

# Dev Ops

DevOps es una metodología de desarrollo de software que combina prácticas de desarrollo (Dev) y operaciones (Ops) para mejorar la colaboración y la eficiencia en la entrega de software. Se trata de una filosofía que busca integrar los equipos de desarrollo de software y los equipos de operaciones de TI para lograr una entrega de software más rápida y confiable. Se enfoca en la automatización de procesos de desarrollo y despliegue de software, la gestión de configuraciones y la monitorización y análisis del rendimiento del software en producción.

## Las fases de Dev Ops son:

### Planificación

En la fase de planificación, 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 realizan la planificación con agilidad y visibilidad.

### Desarrollo

La fase de desarrollo incluye todos los aspectos de la programación (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.

## 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.

## Uso

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 el gobierno. 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.

## Métodos de DevOps

Existen varios métodos de DevOps comunes que las organizaciones usan para acelerar y mejorar el desarrollo y las publicaciones de productos. Normalmente se presentan como prácticas y metodologías de desarrollo de software. Entre los más populares están Scrum, Kanban y Agile:

- **Scrum.** Scrum define la forma en la que los miembros de un equipo deben colaborar para conseguir entre todos acelerar los proyectos de desarrollo y control de calidad. Las prácticas de Scrum incluyen flujos de trabajo principales y terminología específica.
- **Kanban.** Es un sistema de información que controla de modo armónico la fabricación de los productos necesarios en la cantidad y tiempo necesarios en cada uno de los procesos que tienen lugar tanto en el interior de la fábrica, como entre distintas empresas.

- **Agile.** Es un método en donde se estipulan ciclos de desarrollo de software más cortos en lugar de los tradicionales métodos de desarrollo «en cascada» que se prolongaban en el tiempo.

## **Cultura de DevOps**

Aunque la adopción de prácticas de DevOps automatiza y optimiza los procesos con tecnología, todo comienza con la cultura interna de la organización y con las personas que participan en ella. El desafío de cultivar una cultura de DevOps requiere cambios profundos en la forma en la que las personas trabajan y colaboran. Pero cuando las organizaciones se comprometen a implementar una cultura de DevOps, pueden crear un entorno que facilite el desarrollo de equipos de alto rendimiento.

- **Colaboración, visibilidad y alineación**

Una buena cultura de DevOps se distingue, entre otras cosas, por la colaboración entre los equipos, que comienza con la visibilidad. Diferentes equipos, como el de desarrollo o el de operaciones de TI, deben compartir entre sí sus procesos de DevOps, sus prioridades y sus preocupaciones. Estos equipos también deben planificar juntos el trabajo y alinear sus objetivos y los indicadores del éxito en relación con el negocio.

- **Cambios en el ámbito y en la responsabilidad**

A medida que se alinean, los equipos asumen y participan en más fases del ciclo de vida, no solo las que son principales para su rol. Por ejemplo, los desarrolladores asumen responsabilidad no solo por la innovación y la calidad establecidas en la fase de desarrollo, sino también por el rendimiento y la estabilidad que sus cambios producen en la fase de uso. Al mismo tiempo, los operadores de TI se aseguran de incluir la gobernanza, la seguridad y el cumplimiento normativo en las fases de planificación y desarrollo.

- **Ciclos de lanzamiento de versiones más cortos**

Los equipos de DevOps mantienen la agilidad porque lanzan versiones de software en ciclos cortos. Los ciclos de lanzamiento de versiones más cortos facilitan el planificación y la administración de los riesgos, porque el progreso es incremental, lo que reduce el impacto en la estabilidad del sistema. El acortamiento de los ciclos de lanzamiento de versiones permite también a las

organizaciones adaptarse y reaccionar a las necesidades cambiantes de los clientes y a la presión competitiva.

- **Aprendizaje continuo**

Los equipos de DevOps de alto rendimiento establecen una mentalidad de crecimiento. Aceptan el fracaso y responden rápido a los errores, e incorporan lo que aprenden a los procesos, de modo que mejoran continuamente, aumentan la satisfacción del cliente, y agilizan la innovación y la capacidad de adaptación al mercado de forma constante. DevOps es un recorrido, por lo que siempre hay espacio para crecer.

## **Explicación de las prácticas de DevOps**

Existen unas cuantas prácticas fundamentales que ayudan a las organizaciones a innovar con mayor rapidez mediante la automatización y la simplificación de los procesos de desarrollo de software y administración de la infraestructura. La mayoría de estas prácticas se llevan a cabo con las herramientas adecuadas.

### **Integración continua**

La integración continua es una práctica de desarrollo de software mediante la cual los desarrolladores combinan los cambios en el código en un repositorio central de forma periódica, tras lo cual se ejecutan versiones y pruebas automáticas. Los objetivos clave de la integración continua consisten en encontrar y arreglar errores con mayor rapidez, mejorar la calidad del software y reducir el tiempo que se tarda en validar y publicar nuevas actualizaciones de software.

### **Entrega continua**

La entrega continua es una práctica de desarrollo de software mediante la cual se compilan, prueban y preparan automáticamente los cambios en el código y se entregan a la fase de producción. Amplía la integración continua al implementar todos los cambios en el código en un entorno de pruebas o de producción después de la fase de creación. Cuando se la entrega continua se implementa de manera adecuada, los desarrolladores dispondrán siempre de un artefacto listo para su implementación que se ha sometido a un proceso de pruebas estandarizado.

## **Microservicios**

La arquitectura de microservicios es un enfoque de diseño que sirve para crear una sola aplicación como conjunto de servicios pequeños. Cada servicio se ejecuta en su propio proceso y se comunica con otros servicios mediante una interfaz bien definida utilizando un mecanismo ligero, normalmente una interfaz de programación de aplicaciones basada en HTTP (API). Los microservicios se crean en torno a las capacidades empresariales. Cada servicio abarca un único propósito.

## **Infraestructura como código**

La infraestructura como código es una práctica mediante la que se aprovisiona y administra infraestructura con técnicas de desarrollo de código y de software, como el control de versiones y la integración continua. El modelo orientado a la API de la nube permite a los desarrolladores y administradores de sistemas interactuar con la infraestructura mediante programación y a escala, en lugar de configurar y ajustar recursos manualmente. Así, los ingenieros pueden interactuar con la infraestructura con herramientas basadas en código y tratar la infraestructura de un modo parecido a como tratan el código de la aplicación.

## **Monitoreo y registro**

Las organizaciones monitorean métricas y registros para ver cómo el desempeño de las aplicaciones y la infraestructura afecta a la experiencia que el usuario final tiene de su producto. Cuando recopilan, categorizan y analizan los datos y registros generados por las aplicaciones y la infraestructura, las organizaciones pueden entender cómo los cambios y actualizaciones afectan a los usuarios, esto les aporta información sobre la causa raíz de los problemas o cambios inesperados. El monitoreo activo se vuelve cada vez más importante, ya que los servicios deben estar disponibles las 24 horas del día, los 7 días de la semana, a medida que la frecuencia de actualizaciones de las aplicaciones e infraestructura incrementa.

## **Comunicación y colaboración**

El incremento en la comunicación y la colaboración en una organización es uno de los aspectos culturales clave de DevOps. El uso de las herramientas de DevOps y la automatización del proceso de entrega de software establece la colaboración al reunir físicamente los flujos de trabajo y las responsabilidades de los equipos de desarrollo y operaciones. Además, estos equipos establecen normas culturales sólidas que giran en torno a compartir información y facilitar la comunicación mediante el uso de aplicaciones de chat, sistemas de seguimiento de proyectos o problemas y wikis.

## **¿Por qué es tan importante DevOps?**

Uno de los principales valores de DevOps es lograr la satisfacción del cliente y prestar sus servicios en menos tiempo. DevOps también se ha creado para impulsar la innovación empresarial y ser el motor de continuas mejoras en los procesos.

## **Ventajas de DevOps**

Los partidarios de DevOps describen varias ventajas técnicas y empresariales con las que, en última instancia, se consiguen clientes más satisfechos. Entre algunas de las ventajas de DevOps se incluyen las siguientes:

- Una mejor y más rápida entrega de productos
- Resolución de problemas en menos tiempo y con menor complejidad
- Mejor escalabilidad y disponibilidad
- Entornos de funcionamiento más estables
- Mejor utilización de los recursos
- Mayor automatización
- Mayor visibilidad de resultados del sistema
- Mayor innovación