCTICAS DE DEVOPS

Más allá del establecimiento de una cultura o metodología de DevOps, los equipos ponen en práctica el método DevOps implementando determinadas prácticas a lo largo del ciclo de vida de las aplicaciones. Algunas de estas prácticas ayudan a agilizar, automatizar y mejorar una fase específica. Otras abarcan varias fases y ayudan a los equipos a crear procesos homogéneos que favorezcan la productividad.

INTEGRACIÓN Y ENTREGA CONTINUAS (CI/CD)

- La integración continua es una práctica de desarrollo de software en la que los desarrolladores fusionan mediante combinación los cambios de código en la rama de código principal con frecuencia. En la integración continua se utilizan pruebas automáticas, que se ejecutan cada vez que se hace "commit" de código nuevo. De este modo, el código de la rama principal siempre es estable.
- La entrega continua es la implementación automática y frecuente de nuevas versiones de una aplicación en un entorno de producción. Al automatizar los pasos necesarios para la implementación, los equipos reducen los problemas que pueden surgir en ese proceso y permiten actualizaciones más frecuentes.

Cuando se establecen estas dos prácticas, el proceso resultante es CI/CD, que incluye la automatización completa de todos los pasos desde que se hace "commit" del código hasta que se implementa en el entorno de producción. La implementación de CI/CD permite a los equipos centrarse en la creación de código y elimina la carga y la posibilidad de errores humanos en los pasos cotidianos que se realizan manualmente.

CI/CD también agiliza el proceso de implementación de nuevo código y reduce los riesgos que conlleva. Por tanto, las implementaciones son más frecuentes y se realizan en incrementos más pequeños, lo que ayuda a los equipos a ser más ágiles y más productivos, y a confiar más en el código que se ejecuta.

CONTROL DE VERSIONES

Control de versiones es la práctica de administrar el código por versiones, haciendo un seguimiento de las revisiones y del historial de cambios para facilitar la revisión y la recuperación del código. Esta práctica suele implementarse con sistemas de control de versiones, como Git, que permite que varios desarrolladores colaboren para crear código. Estos sistemas proporcionan un proceso claro para fusionar mediante combinación los cambios en el código que tienen lugar en los mismos archivos, controlar los conflictos y revertir los cambios a estados anteriores.

El uso del control de versiones es una práctica de DevOps fundamental que ayuda a los equipos de desarrollo a trabajar juntos, dividir las tareas de programación entre los miembros del equipo y almacenar todo el código para poder recuperarlo fácilmente si fuese necesario.

El control de versiones es también un elemento necesario en otras prácticas, como la integración continua y la infraestructura como código.

DESARROLLO ÁGIL DE SOFTWARE

El método ágil es un enfoque de desarrollo de software que hace hincapié en la colaboración en equipo, en los comentarios de los clientes y usuarios, y en una gran capacidad de adaptación a los cambios mediante ciclos cortos de lanzamiento de versiones. Los equipos que practican la metodología ágil proporcionan mejoras y cambios continuos a los clientes, recopilan sus comentarios y, después, aprenden y ajustan el software en función de lo que el cliente quiere y necesita. El método ágil es muy diferente a otros marcos más tradicionales, como el modelo en cascada, que incluye ciclos de lanzamiento de versiones largos definidos por fases secuenciales. Kanban y Scrum son dos marcos populares asociados al método ágil.

INFRAESTRUCTURA COMO CÓDIGO

La infraestructura como código define las topologías y los recursos del sistema de un modo descriptivo que permite a los equipos administrar esos recursos igual que lo harían con el código. Las diferentes versiones de esas definiciones se pueden almacenar en sistemas de control de versiones, donde se pueden revisar y revertir, de nuevo, igual que el código.

La práctica de la infraestructura como código permite a los equipos implementar recursos del sistema de un modo confiable, repetible y controlado. Además, la infraestructura como código ayuda a automatizar la implementación y reduce el riesgo de errores humanos, especialmente en entornos complejos de gran tamaño. Esta solución repetible y confiable para la implementación de entornos permite a los equipos mantener entornos de desarrollo y pruebas que sean idénticos al entorno de producción. De igual modo, la duplicación de entornos en otros centros de datos y en plataformas en la nube es más sencilla y más eficiente.

ADMINISTRACIÓN DE CONFIGURACIÓN

Administración de la configuración hace referencia a la administración del estado de los recursos de un sistema, incluidos los servidores, las máquinas virtuales y las bases de datos. El uso de herramientas de administración de la configuración permite a los equipos distribuir cambios de un modo controlado y sistemático, lo que reduce el riesgo de modificar la configuración del sistema. Los equipos utilizan herramientas de administración de la configuración para hacer un seguimiento del estado del sistema y evitar alteraciones en la configuración, que es como se desvía la configuración de un recurso del sistema a lo largo del tiempo del estado definido para él.

Al usarla junto con la infraestructura como código, resulta fácil elaborar plantillas y automatizar la definición y la configuración de sistemas, lo que permite a los equipos usar entornos complejos a escala.

SUPERVISIÓN CONTINUA

Supervisión continua significa tener visibilidad total y en tiempo real del rendimiento y el estado de toda la pila de aplicaciones, desde la infraestructura subyacente donde se ejecutan las aplicaciones hasta los componentes de software de niveles superiores. La visibilidad consiste en la recopilación de datos de telemetría y metadatos, así como en el establecimiento de alertas para condiciones predefinidas que garanticen la atención de un operador. La telemetría incluye registros y datos de eventos recopilados de varias partes del sistema que se almacenan donde pueden analizarse y consultarse.

Los equipos de DevOps de alto rendimiento se aseguran de establecer alertas útiles que les permitan tomar medidas y recopilan datos de telemetría muy completos para obtener conclusiones a partir de enormes cantidades de datos. Estas conclusiones ayudan a los equipos a mitigar los problemas en tiempo real y a ver cómo mejorar la aplicación en futuros ciclos de desarrollo.