

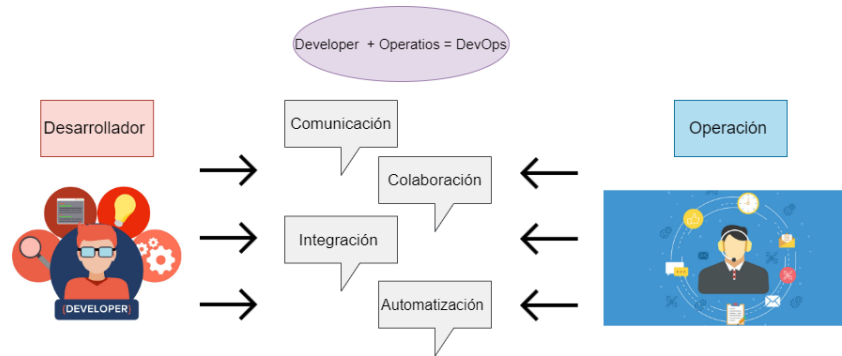
# DevOps

Natalia Reyes Altamirano



# ¿Qué es un DevOps?

- DevOps es un conjunto de prácticas, herramientas y filosofía cultural que sirve para automatizar e integrar los procesos que comparten el equipo de desarrollo de software y el de TI.
- Se centra en el empoderamiento de los equipos, la comunicación y colaboración entre ellos y la automatización de la tecnología.
- El término "DevOps", es una combinación de las palabras "development" (desarrollo) y "operations" (operaciones), y refleja el proceso de integración de estas dos disciplinas en un único proceso continuo.



# ¿En qué consiste DevOps?

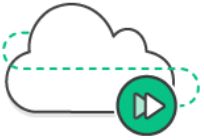
- Los equipos de desarrollo y operaciones ya no están “aislados”. A veces, los dos equipos se fusionan en uno solo, donde los ingenieros trabajan en todo el ciclo de vida de la aplicación, desde el desarrollo y las pruebas hasta la implementación y las operaciones, y desarrollan una variedad de habilidades no limitadas a una única función.
- En algunos modelos de DevOps, los equipos de control de calidad y de seguridad también se integran más con el desarrollo y las operaciones e intervienen durante todo el ciclo de vida de la aplicación.
- Los equipos utilizan prácticas para automatizar los procesos que anteriormente habían sido manuales y lentos. Utilizan una pila de tecnología y herramientas que los ayudan a operar y mejorar aplicaciones de forma rápida y confiable. Además, estas herramientas ayudan a los ingenieros a realizar de forma independiente tareas que normalmente hubieran requerido la ayuda de otros equipos (por ejemplo, implementar código o aprovisionar infraestructura), lo que incrementa todavía más la velocidad del equipo



# Ventajas del modelo DevOps



- Velocidad: El modelo permite avanzar a gran velocidad e innovar con mayor rapidez. Los microservicios y la entrega continua permiten que los equipos se hagan responsables de los servicios y los actualicen con mayor rapidez.



- Entrega rápida: Se incrementa la frecuencia y el ritmo de las versiones, a fin de innovar y mejorar el producto con mayor rapidez. La integración continua y la entrega continua son prácticas que automatizan el proceso de publicación de software, desde la creación hasta la implementación.



- Calidad y fiabilidad: Prácticas como la integración continua y la entrega continua se encargan de que los cambios funcionen correctamente y sean seguros, lo que mejora la calidad de los productos de software.



# Ventajas del modelo DevOps



- Seguridad: Se integra la seguridad en una canalización de integración continua, entrega continua e implementación continua. La seguridad se con auditorías de seguridad activas y pruebas de seguridad



- Colaboración mejorada: La base de DevOps es una cultura de colaboración entre los desarrolladores y los equipos de operaciones, que comparten responsabilidades y combinan el trabajo. Esto hace que los equipos sean más eficientes.



- Implementación rápida: Al aumentar la frecuencia y la velocidad de los lanzamientos, los equipos de DevOps mejoran los productos rápidamente.



# Prácticas de DevOps



- Integración continua: En ella los desarrolladores combinan los cambios en el código en un repositorio central de forma periódica. Los objetivos son 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: Los cambios en el código se compilan, prueban y preparan automáticamente y se entregan a la fase de producción.



- 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).



# Prácticas de DevOps



- Infraestructura como código: Se aprovisiona y administra infraestructura con técnicas de desarrollo de código y de software, como el control de versiones y la integración continua.



- 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.



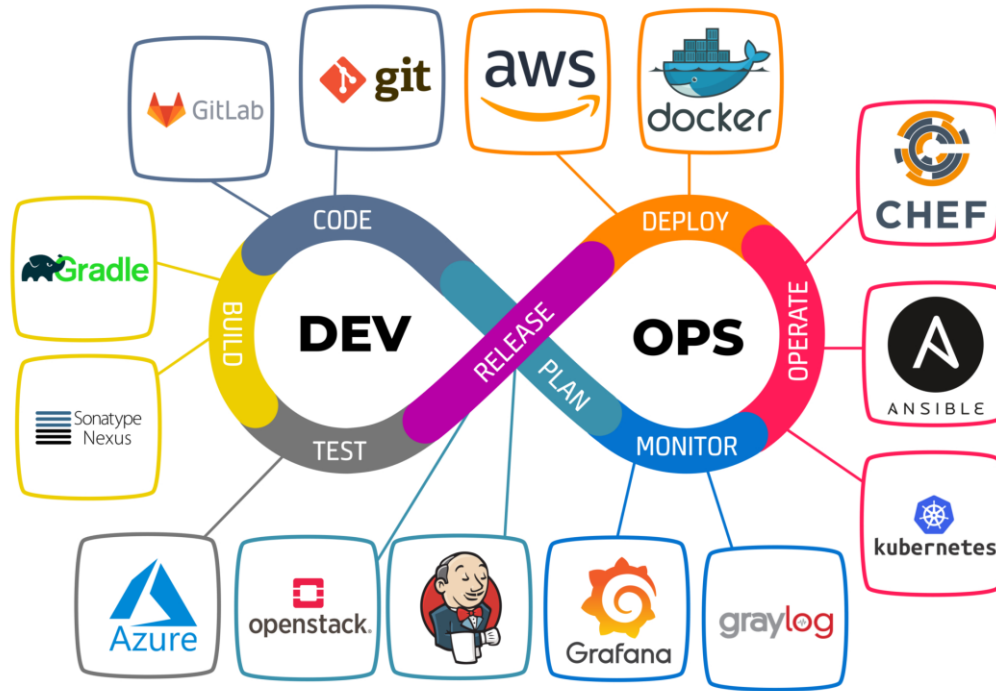
- Automatización: Permite a los equipos avanzar mucho más rápido en el proceso de desarrollo e implementación de software de alta calidad.



- Supervisión: Se supervisa todo el proceso de desarrollo, desde la planificación, el desarrollo, la integración y las pruebas a la implementación y las operaciones.



# Ciclo de vida DevOps





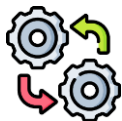
# Ciclo de vida DevOps



- **Plan (Planificación y Diseño):** En esta etapa, los equipos planifican el desarrollo y despliegue del software. Definen los objetivos, requisitos y funcionalidades del proyecto. Se establecen los planes de desarrollo, se diseña la arquitectura y se determinan los recursos necesarios.



- **Monitor (Monitorización):** En esta etapa, se establecen sistemas de monitorización para supervisar el rendimiento, la disponibilidad y otros aspectos del software en producción. Esto permite detectar problemas en tiempo real y tomar medidas para resolverlos antes de que afecten a los usuarios.



- **Operate (Operación y Monitorización):** Aquí se gestionan las operaciones continuas del software en producción. Esto incluye la administración de servidores, la resolución de problemas y la escalabilidad según la demanda. Se aplican parches y actualizaciones, y se mantienen los sistemas en funcionamiento.



DEPLOYMENT

- **Deploy (Despliegue):** En esta etapa, el software se implementa en el entorno de producción o preproducción. Se realiza el proceso de despliegue, que puede incluir la configuración de servidores, la instalación de dependencias y la puesta en marcha del software en sí.



# Ciclo de vida DevOps



- **Release (Liberación):** La liberación implica poner a disposición del público una nueva versión del software. Esto puede involucrar la preparación de la versión, la documentación y la comunicación con los usuarios. La liberación puede estar ligada a cambios importantes en el software.



- **Test (Pruebas):** En esta etapa, se realizan pruebas de calidad en el software. Esto incluye pruebas unitarias, de integración y de sistema para asegurarse de que el software funcione correctamente y cumpla con los requisitos establecidos.



- **Build (Construcción):** En esta fase, se compila y construye el código fuente para crear el ejecutable o el paquete de software. También puede incluir la creación de artefactos como contenedores Docker u otros elementos necesarios para el despliegue.



- **Code (Desarrollo):** Esta es la etapa en la que se desarrolla el código fuente del software. Los desarrolladores escriben y prueban el código para implementar nuevas funcionalidades o corregir problemas existentes. El código es la base del software y se gestiona a través de sistemas de control de versiones.

