



DevOps para la transformación digital

Cómo DevOps puede transformar las TI y el negocio



Qué encontrarás aquí

03

CAPÍTULO 1

Introducción

04

CAPÍTULO 2

Conociendo DevOps y SRE

06

CAPÍTULO 3

Fundamentos de DevOps

10

CAPÍTULO 4

Las mejores prácticas de DevOps

15

CAPÍTULO 5

Beneficios de DevOps

19

CAPÍTULO 6

Desafíos de DevOps

22

CAPÍTULO 7

Métricas de DevOps

26

CAPÍTULO 8

DevOps y la observabilidad inteligente permiten la transformación digital



CAPÍTULO 1

Introducción

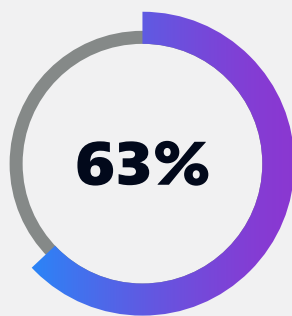
Cada vez más organizaciones han adoptado prácticas de DevOps para agilizar el desarrollo de software, aumentar la productividad de los desarrolladores y mejorar los flujos de trabajo de entrega continua para ofrecer un mejor software más rápido.

Como señaló el pionero de DevOps [Patrick Debois](#) en 2009, las tácticas (no sólo las soluciones tecnológicas) definen un enfoque exitoso de DevOps que puede transformar fundamentalmente la TI. Pero aunque este enfoque táctico ofrece una mayor flexibilidad para los equipos, puede conducir rápidamente a silos de datos y comunicación en toda la organización que pueden perjudicar la calidad del software y la velocidad de entrega. Sin ayuda, puede ser extremadamente difícil obtener una visión estratégica de cómo los equipos de desarrollo realizan sus tareas cotidianas, cómo automatizar las canalizaciones de DevOps y cómo diseñar el software para que sea fiable y resistente en entornos modernos nativos de la nube.

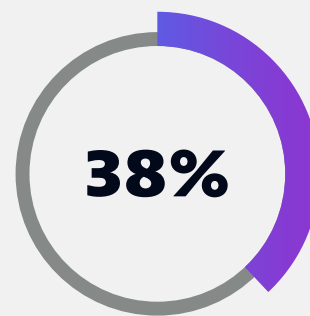
Esa ayuda llega en forma de inteligencia artificial (IA). AIOps, la disciplina de aplicar la IA y los análisis avanzados a las operaciones de TI, ha transformado la forma en que las organizaciones gestionan los sistemas complejos. Utilizando estos mismos principios, las organizaciones pueden adoptar un enfoque más inteligente de DevOps, un enfoque AIOps de DevOps que aprovecha la IA en todo el ciclo de vida de desarrollo de software (SDLC). DevOps, junto con tecnologías y tácticas complementarias, como la ingeniería de fiabilidad del sitio (SRE), tiene el potencial de transformar el negocio.

Para entender mejor el poder transformador de DevOps, exploraremos los fundamentos de DevOps y el creciente papel de SRE; profundizaremos en los beneficios y desafíos clave de DevOps; discutiremos las mejores prácticas de DevOps y las métricas clave de DevOps; y examinaremos cómo la IA y la automatización en cada etapa del ciclo de vida de DevOps pueden transformar la forma en que las organizaciones desarrollan y entregan mejor software más rápido.

DevOps tiene poder potencial de remodelar su negocio al agilizar su TI para ofrecer un mejor valor:¹

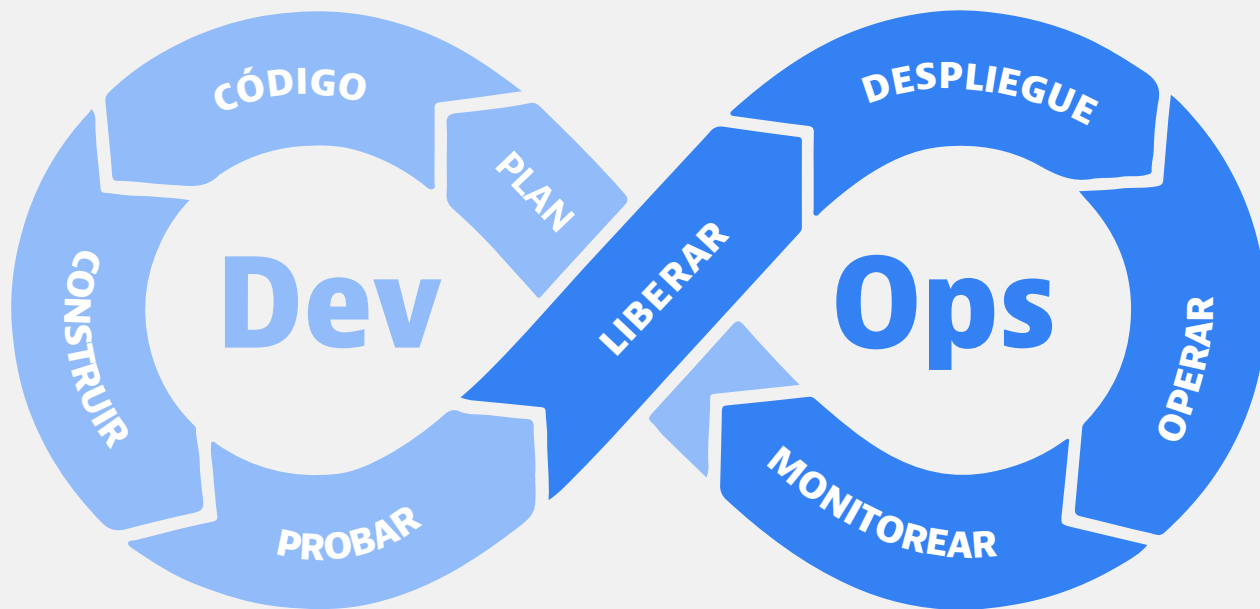


de las organizaciones que utilizan prácticas DevOps **han mejorado la calidad de sus implementaciones.**



de las organizaciones informan **la mejora de calidad del código en general.**

¹Estadísticas de DevOps para los que dudan.. <https://www.upguard.com/blog/devops-success-stats#toc-1>. Agosto de 2021.



CAPÍTULO 2

Conociendo DevOps y SRE

¿Qué es DevOps? El desarrollo se une a las operaciones de TI

La forma más fácil de conceptualizar DevOps es como un bucle continuo. En lugar de procesos discretos, el desarrollo y las operaciones pasan a formar parte de un ciclo continuo que incluye la planificación, la codificación, la construcción, las pruebas, el lanzamiento, el despliegue, el funcionamiento y el monitoreo de aplicaciones y servicios.

La forma más fácil de conceptualizar DevOps es como un bucle continuo. En lugar de procesos discretos, el desarrollo y las operaciones pasan a formar parte de un ciclo continuo que incluye la planificación, la codificación, la construcción, las pruebas, el lanzamiento, el despliegue, el funcionamiento

y el monitoreo de aplicaciones y servicios.

Este enfoque de flujo de trabajo continuo permite a los equipos identificar y abordar inmediatamente los problemas relacionados tanto con la forma como con la función en una fase más temprana del proceso para evitar problemas antes de que el software se ponga en producción.

El reciente desarrollo de las tecnologías nativas de nube, las soluciones de código abierto y las API flexibles han mejorado aún más la eficiencia de DevOps. Con sus raíces en el desarrollo ágil, DevOps es ideal para ayudar a los equipos a seguir el ritmo de la aceleración de los modelos de desarrollo y lanzamiento, como la integración continua y la entrega continua (CI/CD).

¿Qué es SRE? Resiliencia del software incorporada

SRE es una práctica de operaciones de software que gestiona los detalles y las preocupaciones a gran escala de la resiliencia del software para garantizar la disponibilidad, la latencia, el rendimiento y la capacidad de los sistemas de software. Los ingenieros de fiabilidad del sitio entienden las necesidades de los sistemas de software y establecen procesos y estructuras para satisfacer esas necesidades.

El vicepresidente de ingeniería de Google, Ben Sloss, acuñó el término SRE en 2003, cuando él y su equipo comenzaron a aplicar los principios de la ingeniería de software a las operaciones de software para crear sistemas de software más fiables y escalables. La implementación de SRE puede ayudar a las organizaciones a reducir la fricción entre los componentes de desarrollo y operaciones de los equipos de DevOps, agilizando la eficiencia y reduciendo las tasas de error.

La SRE complementa las prácticas de DevOps ofreciendo una mayor automatización para reducir la dependencia de las tareas manuales. Estas prácticas ayudan a los equipos de desarrollo a resolver sus problemas y a ofrecer fiabilidad por diseño en una fase más temprana del proceso de desarrollo.

En última instancia, SRE ayuda a las organizaciones a alcanzar sus objetivos operativos, como la reducción del tiempo de inactividad o resoluciones más rápidas, mediante la definición y automatización de los objetivos de nivel de servicio (SLO).

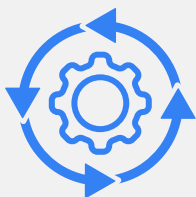
¿Cómo interactúan SRE y DevOps?

SRE y DevOps son esencialmente dos caras de la misma moneda. Mientras que los marcos de DevOps se centran en la colaboración a lo largo de todo el ciclo de vida y en la eliminación de los silos, una SRE sólida ayuda a implementar y automatizar las prácticas de DevOps utilizando los SLO y garantiza que esos sistemas –y el software que producen– sean resistentes.

Según Andi Grabner, activista de DevOps en Dynatrace, “DevOps y SRE son un equilibrio entre velocidad y seguridad”. Mientras que DevOps ayuda a las organizaciones a moverse de izquierda a derecha a lo largo del ciclo de vida del desarrollo y las operaciones para impulsar la velocidad general, SRE se mueve de derecha a izquierda para ayudar a reducir las tasas de fracaso en las primeras etapas del ciclo de desarrollo.

Aunque es posible tener DevOps sin SRE, estos dos procesos funcionan mejor en paralelo, creando efectivamente un ciclo continuo que ofrece mejoras continuas en ambas direcciones de la tubería de CI/CD. Como práctica combinada, las empresas pueden tratar de aumentar la automatización, acelerar la entrega, mejorar la calidad del software y mucho más.

Con esta información de fondo en la mano, vamos a sumergirnos ahora en los componentes fundamentales de la mentalidad DevOps para tener una mejor idea de dónde tendrá que concentrar su equipo sus esfuerzos para sacar el máximo provecho de las prácticas DevOps.



Las SRE se pusieron en práctica para aumentar la resistencia de las organizaciones mediante la **automatización de muchas tareas manuales** en las primeras fases del desarrollo.



CAPÍTULO 3

Fundamentos de DevOps

DevOps es un cambio cultural que requiere visión, planificación, aceptación por parte de los ejecutivos y una estrecha colaboración para establecer con éxito una forma más integrada de desarrollar y entregar aplicaciones. Al adoptar algunas prácticas fundamentales, los equipos pueden mejorar su eficiencia y desarrollar una comprensión más profunda de sus flujos de trabajo, conjuntos de herramientas y procesos para que puedan lanzar un mejor software más rápido.

Dado que DevOps es un proceso continuo, estas prácticas también deben ser continuas y permanentes. Este capítulo cubre los principios básicos o prácticas que forman los fundamentos de la adopción de un enfoque DevOps.

Integración continua

La integración continua (CI) es una práctica de desarrollo de software en la que los desarrolladores envían regularmente su código a un repositorio compartido. Dado que la arquitectura de microservicios está distribuida, la CI permite a los desarrolladores poseer trozos de código discretos y manejables, así como características individuales, y trabajar en ellas en paralelo. La naturaleza distribuida de estas aplicaciones permite actualizaciones frecuentes, a menudo varias veces al día. Sin embargo, los desarrolladores no pueden enviar actualizaciones al azar. CI está muy controlado; los nuevos commits activan la creación de

nuevas compilaciones de prueba a través de el sistema de gestión de la compilación. Se rechaza el código redundante y se minimizan los cambios de ruptura una vez que se alteran las ramas maestras. Se fomentan los cambios incrementales. Además, la reducción de la conciliación evita la congelación obligatoria del código que suele derivarse de los conflictos. En general, la integración continua permite a los equipos crear y probar el software con mayor rapidez y eficacia. Al fusionar el código con regularidad, los equipos también tienen siempre una compilación actualizada que acelera las pruebas y la corrección de errores, aumenta la confianza en la fusión y ayuda a acortar el proceso de desarrollo.

Entrega continua

Mientras que la CI se centra en actualizaciones regulares e independientes del código en un repositorio central, la entrega continua (CD) se centra en la liberación de bloques de código completos en un repositorio a intervalos regulares. Estos bloques de código deben estar siempre en un estado desplegable para las pruebas o la liberación a la producción. La CD se confunde a menudo con el despliegue continuo -el siguiente proceso en la línea- que libera el código finalizado en producción. El despliegue es el acto de poner a disposición de los usuarios finales software nuevo y actualizado. En consecuencia, la CD denota principalmente "entrega continua", o tanto "entrega continua como despliegue", pero rara vez sólo despliegue continuo. La cd toma el código y lo añade a un repositorio, como GitHub o, en el caso de un entorno basado en microservicios, un registro de contenedores. El objetivo final es aumentar la consistencia de las versiones manteniendo el código en un estado de despliegue permanente. El desarrollo de software se vuelve más ágil y más predecible como resultado.

Pruebas continuas y validación

Las pruebas continuas en DevOps son importantes en cada etapa del SDLC. Implica a muchas partes interesadas, como el equipo de desarrollo, el control de calidad y el personal operativo. El objetivo de las pruebas continuas es evaluar la calidad del software a medida que avanza por cada etapa del ciclo de vida de la entrega. Esto no sólo detiene el código

defectuoso en su camino, sino que también proporciona una retroalimentación rápida y continua a los equipos de desarrollo con la información que necesitan para abordar cualquier problema de calidad.

Aunque las pruebas continuas son importantes para escalar, la validación manual de los resultados de las pruebas hace descarrilar el proceso de entrega del software. Aquí es donde entra en juego la validación continua, que automatiza el proceso de evaluación de los resultados de las pruebas con respecto a sus objetivos de nivel de servicio predefinidos. La validación continua complementa la implementación de las pruebas continuas eliminando cualquier análisis manual necesario, ya sea comparando datos en tableros o marcando casillas en una hoja de cálculo. En su lugar, los equipos de DevOps pueden establecer técnicas como las puertas de calidad que hacen cumplir automáticamente los criterios de calidad predefinidos y evitan que el código defectuoso pase a la siguiente fase.



Monitoreo continuo y observabilidad

Aunque las organizaciones se esfuerzan por conseguir procesos de CI/CD herméticos, a menudo hay oportunidades de mejora. El monitoreo y la capacidad de observación son fundamentales para comprender la viabilidad del código a medida que avanza por el proceso. Aunque la detección de problemas y vulnerabilidades siempre es importante, la gran cantidad de datos de observabilidad asociados a las modernas aplicaciones multicloud significa que simplemente no hay forma de realizar un seguimiento manual de todo lo que ocurre en la pila de software.

Tradicionalmente, el monitoreo de DevOps se asociaba estrechamente con los equipos de “operaciones”, pero desde entonces ha evolucionado a lo largo de todo el ciclo de vida del desarrollo de software (SDLC), ya que las partes interesadas clave exigen cada vez más respuestas para resolver sus problemas con mayor rapidez. Estas respuestas son posibles cuando se dispone de un sistema que monitoriza y analiza continuamente los datos de observabilidad. La captura continua de datos puede ser muy poderosa cuando se aprovecha de forma inteligente, y aquí es donde la introducción de la observabilidad es fundamental para DevOps. La observabilidad es algo más que recopilar métricas y organizarlas en cuadros de mando. Contar con un motor de IA que trabaje las 24 horas del día, los 365 días del año, para analizar los datos y proporcionar respuestas a las anomalías y los problemas ayuda a los equipos a solucionar los problemas más rápidamente y a tomar mejores decisiones de lanzamiento. Esto conduce a una mejor calidad del código y a un mejor rendimiento de la aplicación, lo que se traduce en una mejor experiencia para el usuario final. A medida que aumenta la complejidad del software, se hace más difícil para los equipos de DevOps entregar nuevas características y lanzamientos más rápidamente sin sacrificar la calidad. Por lo tanto, potenciar a sus equipos con observabilidad continua y un motor de IA

para analizar todos los datos y proporcionar respuestas es fundamental para el éxito.

Seguridad continua

Otra práctica fundamental de DevOps es la [seguridad continua](#) basada en las pruebas, el monitoreo, la autorización y el seguimiento del inventario. Esta es la evolución hacia DevSecOps. En pocas palabras, la seguridad continua es el proceso de hacer que la seguridad forme parte del proceso CI/CD, cubriendo todo el SDLC, añadiendo una capa extra sobre el proceso y los pipelines de DevOps para asegurar que su infraestructura y aplicaciones no tienen vulnerabilidades y riesgos asociados. Hoy en día, a medida que vemos que los entornos se vuelven cada vez más complejos, un enfoque de la seguridad como algo adicional no es escalable ni sostenible y, por lo tanto, es imperativo incorporar la seguridad a los procesos automatizados para permitir las pruebas continuas a lo largo del ciclo de vida del desarrollo.

Al igual que la mentalidad de cambio a la izquierda con respecto a la calidad en las pruebas, las medidas de seguridad deben incorporarse a la planificación y la creación desde el primer día, y deben producirse constantemente a lo largo del ciclo de vida del desarrollo, incluso cuando

DevOps es un cambio cultural

Esta nueva cultura requiere la adopción de mejores prácticas como



**Entrega
continua**



**Integración
continua**



**Pruebas
continuas**



**Monitorización
continua /
observabilidad**



**Colaboración
entre equipos**

el software se ejecuta en producción. Además, cualquier medida de seguridad continua que se implemente debe ser automatizada según corresponda, para no obstaculizar la eficiencia. En términos de la cultura, el personal de seguridad debe ser considerado como socios de pleno derecho en el proceso de DevOps, a la par que los desarrolladores y los especialistas en operaciones, de ahí el cambio hacia DevSecOps.

Colaboración entre equipos

Romper con los silos es primordial para garantizar una buena comunicación y unificación en todo el proceso de DevOps. La ejecución efectiva de DevOps significa establecer una única fuente de verdad, agregando datos de muchas fuentes en una ubicación colectiva. Los probadores, los ingenieros, el departamento de control de calidad e incluso las partes interesadas no técnicas pueden obtener información valiosa de estos datos. Bajo este paradigma, cada uno contribuirá a su manera a la creación de software que impulse resultados empresariales de alto nivel.

La cooperación perfecta entre los desarrolladores y los equipos de operaciones es especialmente importante. Los procesos DevOps coexisten en un ciclo continuo conocido como [bucle de retroalimentación](#). Las diferentes partes de

un proyecto se completan y son revisadas por las partes interesadas, y la retroalimentación se devuelve a partir de esos pasos. El código debe escribirse, probarse, validarse, entregarse, construirse y, finalmente, desplegarse.

La colaboración entre equipos se beneficia de la aceleración de este ciclo. En consecuencia, la automatización se ha convertido en un ingrediente clave para acortar el ciclo de extremo a extremo, principalmente al reducir la fricción causada por las múltiples partes que trabajan al mismo tiempo.

Llevar DevOps al siguiente nivel

Aunque hemos establecido los fundamentos, las organizaciones no quieren conformarse con lo básico, sino que quieren llevar sus DevOps al siguiente nivel. Para seguir el ritmo de la innovación, y la necesidad de entregar servicios al mercado más rápidamente, las organizaciones necesitan llevar sus prácticas de DevOps al siguiente nivel. La evolución de estas prácticas fundamentales hacia un DevOps de rendimiento de élite requiere algunas mejores prácticas, que cubriremos en el próximo capítulo.





CAPÍTULO 4

Las mejores prácticas de DevOps

Muchas organizaciones afirman tener un proceso DevOps en pleno funcionamiento. Pero DevOps es más que un flujo de trabajo y unas pocas herramientas que su organización puede implementar y seguir adelante. Es útil pensar en ello como una filosofía -una cultura y una mentalidad- que requiere una optimización continua, creatividad y flexibilidad para mantenerse. Tal vez una organización ha implementado cambios organizativos y herramientas que sientan las bases para un buen proceso de DevOps, pero pueden estar perdiendo algunos de los beneficios que DevOps puede ofrecer.

Con este viaje de mejora en mente, exploremos algunas de las mejores prácticas de DevOps que pueden llevar su inversión en los fundamentos al siguiente nivel y asegurar que está sacando el máximo provecho de su estrategia de DevOps.



Automatización

La automatización es la piedra angular de la estrategia DevOps de toda empresa. En resumen, la automatización reduce el trabajo, le ayuda a acelerar sus canales de entrega en todo el SDLC y le permite escalar su práctica de DevOps. Tradicionalmente, procesos como las pruebas, la monitorización, el descubrimiento de errores y la corrección de los mismos comprendían un poco de automatización y mucha intervención manual. Esto funcionaba cuando los equipos pequeños trabajaban en aplicaciones monolíticas. Pero con los modernos microservicios basados en aplicaciones y con la transformación digital ejerciendo aún más presión sobre las TI, la automatización es crucial para aumentar la velocidad y la calidad impulsando procesos coherentes en cada etapa del ciclo de vida de DevOps. Como

resultado, puede enviar el código a producción con más frecuencia y producir software consistente, fiable y seguro, al tiempo que ahorra a su equipo de DevOps un tiempo valioso que pueden dedicar a la innovación.



Monitoreo y observabilidad

La monitorización y la observabilidad son esenciales para incorporar en cada etapa del ciclo de vida del desarrollo de software, desde la preproducción hasta la producción. Mientras que la automatización de tantos procesos como sea posible aumenta la eficiencia de sus flujos de trabajo de DevOps, el monitoreo y la observabilidad proporcionan a sus equipos visibilidad en esos procesos automatizados para detectar y señalar las causas de raíz de cualquier problema o cuello de botella.

Muchas herramientas proporcionan datos y cuadros de mando para hacer un seguimiento del estado de los sistemas individuales. Pero para desarrollar una estrategia de observabilidad eficaz que produzca respuestas procesables sobre los sistemas en toda la cadena de herramientas de DevOps, se necesita algo más que datos en los cuadros de mando: se necesita un enfoque inteligente.

- 1. Haga que sus sistemas sean observables:** adopte un estándar, como OpenTelemetry. Aproveche una plataforma de observabilidad basada en IA que pueda instrumentar y detectar automáticamente las anomalías, para que no tenga que hacerlo manualmente.
- 2. Establezca la observabilidad de extremo a extremo** desde la preproducción hasta la producción para cada aplicación o entorno.
- 3. Asegurarse de que puede comprender el impacto de negocio** de un evento o transacción analizándolo en el contexto de los procesos anteriores y posteriores a él.
- 4. Aproveche la AI** para detectar automáticamente los problemas y proporcionar un análisis para identificar inmediatamente las causas de origen y activar la reparación automática.

Esta es una práctica importante que debe implementarse para que su equipo pueda identificar fallos o problemas de rendimiento antes de que los clientes sientan cualquier impacto. *impact is felt by your customers.*



AIOps

Los datos pueden ser el mejor amigo de un equipo de TI, especialmente cuando se trata de probar y entregar el código y monitorear los servicios de manera más eficiente. Sin embargo, el procesamiento de la enorme cantidad de datos creados por las aplicaciones actuales está más allá de la capacidad de los humanos por sí solos. Esto abre el camino a un motor de IA que pueda analizar constantemente todos los datos observables hasta el nivel de detalle del código, y que dé al equipo de desarrollo el poder de identificar problemas, obtener respuestas y remediar rápidamente los problemas cuando se produzcan.

Aprovechar la IA como parte de sus procesos de DevOps le permite mejorar la funcionalidad y la automatización en los ciclos de desarrollo, pruebas, seguridad, entrega y lanzamiento, así como monitorear constantemente el rendimiento del software desplegado de forma mucho más eficiente que utilizando esfuerzos manuales.



Calidad Shift-left

Los SRE viven y respiran los objetivos de nivel de servicio (SLO). Garantizar que los niveles de servicio de producción se ajustan a lo previsto requiere una evaluación continua de los indicadores de nivel de servicio (SLI) con respecto a los SLO. Pero esto plantea la siguiente pregunta: ¿por qué los desarrolladores no deberían asegurarse de que el código que construyen cumple los mismos SLO de producción? Este concepto de desplazamiento a la izquierda mejora la calidad del software, ayuda a detectar problemas mucho antes en el ciclo de vida e impide que el código que no cumple los

SLO de producción avance a la siguiente fase. Los resultados son un menor número de violaciones de SLO en producción, un ahorro de tiempo y dinero debido a la disminución o ausencia de salas de guerra, pero lo más importante es que se garantiza el cumplimiento del 100% de los acuerdos de nivel de servicio (SLA) del negocio.

Una forma de automatizar este proceso de cambio es a través de las puertas de calidad, que permiten comparar automáticamente los SLI de cualquier herramienta de canalización (como monitoreo y las pruebas) con los SLO predefinidos. Si el código no pasa la puerta de calidad basada en los SLO, no puede pasar a la siguiente fase, y el sistema notifica automáticamente al equipo de desarrollo para que solucione el problema.



Fiabilidad shift-right

La entrega progresiva (también conocida como shift-right) se centra en ampliar las prácticas generales de CI/CD para ayudar a entregar aplicaciones y servicios con más control. Permite a las organizaciones gestionar con precisión cómo y cuándo se entregan las nuevas funciones, actualizaciones y correcciones para minimizar el posible impacto negativo en la base de usuarios. Algunas de las prácticas más comunes son las implantaciones de color verde azulado, las pruebas A/B, las implantaciones de tipo canario y los indicadores de características.

- **Despliegues blue-green**

Este modelo de despliegue de aplicaciones hace que los usuarios pasen gradualmente de una versión actual de una aplicación o servicio (la versión "azul") a una nueva versión (la versión "verde") mientras tanto la azul como la verde están funcionando en producción. Este cambio debe ser perfecto para el usuario, y la versión azul puede estar preparada en caso de que un problema imprevisto con la verde requiera la vuelta a la versión anterior, más estable.

- **Pruebas A/B**

También conocido como split testing, el A/B testing se

refiere a procesos de experimentación aleatorios en los que dos o más versiones de alguna variable -por ejemplo, un servicio, una página web o un elemento de la página- se muestran a diferentes usuarios finales. A partir de ahí, se puede monitorear el rendimiento de la aplicación, así como el comportamiento y la satisfacción de los usuarios, para determinar qué opción es la mejor para los objetivos del negocio.

- **Indicadores Toggles**

También conocidos como toggles, los feature flags son una práctica de desarrollo que permite a los equipos de software y desarrollo activar y desactivar partes de un código base con un simple interruptor (o flag). Estos toggles ayudan a las organizaciones a desacoplar los despliegues de código de los lanzamientos de características, lo que les permite realizar cambios de código en producción que permanecen ocultos para los usuarios hasta que se activan. El resultado es una mayor velocidad de despliegue, una mayor estabilidad del sistema y una mejor colaboración entre equipos.

- **Despliegues Canary**

Todos los despliegues en producción conllevan un riesgo, incluso con monitoreo y pruebas exhaustivas. Un método para que los desarrolladores mitiguen las interrupciones graves es el despliegue de canarios. El término tiene su origen en la época en que los canarios se utilizaban para detectar gases tóxicos en las minas de carbón. Si el canario moría, los mineros sabían que debían salir antes de que el gas les alcanzara. Un despliegue de canarios es una versión de software que se despliega a un pequeño porcentaje -denominado canario- de toda la base de usuarios. Si las cosas funcionan bien en tu canario, puedes desplegar la versión al resto de la base de usuarios. Si las cosas no funcionan bien, al menos el impacto es mucho menor, menos perturbador, y se puede revertir el software. Los despliegues canarios ofrecen la posibilidad de probar a los usuarios reales, que pueden proporcionar información real, al tiempo que reducen el riesgo al mitigar el impacto si los problemas conducen a un producto de mejor calidad.



Seguridad shift-left & shift-right

Los conceptos de shift-left y shift-right también se aplican a la seguridad. En primer lugar, hablemos del turno a la izquierda.

Desde que los equipos de DevOps empezaron a utilizar contenedores como forma de empaquetar aplicaciones y empezaron a lanzar software a una cadencia más rápida, ha habido un deseo de automatizar las pruebas de seguridad de las aplicaciones y proporcionar los resultados de las pruebas antes en el ciclo de vida de desarrollo del software. Al proporcionar los resultados de las pruebas antes, los desarrolladores de software pueden corregir los fallos de seguridad con mayor rapidez y facilidad. No tienen que recordar un cambio que hicieron hace semanas y que introdujo accidentalmente una vulnerabilidad de seguridad, y desentrañar todo lo que se ha hecho desde entonces.

Además de la información que se proporciona antes, las decisiones de liberación automatizada pueden hacerse antes, basándose en los resultados de las pruebas de seguridad. Este ha sido el santo grial de DevSecOps: proporcionar más automatización y menos trabajo manual. El resultado es un software mejor, de mayor rendimiento y más seguro, con menos trabajo para los seres humanos.

¿Y la seguridad shift-right? También es importante. Tras varios años de “cambio a la izquierda”, las empresas se están dando cuenta de que también necesitan mantener la visibilidad en el entorno de producción. Hemos visto muchos ataques exitosos contra entornos Kubernetes, desde las imágenes maliciosas que se insertaron en el hub de Docker en 2020, hasta los ataques contra Azure y Tesla por parte de “cryptojackers”. Por ello, el 44% de las empresas afirma que tiene previsto adoptar nuevos controles de seguridad en tiempo de ejecución (shift-right) en los próximos 12-24 meses.

En pocas palabras, estas son las razones por las que la seguridad automatizada (DevSecOps) puede y debe pasar a

los entornos de producción:

- El entorno de producción está conectado a Internet, que es donde se producen la mayoría de los ataques.
- El escaneo del código fuente en el entorno de desarrollo no puede ofrecer la misma información que se obtiene al observar una aplicación cuando se ejecuta en producción. Por ejemplo, los análisis estáticos del código fuente no pueden mostrarle qué bibliotecas se cargan realmente, cómo se utilizan, si un proceso está expuesto a Internet o si un proceso interactúa con datos corporativos sensibles.
- Algunas aplicaciones que se ejecutan en producción, como las que se compran a terceros, pueden no haber pasado por su entorno de desarrollo, por lo que nunca tuvieron la oportunidad de ser analizadas por las herramientas de seguridad en el desarrollo.
- A menudo se descubren nuevas vulnerabilidades de día cero después de que una aplicación se haya desplegado en producción. Al implementar el monitoreo continuo de la seguridad de las aplicaciones en producción, puede estar al tanto de estos riesgos.



Construir la resiliencia con la ingeniería del caos

La [ingeniería del caos](#) es una disciplina de desarrollo que somete al software a fallos en un entorno de producción simulado como forma de crear resistencia en los sistemas de software de producción distribuidos. Esta práctica genera confianza en la capacidad del software para soportar circunstancias inesperadas o improbables, como cortes, ralentizaciones, cargas excesivas, etc.

Probar el rendimiento de su aplicación en circunstancias aleatorias y extremas es un ejercicio útil para garantizar que su equipo ofrece sistemas duraderos, fiables y altamente disponibles en cualquier situación. La única forma de hacerlo es en entornos de producción con usuarios reales y niveles de carga reales.



Adoptar un enfoque de plataforma para su flujo de valor de DevOps

No hay escasez de herramientas de DevOps que los equipos de TI puedan utilizar hoy en día para ejecutar diferentes partes del ciclo de vida de DevOps. Pero a medida que su enfoque de DevOps madura y busca escalar DevOps a través de múltiples aplicaciones, la dispersión de la cadena de herramientas se convierte en una realidad. Lo que antes funcionaba bien se convierte en algo manual, engorroso y costoso, y vuelve a un enfoque en silos. Imagínese tener varios equipos tratando de utilizar las mismas herramientas, cada uno para sus propias aplicaciones.

La estandarización de un enfoque de plataforma que proporciona automatización, inteligencia y capacidad de observación sobre los procesos habituales de DevOps ayuda a reducir los gastos generales, reducir el trabajo y mejorar la eficiencia. Un enfoque de plataforma todo en uno crea una única fuente de verdad que elimina los silos, integra las cadenas de herramientas y permite modelos de autoservicio. Este enfoque ayuda a automatizar todo el proceso de desarrollo y proporciona a los desarrolladores y a los equipos de operaciones las herramientas y los datos adecuados para cada etapa del ciclo de DevOps, desde la codificación hasta la entrega y viceversa.

Impulsar el desarrollo futurista hoy

El objetivo de cualquier líder empresarial y tecnológico es facilitar el desarrollo de aplicaciones y servicios. Estas mejores prácticas emergentes incluyen formas eficientes de desarrollar aplicaciones, código y servicios críticos. Además, al aprovechar estas herramientas emergentes, puede ofrecer una arquitectura de desarrollo más proactiva y prescriptiva capaz de satisfacer las demandas digitales actuales.



Los fundamentos de DevOps son un conjunto de procesos y tecnologías que **potencian la automatización, la monitorización y la observabilidad, y las AIOps.**

Estos procesos conducen a una mentalidad de shift-left y shift-right que da como resultado **la entrega continua de versiones de calidad y resistentes.**



CAPÍTULO 5

Beneficios de DevOps

A medida que los equipos adoptan los cambios tanto en la cultura como en los procesos, el enfoque holístico de DevOps para la creación de software e infraestructura puede pagar dividendos, incluso a nivel organizativo. Una vez que los equipos han establecido algunas prácticas recomendadas y métricas clave para supervisar y gestionar, los equipos pueden esperar ver algunos beneficios principales.

Mayor velocidad de entrega

A medida que las empresas se transforman digitalmente, los equipos de desarrollo se ven sometidos a una fuerte presión para crear y entregar software más rápido y con mayor frecuencia sin sacrificar la calidad. Al ser encuestados, el 63% de los profesionales de DevOps informan de que DevOps les permite [lanzar software con más frecuencia](#). Por el contrario, el desarrollo lleva un 41% más de tiempo

en las organizaciones que no utilizan prácticas DevOps. Ese tiempo podría emplearse mejor en la creación de nuevas características o en la implementación de nuevos procesos. ¿La diferencia? DevOps anima a los equipos a desarrollar el código en trozos más pequeños y a democratizar el acceso al código, lo que significa que los desarrolladores trabajan en versiones más pequeñas y frecuentes y con bucles de retroalimentación más estrechos para iterar y lanzar el software más rápidamente.

Los procesos de DevOps rompen los silos y fomentan una mejor colaboración y bucles de retroalimentación entre los equipos. Esta conexión y coordinación interfuncional reduce los plazos de entrega al permitir a los equipos automatizar procesos como el monitoreo, la evaluación de pruebas y la corrección que antes se realizaban manualmente. A través de esta automatización, los equipos pueden desarrollar modelos de autoservicio para acelerar los canales de

Los beneficios de DevOps

Las prácticas DevOps son una inversión cuyos dividendos aumentan con el tiempo y la experiencia. Algunos de estos beneficios incluyen:



Mayor velocidad de entrega gracias a la mejora de los procesos



Mayor calidad de las versiones de software gracias a la mejora de las pruebas



Mejora de la productividad y la colaboración gracias a la capacitación de los desarrolladores



Mejores resultados de negocio gracias a clientes más satisfechos

entrega y escalar los procesos de DevOps en toda la organización.

Aumento de la frecuencia de los lanzamientos

Una filosofía central de DevOps es centrarse en los cambios de código más pequeños, optando por la agilidad y el rápido lanzamiento de características, en lugar de mantener enormes bases de código y realizar lanzamientos monolíticos poco frecuentes. Estos cambios más pequeños son más fáciles de enviar a repositorios de código como GitHub o BitBucket, y más fáciles de probar.

Las publicaciones más frecuentes permiten a los usuarios acceder más rápidamente a nuevas características y funciones. También significan que los desarrolladores reciben más rápidamente los comentarios del mundo real, lo que significa que pueden responder a los problemas y hacer optimizaciones más rápidamente.

Los lanzamientos frecuentes también promueven mejoras continuas en los procesos y flujos de trabajo de DevOps. Algunas empresas, como Google, clasifican a sus equipos en función de su rendimiento de lanzamiento, lo que fomenta una actitud de no conformismo y hace hincapié en los tiempos de respuesta rápidos.

Reducción del riesgo y aumento de la confianza en el lanzamiento mediante un software de mayor calidad

Otro principio central de DevOps es la realización de pruebas frecuentes y automatizadas en todas las etapas del ciclo de vida del desarrollo, que revelan los problemas mucho antes de que lleguen a la producción. A través de prácticas como el cambio a la izquierda, en el que el código se evalúa en función de los SLO de producción, los equipos de desarrollo descubren que el código defectuoso se detiene automáticamente para que no pase a la siguiente fase, lo que ayuda a mejorar la calidad general del software y a reducir las tasas de fallos y defectos.

Los resultados de las pruebas continuas son impresionantes: el 63% de las organizaciones que utilizan prácticas DevOps [han mejorado la calidad de sus despliegues](#), mientras que el 38% informan de una mejora en la calidad del código en general. Los equipos también dedican un 21% menos de tiempo a apagar incendios, lo que deja más tiempo para innovar y mejorar los procesos.

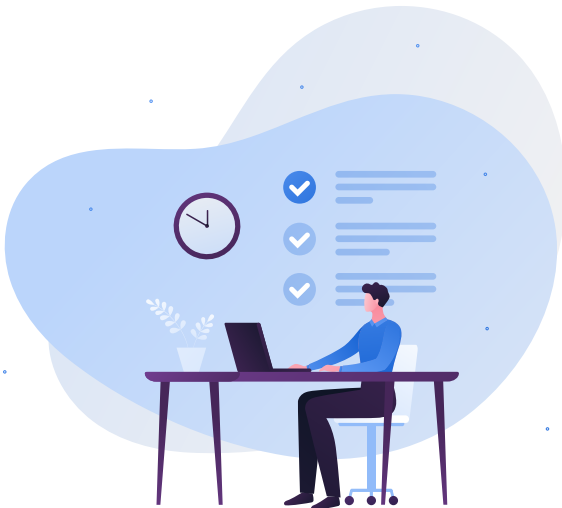
La reducción del riesgo se traduce en un aumento de la confianza en el lanzamiento. Cuando se automatizan de forma fiable más pruebas y procesos, incluidas las decisiones de lanzamiento, utilizando puertas de calidad

basadas en SLO, los equipos pueden lanzar el software sabiendo que está bien probado y que satisface las necesidades de los usuarios.

Potenciación de los desarrolladores

La planificación, la automatización y las pruebas frecuentes que acompañan a las prácticas de DevOps proporcionan a los desarrolladores un rápido ciclo de retroalimentación que puede aumentar su confianza y capacitarlos para trabajar de forma independiente.

Al automatizar las tareas tediosas y las aprobaciones rutinarias siempre que sea posible, los equipos pueden maximizar su eficiencia y trabajar con confianza sabiendo que están siguiendo flujos de trabajo preaprobados. Estos procesos permiten que se desarrolle en los equipos una cultura de propiedad y responsabilidad grupal. Cuando surgen problemas, la cultura DevOps se abstiene de culpar a los errores. En su lugar, se centra en la mejora del proceso,



las lecciones aprendidas y el crecimiento. Esta cultura puede elevar la moral y contribuir a un mejor desarrollo del software.

Mejor colaboración y productividad de los desarrolladores

La cultura DevOps también fomenta un fuerte trabajo en equipo, alejándose de los silos en favor de flujos de trabajo unificados. Como los equipos se enfrentan a fragmentos de código más pequeños y al rápido desarrollo

de características, los equipos pueden participar en una colaboración significativa para garantizar que las piezas del rompecabezas encajen.

DevOps, y por extensión DevSecOps, refuerza los vínculos entre los desarrolladores, los informáticos, los de operaciones y los profesionales de la seguridad de toda la organización, ya que los empleados de cada disciplina son más conscientes de la contribución de cada uno. Con una fuerte iniciativa DevOps, la transparencia y la visibilidad son el estado natural de un proyecto. La recopilación de métricas y un enfoque de todas las manos en la cubierta mantienen a todo el mundo al tanto. Las herramientas de DevOps centralizadas y las plataformas de orquestación permiten a los equipos compartir información y recursos.

Mayor seguridad

Desde su introducción, DevOps se ha extendido más allá de los equipos de desarrollo y operaciones para incluir ahora a los equipos de seguridad, lo que se conoce como DevSecOps. Esta metodología integra a la perfección las pruebas de seguridad y la protección a lo largo del ciclo de vida de desarrollo y despliegue del software. Al igual que DevOps, DevSecOps consiste en adoptar un enfoque colaborativo, y es importante que las organizaciones lo adopten, ya que la velocidad de DevOps puede hacer que las aplicaciones en el ciclo de desarrollo queden vulnerables a los ataques de seguridad.

La adopción de DevSecOps permite a su organización mantener un enfoque de colaboración a través del desarrollo, al tiempo que garantiza que la seguridad no se vea comprometida. Las evaluaciones de seguridad no pueden esperar hasta después del ciclo de desarrollo, sino que deben realizarse simultáneamente. De este modo, los equipos de DevSecOps pueden detectar y responder a los fallos del software en producción de forma más rápida y eficiente. Esto se traduce en un desarrollo, una innovación y una entrega de software más rápidos.

Reducción del tiempo medio de resolución (MTTR)

Los sistemas complejos y multifacéticos inevitablemente experimentan fallos, y los equipos deben prepararse en consecuencia. La reducción del MTTR es una medida del tiempo que se tarda en solucionar estos fallos (normalmente en horas o días). También se puede visualizar como el tiempo medio de inactividad en una serie de fallos.

Naturalmente, los equipos altamente en los principios de DevOps tienen un MTTR más bajo. Esta es una medida precisa tanto de la adhesión a las mejores prácticas como de las habilidades colectivas en general.

En muchos casos, la reducción del MTTR es simplemente una función de la rapidez con la que los equipos pueden localizar e identificar los problemas. Para identificar rápidamente los problemas con precisión, los equipos deben tener plena capacidad de observación sobre su infraestructura en un entorno DevOps. La reducción del MTTR también se basa en monitoreo, el análisis de precisión y la formulación de planes de corrección. Los equipos también dedicarán menos tiempo a las llamadas de soporte, no sólo porque el código es de mayor calidad, sino porque los equipos pueden identificar las causas de raíz más rápidamente.

Mayor fiabilidad y resiliencia del software

Las prácticas DevOps de gestión de bloques de código más pequeños, pruebas continuas y automatización de procesos también animan a los equipos a crear fiabilidad y resistencia en el software desde el principio. Cuando el software se diseña teniendo en cuenta la fiabilidad y la resistencia, es mucho más fácil revertir el código según sea necesario, o incluso eliminar el código activo de la producción cuando surgen problemas sin romper otras características.

Dado que las aplicaciones son conjuntos de funciones básicas individuales (especialmente en la era de las plataformas en la nube y los microservicios), es más fácil desactivar o eliminar las funciones defectuosas sin afectar a todo el paquete. Los equipos de DevOps pueden entonces resolver los problemas y volver a lanzar los servicios cuando

estén listos, manteniendo la fiabilidad del sistema. Varias herramientas hacen esto posible en uno o dos clics. Además, los datos capturados durante el funcionamiento del software permiten a los equipos diseñar mejores pruebas y prevenir futuros problemas.

Clientes más felices

DevOps anima y permite a los equipos ser más receptivos a los comentarios. Los procesos como los rollbacks no quedan relegados a la mera corrección de errores: también son útiles para eliminar las características que no son bien recibidas y mejorarlas creando otro método ágil para satisfacer las demandas de los usuarios. Los usuarios finales también son más propensos a ver el software como "innovador" o "capaz" cuando la entrega de características es continua y sin problemas.

Mejores resultados de negocio

A menudo hablamos de los beneficios que DevOps aporta a los equipos de desarrolladores y operaciones directamente, pero los beneficios de DevOps también se extienden al ámbito de todo el negocio hasta el Clevel. Y Patrick Debois, conocido como el creador del movimiento DevOps, señala que la mayor ventaja de DevOps es la comprensión que proporciona.

DevOps permite que una empresa sea más versátil y esté más orientada a la información para satisfacer las necesidades del cliente y del negocio. Los beneficios que produce la adopción de un enfoque DevOps, a su vez, conducen a una mayor eficiencia, ética de trabajo y, en última instancia, a un mayor beneficio que puede revertirse en el negocio para el crecimiento futuro.

Probar el progreso

Los dividendos de las prácticas DevOps son un beneficio que aumenta con el tiempo y la experiencia. Establecer una práctica DevOps es una inversión que también viene con sus desafíos.



CAPÍTULO 6

Desafíos de DevOps

Si bien los beneficios de DevOps son claros, establecer las herramientas, los procesos y los cambios culturales necesarios para lograr una práctica de DevOps exitosa puede ser un desafío. Al trabajar juntos, los equipos de desarrollo, TI y operaciones pueden eliminar los obstáculos y centrarse en mejorar la forma de crear, desplegar y supervisar continuamente el software.

Aunque el trabajo conjunto es un componente vital, no es el único ingrediente para el éxito. Hacer la transición a DevOps requiere planificación y preparación. A continuación se presentan algunos escollos que hay que evitar cuando las organizaciones planifican e implementan las prácticas de DevOps.

Falta de aceptación por parte de la dirección

Para que una organización moderna logre sus objetivos de

negocio, el liderazgo necesita confiar en su personal técnico y desarrollar una comprensión de los objetivos y puntos de dolor de TI.

Dado que DevOps integra disciplinas a lo largo del ciclo de vida del desarrollo y las operaciones, desde el diseño del producto hasta las llamadas de atención al cliente, no se puede conseguir un DevOps sólido a menos que las organizaciones obtengan la aceptación de los responsables de la toma de decisiones que dirigen la organización o de los departamentos implicados. Las organizaciones más eficientes pueden incluso necesitar el apoyo de toda la cadena: el director general. Otras quizá sólo necesiten un defensor en el nivel superior de la dirección. En cualquier caso, los directivos La dirección que lleva las riendas de la estrategia de compras y productos debe entender lo que se necesita para que los equipos de TI y DevOps trabajen juntos para mejorar la forma en que la organización ofrece

Los retos de DevOps

DevOps se enfrenta a muchos desafíos sin una estrategia y disciplina adecuadas dentro de su organización. Estos pueden incluir:



Falta de aceptación por parte del liderazgo



Mala estrategia de observabilidad



Procesos manuales



No identificar / obtener las métricas adecuadas



Vulnerabilidades de seguridad

sus bienes y servicios a los clientes y promover los cambios culturales necesarios para que el DevOps interorganizativo tenga éxito.

No tener una estrategia de observabilidad

La observabilidad no es lo mismo que la monitorización. En un escenario de monitorización, los cuadros de mando preconfigurados están pensados para alertar a los equipos de los problemas de rendimiento que esperan ver más adelante. Sin embargo, estos cuadros de mando se basan en un supuesto clave: que los equipos pueden predecir qué tipo de problemas encontrarán antes de que se produzcan.

La observabilidad se basa en los resultados de un sistema y permite a los equipos comprender exactamente qué es lo que va lento o se rompe. Con una observabilidad adecuada en las aplicaciones y plataformas nativas de la nube, los equipos de desarrollo pueden aprovechar los datos de telemetría para obtener más información sobre las aplicaciones y los sistemas, automatizar más procesos y publicar código de mayor calidad más rápidamente. Conseguir la observabilidad de extremo a extremo en un entorno de software requiere una combinación de consideración cuidadosa y tecnología potente, y es una parte crítica para garantizar la escalabilidad y el éxito de DevOps.

No automatizar TODOS los procesos manuales

Un objetivo central de DevOps es automatizar tantos procesos y puntos de decisión como sea posible para mejorar el rendimiento y la calidad del software. En este caso, los equipos deben automatizar las pruebas, pero también los flujos de trabajo, como el avance del software desde las pruebas hasta la publicación o la confirmación del código en un repositorio.

Los equipos también pueden depender demasiado del conocimiento tribal que vive en la cabeza de unos pocos, lo que requiere aprobaciones manuales que crean cuellos de botella para la automatización.

Como el número y el tipo de tecnologías que abarcan los entornos DevOps crecen y fluctúan constantemente, el autoescalado basado en la demanda se convierte en un imperativo. La automatización de ciertos tipos de respuestas, como las alertas o la corrección automática de problemas de rendimiento, es otra capacidad clave.

No llevar AIOps a DevOps

La IA puede considerarse un taburete de tres patas, junto con la observabilidad y la automatización. La toma de decisiones inteligente con [IA basada en la causalidad](#) ayuda a los equipos de desarrollo a comprender la causa raíz y a señalar con precisión dónde se producen los errores y qué ha

provocado el fallo de una aplicación.

Pero la IA no es sólo para las medidas reactivas, la IA ayuda a predecir posibles violaciones de los SLO o fallos de la aplicación antes de Incluso lleguen a la producción, lo que permite a los equipos remediar y abordar rápidamente cualquier problema antes de que afecte a los usuarios. Aprovechar la IA en todo el proceso ayuda a los departamentos de TI a escalar sus DevOps para incluir miles de aplicaciones y microservicios para analizar millones y miles de millones de dependencias.

No adoptar un enfoque de autoservicio

Una parte importante de la transformación de TI es la capacidad de dar a los tecnólogos las herramientas que necesitan para tener éxito en el día a día. Dentro del desarrollo de aplicaciones y servicios, es importante considerar el autoservicio como una forma de reducir los tiempos de espera, desplegar nuevas características, acortar los bucles de retroalimentación entre los diferentes equipos, alinear las herramientas y mejorar los conductos de CI/CD. El autoservicio se ignora con demasiada frecuencia hasta el final de un esfuerzo de transformación o se deja de lado por completo. Para unificar los equipos de DevOps, las organizaciones deben construir un plan de autoservicio en la estrategia. Establecer la automatización y la observabilidad en su estrategia de DevOps hace que la creación de modelos de autoservicio y el escalado de DevOps sean sencillos.

No pensar en la seguridad al diseñar los procesos

Muchas organizaciones tratan la seguridad como una experiencia separada que se aplica después de desarrollar el código. Pero cualquier iniciativa DevOps debería tener un plan para integrar la seguridad lo más estrechamente posible. Lo ideal sería que el equipo de seguridad fuera un socio de pleno derecho en el ciclo de vida del desarrollo de software, en pie de igualdad con el desarrollo y las operaciones. Este es el significado de DevSecOps. Al desplazar la seguridad hacia la izquierda e incorporarla al producto en cada etapa del proceso de desarrollo y entrega, los equipos hacen que las aplicaciones y los servicios sean más resistentes contra un mayor número de amenazas ahora y en el futuro. DevSecOps garantiza la visibilidad de

la vulnerabilidad del código, el mapeo de dependencias, las revisiones seguras del SDLC, un profundo conocimiento de cómo tolera un objetivo un ataque real y hasta dónde puede llegar un atacante. No incluir la seguridad en DevOps -o al menos no crear una hoja de ruta para su inclusión en el futuro- es un paso en falso crítico que sacrificará innumerables conocimientos importantes a lo largo de la vida del producto.

No medir las métricas correctas

Si los equipos están revisando los registros de errores o tratando de reconstruir un problema de forma reactiva, es posible que no estén observando las métricas correctas. Identificar lo que las organizaciones necesitan medir es clave para obtener información valiosa de los datos que producen los sistemas.

Aunque hay ciertos puntos de referencia que todas las organizaciones deberían vigilar, como el rendimiento y la latencia, los tiempos de respuesta medios, el tiempo de espera, los errores y el impacto en la memoria, cada organización tendrá métricas para los sistemas y procesos que son únicas. Comience con un conjunto de métricas clave que evalúen la calidad del código y la eficacia de las pruebas, así como la eficiencia del flujo de trabajo y los tiempos de respuesta a los incidentes. Para garantizar la aceptación por parte de la dirección, los equipos deben entender los puntos de dolor clave que preocupan a la organización en general. Dado que las herramientas y los procesos de DevOps afectan a una huella digital tan amplia, ofrecen a las organizaciones la oportunidad de descubrir, medir y mejorar las tendencias clave en toda la organización.

Un esfuerzo de equipo

Alinear equipos de diferentes disciplinas nunca es una tarea fácil. Sin embargo, los líderes del espacio tecnológico ven cómo se difumina la línea entre TI, operaciones y desarrollo, ya que los equipos dependen unos de otros para garantizar el éxito del negocio. En un ecosistema saludable en el que las operaciones de TI y de desarrollo trabajan juntas para lograr un objetivo común, las organizaciones pueden impulsar verdaderos esfuerzos de transformación de TI con pleno apoyo al personal y las iniciativas de DevOps.



CAPÍTULO 7

Métricas de DevOps

En el corazón de todas las prácticas exitosas de DevOps y SRE están las métricas. La telemetría de cada etapa del flujo de trabajo de DevOps -desde el desarrollo y las pruebas hasta el despliegue y las operaciones- proporciona pistas críticas sobre el rendimiento de su software y la eficiencia y eficacia de sus procesos de DevOps.

Se necesitan datos fiables y medibles para automatizar las pruebas, las confirmaciones y los lanzamientos. Para establecer las mejores prácticas, las organizaciones pueden empezar con el proyecto que el equipo de Investigación y Evaluación de DevOps (DORA) de Google estableció, conocido como "Las cuatro claves", que define las cuatro métricas básicas que indican el rendimiento de un equipo de DevOps. Entonces, ¿cuáles son las cuatro claves de DORA, y qué otras métricas clave pueden seguir las organizaciones para mejorar sus prácticas de DevOps y SRE?

Frecuencia de despliegue

La frecuencia de despliegue mide la frecuencia con la que una organización libera con éxito a la producción.

DevOps y la integración continua/entrega continua (CI/CD) van de la mano. Los equipos trabajan ahora con bloques de código más pequeños, y las pruebas y la validación continuas permiten realizar confirmaciones más rápidas. Este ritmo de desarrollo significa que los equipos pueden publicar con más frecuencia, a menudo varias versiones al día.

Una alta frecuencia de despliegue es fundamental para responder a la demanda de los clientes. Cuanto más rápido puedan las organizaciones ofrecer correcciones de errores, mejoras y nuevas funciones, más rápido podrán los desarrolladores recibir valiosos comentarios del mundo

Cuatro métricas clave

DevOps se basa en datos y su éxito depende de métricas fiables.

Según [DORA](#) estas incluyen (pero no se limitan a):



Frecuencia de despliegue



Tiempo de espera para los cambios



Tasa de fracaso de los cambios



Tiempo medio de resolución (MTTR)

real y los usuarios podrán obtener un valor que refuerce su marca.

La frecuencia de despliegue es una métrica tanto a largo como a corto plazo. Por ejemplo, se puede medir el número de commits de código que se envían diariamente o semanalmente en respuesta a los cambios de proceso, tal vez como indicador de eficiencia. Durante períodos más largos, los equipos pueden hacer un seguimiento de si sus números de despliegue están aumentando con el tiempo. Los calendarios de despliegue lento pueden indicar cuellos de botella o retrasos en el servicio que requieren atención.

Tiempo de espera para los cambios

El tiempo de espera mide la cantidad de tiempo que tarda el código comprometido en llegar a la producción.

El tiempo de espera entra en juego cuando se responde a problemas específicos relacionados con la aplicación e indica la rapidez con la que su equipo puede remediar un error o un fallo de las herramientas. Al igual que la frecuencia de desarrollo, el plazo de entrega de los cambios ayuda a los equipos a comprender la eficacia de sus procesos.

El tiempo de espera es fácil de promediar y cuantificar, lo que lo convierte en una métrica accesible para todos los interesados en la aplicación.

Sin embargo, el tiempo de espera no es 100% blanco o negro. Aunque los plazos de entrega más largos pueden indicar

problemas, también pueden ser el resultado de que un equipo se centre en proyectos complejos. Estos esfuerzos, naturalmente, llevarán más tiempo. Es importante investigar el contexto detrás de las cifras de los plazos de entrega y evaluarlas en consecuencia. Mientras que las organizaciones promedio pueden tener tiempos de espera que van de una semana a un mes, algunos equipos de DevOps pueden impulsar cambios de producción en menos de 24 horas.

Dos formas importantes de mejorar el tiempo de entrega de los cambios es implementar pruebas de control de calidad en múltiples entornos de desarrollo y automatizar las pruebas y los procesos de DevOps.

Tasa de fracaso de los cambios

La tasa de fracaso de los cambios mide el porcentaje de despliegues que dan lugar a un fallo en la producción que requiere una corrección de errores o roll-backs

Los equipos pueden comprometer y desplegar los cambios con una alta frecuencia de despliegue y tasas de tiempo de espera para los cambios, pero esos esfuerzos se ven disminuidos si los problemas se cuelan en la producción. Un alto índice de fallos en los cambios (superior al 40%) puede indicar que los procedimientos de prueba son deficientes, y obliga a los equipos a realizar pequeños cambios con frecuencia, lo que erosiona la eficiencia.

El objetivo de medir la tasa de fracaso de los cambios

es evolucionar hacia un proceso DevOps totalmente automatizado. Al automatizar las pruebas y los procesos, el software liberado es más consistente y fiable, y es más probable que tenga éxito en la producción.

El cálculo de la tasa de fracaso de los cambios requiere la capacidad de contar el total de despliegues y vincularlos a incidentes notables. Una solución automatizada de AIOps puede encontrar estos incidentes en los informes de GitHub u otros repositorios de código, y en las alertas del sistema de monitorización y los tickets de los usuarios. Dado que los procesos y sistemas de todas las organizaciones son únicos, la forma exacta de medir la tasa de fracaso de los cambios puede variar mucho.



Tiempo medio de resolución (MTTR)

El tiempo medio de resolución mide el tiempo que tarda una organización en recuperarse de un fallo en producción.

Los usuarios dependen de la disponibilidad de las funciones, y el 99,99+% de tiempo de actividad es el objetivo codiciado. El tiempo de resolución es esencial para garantizar que los equipos se recuperan de las interrupciones no planificadas o de las deficiencias del servicio de forma inmediata y tan eficaz como sea posible, para que las organizaciones no provoquen la frustración de los usuarios y la pérdida de ingresos.

Para determinar el MTTR, puede evaluar el tiempo transcurrido entre el momento en que se produjo un incidente y el momento en que se resolvió. ¿Qué despliegue resolvió esta incidencia? La observación de los datos de despliegue y de la experiencia del usuario es clave para saber si el servicio se ha restaurado de forma eficaz. Los tiempos de restauración prolongados pueden apuntar a una mala alerta o a una mala supervisión y pueden dar lugar a un mayor número de sistemas afectados.

Una de las mejores prácticas para conseguir un MTTR rápido es desplegar el software en pequeños incrementos para reducir el riesgo y desplegar soluciones de monitorización automatizadas para adelantarse a los fallos. El MTTR es otra métrica que varía mucho entre sistemas.

Más allá del DORA: otras métricas

importantes que hay que controlar

Aunque las cuatro claves de DORA son una base esencial, hay una miríada de métricas que puede utilizar para hacer un seguimiento de la eficacia de sus procesos de DevOps y SRE. Aquí hay seis métricas más que puede rastrear para evaluar holísticamente la efectividad de su tubería.

1. Tasa de escape de defectos

También conocida como velocidad de escape de defectos, esta métrica mide la tasa de problemas que “escapan” a la detección durante el desarrollo y se descubren en producción. A continuación, se puede calcular la tasa de defectos por periodo de tiempo, por versión o por despliegue. Los índices de escape más elevados pueden indicar problemas de pruebas y deficiencias de automatización asociadas. Es totalmente posible que las herramientas sean defectuosas cuando los defectos se propagan.

2. Tiempo medio de detección (MTTD)

El MTTD mide la rapidez con la que los equipos descubren los problemas en promedio. Los fallos generalizados del sistema, las vulnerabilidades y las interrupciones pueden causar estragos en las aplicaciones cuanto más tiempo persistan. Identificar y resolver estos problemas es fundamental para reducir el impacto general de un problema en sus aplicaciones, infraestructura y usuarios.

La supervisión eficaz es un principio central de DevOps.

Conseguir un MTTD bajo requiere que los equipos implementen una monitorización eficaz, alertas y observabilidad de extremo a extremo para detectar inmediatamente una anomalía o una degradación del servicio. El origen de un problema puede ser obvio, como una interrupción del servicio central, pero también puede ser más difícil de detectar, como un fallo del back-end, un problema con una herramienta de código abierto o un fragmento de código defectuoso. La detección de este tipo de problemas requiere una monitorización de toda la pila con visibilidad a nivel de código.

3. Porcentaje de código cubierto por las pruebas automatizadas

Otro objetivo central de DevOps es la automatización siempre que sea posible. En consecuencia, las pruebas automatizadas son esenciales para detectar más errores de sintaxis, seguridad y compatibilidad en las compilaciones.

Para lograrlo, las organizaciones pueden implementar entornos de prueba que simulen automáticamente cómo se comporta el código en diversas circunstancias. Aumentar el porcentaje global de código que pasa por las pruebas y la validación automatizadas hace que las pruebas sean más rápidas y sencillas, lo que acelera el pipeline de DevOps, y acorta el bucle de retroalimentación desde la detección de problemas hasta su resolución.

4. Disponibilidad de la aplicación

[La disponibilidad de la aplicación](#) es una medida utilizada para evaluar el grado en que una aplicación funciona plenamente y está disponible para satisfacer los requisitos del negocio y del usuario final. Un sistema de alta disponibilidad está diseñado para cumplir el KPI estándar de cinco 9s (99,999%). Garantizar una mayor disponibilidad de las aplicaciones mantiene a los clientes comprometidos y conectados con sus servicios.

Aunque el tiempo de inactividad no siempre se espera, a menudo se planifica como resultado del mantenimiento. La comunicación entre los miembros del equipo de

DevOps y SRE es crucial para resolver fallos imprevistos y garantizar que tanto el frontend como el backend funcionen sin problemas.

5. Uso y tráfico de la aplicación

El uso y el tráfico de la aplicación supervisa el número de usuarios que acceden a su sistema e informa de muchas otras métricas, incluido el tiempo de actividad del sistema. Las estadísticas de uso son útiles para los equipos, ya que no es raro que las actualizaciones de las aplicaciones afecten a la actividad de los usuarios, ya sea buena o mala. Esto puede ocurrir cuando surgen problemas, o cuando las características largamente esperadas finalmente caen, causando un pico de tráfico. Disponer de estas métricas permite a los miembros del equipo de DevOps reaccionar y gestionar estos picos de forma eficaz.

En el otro extremo del espectro, cuando la actividad de los usuarios se ralentiza, esto podría sugerir que un servicio está interrumpido de alguna manera. Aunque los cambios repentinos son mucho más reveladores para los equipos, también es importante evaluar las tendencias a largo plazo para los problemas que se desarrollan con el tiempo.

DevOps se basa en los datos

Dondequiera que se encuentre en su viaje hacia la madurez de DevOps, la capacidad de medir y dar sentido a los datos procedentes de cada etapa de su flujo de trabajo le ayudará a perfeccionar sus estrategias. En todas las áreas, desde el rendimiento, la fiabilidad y la estabilidad de las aplicaciones hasta la eficacia y la eficiencia de DevOps, las métricas son la lente de sus prácticas de DevOps para ayudarle a impulsar la mejora continua. Aprovechar una solución de observabilidad que ofrezca datos y análisis de alta fidelidad en los que pueda confiar para alimentar estas métricas permite a los equipos aumentar la capacidad de respuesta, afinar los procesos y ofrecer un mejor software más rápidamente.



CAPÍTULO 8

DevOps y la observabilidad inteligente permiten la transformación digital

La transformación digital es ahora fundamental para que las empresas alcancen sus objetivos de negocio. Mediante la creación de un bucle de retroalimentación que se refuerza continuamente, DevOps ofrece una manera para que las organizaciones evolucionen la cultura corporativa y potencien la próxima generación de creación, gestión y seguridad del software.

DevOps combina el desarrollo y las operaciones en un marco unificado que rompe los silos y fomenta la colaboración durante todo el ciclo de vida. En este entorno, los SRE pueden implementar operaciones que garanticen la disponibilidad, la latencia, el rendimiento y la resistencia de

los sistemas de software. Asimismo, las prácticas de CI/CD pueden proporcionar un desarrollo, una prueba, una entrega y un despliegue bien alineados y automatizados.

Las prácticas de DevOps eficientes y efectivas también dependen de la supervisión de las métricas clave, como las Cuatro Claves de DORA que miden la frecuencia de despliegue, el tiempo de espera para los cambios, la tasa de fracaso de los cambios, la disponibilidad de la aplicación y el tiempo medio para restaurar el servicio (MTTR), entre otros. Una vez sentadas las bases de DevOps, las organizaciones pueden desarrollar las mejores prácticas, como la automatización, la supervisión y la observabilidad, y AIOps,

que potencian la entrega continua de software a escala.

DevOps no solo mejora los flujos de trabajo, sino que también ofrece beneficios cuantificables de principio a fin, como el aumento de la velocidad de entrega, el aumento de la frecuencia de lanzamiento, la reducción del riesgo y la reducción del MTTR.

Para facilitar las prácticas de DevOps, SRE y CI/CD, la Plataforma de Inteligencia de Software de Dynatrace, impulsada por la IA, se integra perfectamente con la cadena de herramientas de DevOps de una organización y automatiza las tareas a lo largo del ciclo de vida de

DevOps. Con la automatización continua y la determinación precisa de la causa raíz, Dynatrace hace posible que las organizaciones desarrollen el potencial de DevOps y simplifiquen la complejidad de la nube. Al permitir una innovación más rápida, una colaboración más eficiente y la capacidad de integrar la seguridad de las aplicaciones como parte de las soluciones emergentes de DevSecOps, Dynatrace ofrece respuestas precisas para cada fase del ciclo de vida de desarrollo y entrega de software.

¿Está preparado para aprovechar el potencial transformador de la entrega digital? Descubra cómo puede ayudarle Dynatrace.

Conozca más



Acerca de **Dynatrace**

Dynatrace (NYSE: DT) exists to make the world's software work perfectly. Our unified software intelligence platform combines broad and deep observability and continuous runtime application security with the most advanced AIOps to provide answers and intelligent automation from data at enormous scale. This enables innovators to modernize and automate cloud operations, deliver software faster and more securely, and ensure flawless digital experiences. That is why the world's largest organizations trust the Dynatrace® platform to accelerate digital transformation.

Curious to see how you can simplify your cloud and maximize the impact of your digital teams? Let us show you. Sign up for a free [15-day Dynatrace trial](#).

[dynatrace.com/blog](#) [@dynatrace](#)

