

Cloud Computing, DevOps y DevOps Culture

---

# Tema 1. ¿Qué es DevOps?

# Índice

## Esquema

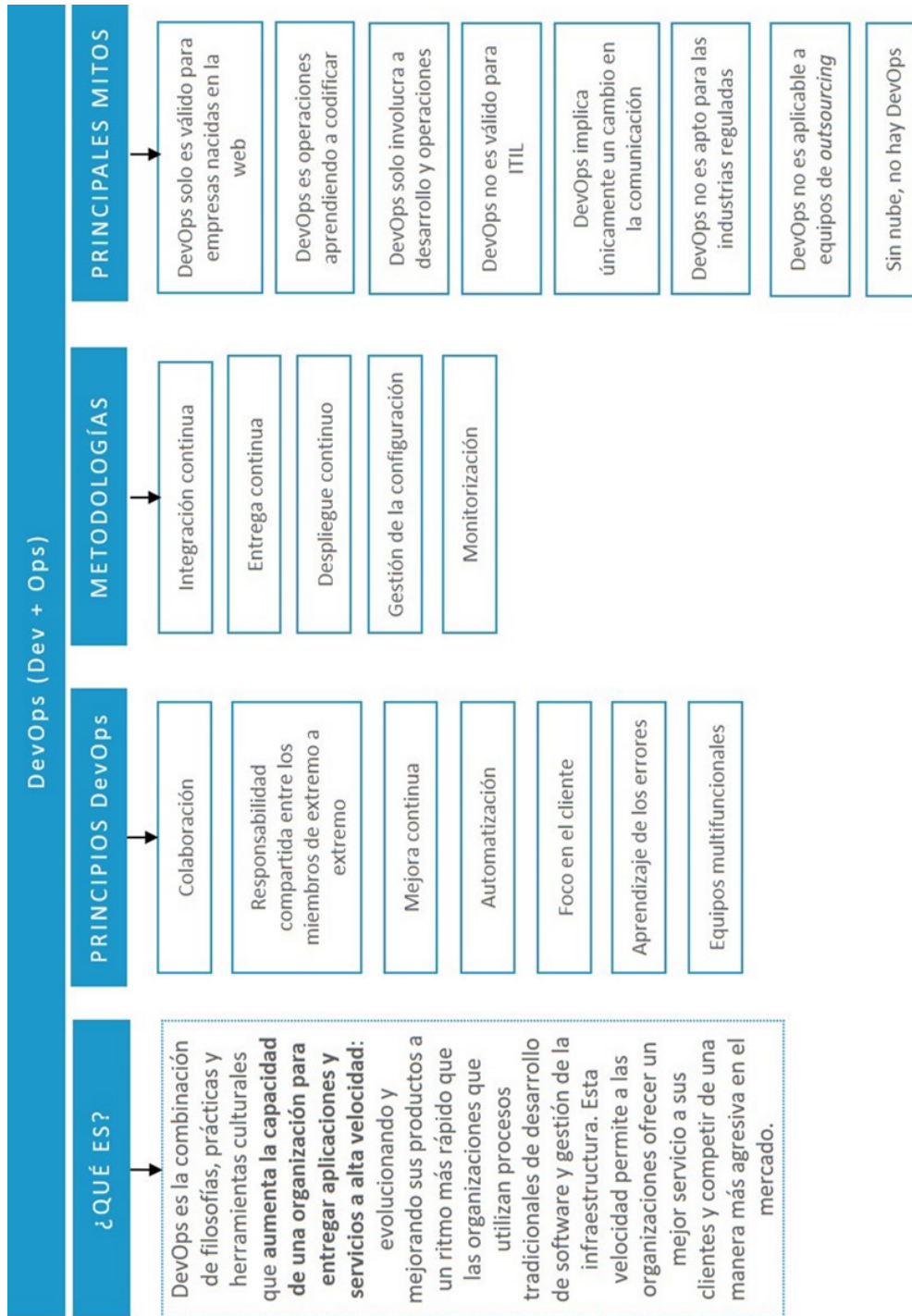
### Ideas clave

- 1.1. Introducción y objetivos
- 1.2. ¿A qué llamamos DevOps?
- 1.3. Principios DevOps
- 1.4. Metodologías y estrategias DevOps
- 1.5. Las capacidades de DevOps
- 1.6. Mitos DevOps
- 1.7. Referencias bibliográficas

### A fondo

- ¿Qué tareas hace un profesional de DevOps?
- What is DevOps?

### Test



## 1.1. Introducción y objetivos

DevOps no puede entenderse únicamente como un conjunto de tecnologías y mejores prácticas, sino que debe entenderse como una revolución cultural. Es una corriente de cambio dentro de las organizaciones en la que se prima la colaboración y la interacción abierta entre los equipos de tecnología de la información (en siglas, TI) y aquellos de desarrollo. El objetivo primordial de esta sinergia es la de alcanzar la máxima eficiencia y agilidad en el desarrollo, despliegue y mantenimiento de las aplicaciones.

Numerosas organizaciones han incluido la virtualización como la mejor forma de optimizar sus recursos computacionales, ya que da nueva vida a infraestructuras obsoletas y obtienen una flexibilidad mucho mayor a la hora de utilizar sus recursos computacionales. La revolución de la virtualización ya no es tal, y el nuevo ciclo de cambio ya está en marcha. Ya no es suficiente hablar de Cloud Computing como una forma de obtener recursos computacionales de manera elástica y en un modelo de pago por uso, sino que además aparece el concepto de *datacenter* virtual, donde podemos tener las ventajas de un *datacenter* con los beneficios de la nube.

A través del estudio de este tema, se pretenden conseguir los siguientes objetivos:

- ▶ Comprender el concepto de DevOps y cómo se aplican las tecnologías y mejores prácticas que se conocen como DevOps.
- ▶ Conocer los factores que intervienen en la implementación del cambio cultural necesario para la adopción de la metodología DevOps en una organización.
- ▶ Analizar los beneficios que reporta DevOps en los proyectos de software.

## 1.2. ¿A qué llamamos DevOps?

**DevOps** es un término relativamente nuevo para describir lo que también ha sido llamado como **administración de sistemas ágiles**. DevOps se centra en el trabajo y la colaboración de los equipos de desarrollo y operaciones.

En la actualidad, las empresas se mueven a la velocidad de la nube, y es por ello por lo que DevOps se ha convertido en un enfoque cada vez más extendido y popular para la entrega de *software*. Los equipos de desarrollo y operaciones utilizan esta metodología para construir, probar, implementar y monitorizar las aplicaciones debido a que les permite: actuar con velocidad, mantener altos índices de calidad y controlar los cambios con suma rapidez.

DevOps es esencial para cualquier empresa que aspire a ser ágil y que pretenda ser capaz de responder rápidamente a las demandas del mercado. Por lo tanto, DevOps es un enfoque:

- ▶ Hacia la entrega de software ágil.
- ▶ Que promueve una colaboración estrecha entre las líneas de negocio, desarrollo y operaciones de TI.
- ▶ Que elimina las barreras entre las partes interesadas y los clientes.

Es necesario que todas las partes involucradas en el proceso de entrega de software estén dispuestas a colaborar entre sí. Los **equipos de desarrollo** necesitan diseñar, desarrollar, entregar y ejecutar el software de la manera más rápida y confiable posible. Los **equipos de operaciones** deben identificar y resolver los problemas de forma temprana mediante la monitorización, la predicción de fallos, la administración de los entornos y la solución de problemas. La combinación de este enfoque común permite monitorizar y analizar los cuellos de botella y optimizar las capacidades de

DevOps, que busca, ante todo, fomentar la colaboración entre el área de negocio y los departamentos de desarrollo y operaciones para entregar y ejecutar el software lo antes posible.

### **DevOps como herramienta de cambio para las organizaciones**

Para ir adentrándonos en la materia, necesitamos comprender la **necesidad** y el **valor** de DevOps a nivel de negocio, así como los **principios** de DevOps. Como sabrán, realizar un cambio en el «Business as usual» siempre es difícil y, por lo general, requiere de una inversión. Así que, cada vez que una organización adopta cualquier nueva tecnología, metodología o enfoque, la adopción de esta tiene que ser impulsada por una **necesidad de negocio**. Es decir que, al desarrollar un modelo de negocio para la adopción de DevOps, se debe entender la necesidad de la empresa, incluyendo los retos a los que se enfrenta.

En un entorno DevOps, los equipos son responsables de ofrecer nuevas características, pero también estabilidad, escalabilidad, fiabilidad y otro sinfín de propiedades comunes a un software de calidad. Es por ello por lo que las responsabilidades se equilibran para garantizar que ambos equipos (desarrollo y operaciones) tengan visibilidad del rendimiento de la aplicación a través de todas las etapas del ciclo de vida.

Debido a los beneficios que reporta a todo el conjunto de la organización, DevOps continúa ganando adeptos año tras año desde su aparición. Según los datos de la popular web DevOps.com, la tasa de adopción aumentó sustancialmente, de 2015 a 2016, del 66 % al 74 %. ¿Qué impulsa a las empresas hacia la adopción de DevOps? La respuesta es muy sencilla: los largos plazos de entrega del software hasta su paso a producción, derivado de los sistemas de trabajo tradicionales, impiden a las empresas brindar servicios de vanguardia y mejorar la experiencia del cliente. Para mantener el ritmo de las demandas del mercado, los equipos de TI deben construir, implementar, probar y lanzar software en ciclos cada vez más rápidos.

Debido a que DevOps mejora la forma en que una empresa entrega valor a sus clientes, proveedores y socios, podemos afirmar que es un proceso esencial de negocio y no solo una capacidad de TI. No debemos olvidar que esta metodología ofrece un retorno significativo de la inversión, en inglés *Return of Investment* (en adelante, ROI), por diversas razones:

### **Aceleración de la innovación**

Con un sólido equipo de operaciones y desarrollo que trabaje acompasadamente, las aplicaciones se pueden desarrollar y desplegar con mayor rapidez. Esto es vital, ya que **el éxito empresarial actual depende, en gran medida, de la capacidad de una organización para innovar más rápido que la competencia**. Los ingenieros DevOps pueden aprovechar las herramientas tecnológicas de las que disponen para comprobar los datos de rendimiento de sus aplicaciones en tiempo real, y así, comprender el impacto de los cambios introducidos en el código. A su vez, las soluciones a los problemas derivados del software se proveen en plazos de tiempo más cortos porque los miembros del equipo solo necesitan verificar los últimos cambios de código para detectar el origen de los errores.

## Mejora de la colaboración

En vez de intentar desdibujar las diferencias entre ambos departamentos (desarrollo y operaciones), un entorno DevOps exitoso procura **construir un puente basado en la comunicación y la colaboración para crear una sinergia**. La cultura de desarrollo de software, entonces, se enfoca en objetivos, responsabilidades y logros compartidos, en lugar de objetivos individuales. Cuando estos equipos adquieren un buen nivel de confianza y colaboración, pueden experimentar, investigar e innovar de manera más efectiva. El entorno de desarrollo se vuelve, progresivamente, más uniforme a medida que todos los miembros del equipo trabajan para alcanzar objetivos compartidos.

## Incremento de la eficiencia

Las herramientas automatizadas y las plataformas de producción estandarizadas son elementos clave de las mejores prácticas de DevOps, que ayudan a que las implementaciones sean más predecibles y liberen al personal de TI de tediosas tareas repetitivas. Con un entorno de pruebas e integración automatizadas, los desarrolladores no necesitan desperdiciar su tiempo confiando en que se completen los procesos de integración de código. Estas herramientas ofrecen oportunidades adicionales para mejorar la eficiencia:

- ▶ La **infraestructura escalable**, como las soluciones basadas en la nube, ayudan a acelerar las pruebas y los procesos de implementación al aumentar el acceso a los recursos de hardware.
- ▶ Las **herramientas de compilación y desarrollo** ayudan a acortar los ciclos de desarrollo y acelerar la entrega del producto.
- ▶ Los **flujos de trabajo de entrega continua** ayudan a entregar software de manera más rápida y frecuente.



## Reducción del número de fallos

Los ciclos de desarrollo más cortos, derivados de DevOps, promueven entregas de código más frecuentes. Con estas implementaciones modulares, los equipos pueden descubrir e identificar, de forma temprana, problemas asociados a la configuración, el código o la infraestructura. **DevOps eleva el grado de compromiso de los miembros del equipo durante todo el ciclo de vida de una aplicación, lo que da como resultado un código de mayor calidad.** Por lo general, se requieren menos cambios y modificaciones porque los desarrolladores identifican y eliminan posibles problemas a medida que escriben el código. Según un informe reciente elaborado por Puppet Labs, llamado [State of DevOps](#), las organizaciones que adoptan una cultura DevOps tienen 60 veces menos fallos que aquellas que no implementan DevOps.

## Aceleración del tiempo de recuperación

Como hemos dicho anteriormente, debido a que las implementaciones de DevOps son más frecuentes y «pequeñas», **los errores son fácilmente detectables y es por ello por lo que las soluciones se encuentran con mayor rapidez.** El equipo necesitará verificar los últimos cambios en el código para poder resolver una incidencia en vez de revisar enormes cantidades de código. Los tiempos de resolución son intrínsecamente más rápidos porque, además, la responsabilidad en la solución de los problemas y las correcciones pertinentes corresponden a un solo equipo: el equipo DevOps. En el informe [State of DevOps](#) de Puppet Labs, se establece que los equipos DevOps de alto rendimiento se recuperan de los fallos 168 veces más rápido que aquellos con menor rendimiento.

## Aumento de la satisfacción laboral

DevOps promueve un ambiente de trabajo basado en el rendimiento, en contraposición con una cultura basada en reglas, estatus o poder. Esto reduce los obstáculos burocráticos y fomenta la responsabilidad compartida entre todos los

miembros. El resultado es un equipo de trabajo más satisfecho y productivo, que ayuda a impulsar el rendimiento del negocio. Los desarrolladores y los ingenieros de operaciones, generalmente, prefieren un entorno DevOps porque pueden trabajar de manera más eficiente y cambiar de roles cuando sea necesario. A su vez, tienen una mejor comprensión sobre cómo impacta su función dentro de TI y dentro de la organización en su conjunto.

La entrega rápida de software es crucial en la era digital actual y la cultura de DevOps es el motor de este proceso. En otras palabras, permite a las empresas acelerar la comercialización de sus servicios y productos al mercado y desplegar nuevas funciones, de manera rápida y eficiente. La adopción de DevOps no es un proceso simple, pero cuando se realiza correctamente, la inversión genera dividendos que duplican el esfuerzo inicial.

## 1.3. Principios DevOps

En el corazón de los principios DevOps se encuentra el **aprendizaje colaborativo** y las **relaciones de colaboración** entre desarrollo y operaciones. Estos se centran en aumentar el ritmo de trabajo de forma planificada para obtener despliegues e implementaciones más frecuentes, al tiempo que mejora la estabilidad, la resistencia y la seguridad del entorno de producción. Para que una organización pueda abrazar los principios de DevOps, debe reforzar y enfatizar este enfoque holístico en todas sus áreas, y no únicamente en los departamentos de desarrollo y operaciones.

Estos principios fundamentales pueden resumirse en la siguiente lista:

### **Fomentar un ambiente de trabajo colaborativo**

Como ya hemos comentado, la esencia de la metodología DevOps es el trabajo conjunto entre los equipos de desarrollo y operaciones, a fin de crear una sinergia que se centre en la consecución de objetivos comunes. Para lograr este objetivo, las organizaciones necesitan fomentar una comunicación eficiente y permitir que ambos equipos compartan ideas y resuelvan problemas de forma conjunta. Esta «rotura de los silos» permite a las empresas alinear a sus trabajadores, procesos y herramientas hacia un enfoque centrado en el cliente.

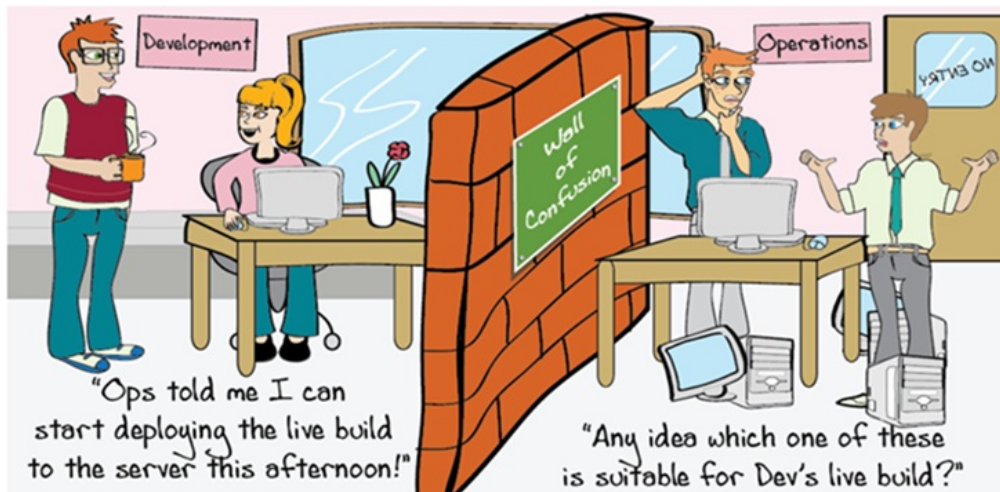


Figura 1. What is DevOps? Fuente: Buehring, 2020.

## Establecer una responsabilidad de extremo a extremo

En el modelo tradicional de desarrollo de software, desarrollo y operaciones tenían roles separados. Pero en un entorno DevOps, ambos grupos trabajan como un equipo que es totalmente responsable de la aplicación, de principio a fin. Tradicionalmente, los desarrolladores escribían código y operaciones desplegaba ese código, pero eso producía todo tipo de ineficiencias, problemas de rendimiento y entornos impredecibles. Todos los miembros del equipo DevOps son responsables de velar por la calidad y fiabilidad de las aplicaciones que desarrollan.

## Fomentar la mejora continua

La responsabilidad de extremo a extremo involucra a todas las áreas de la organización, ya que estas deben adaptarse continuamente a las circunstancias cambiantes (ya sea el surgimiento de nuevas tecnologías, las nuevas y variadas necesidades de los clientes, o los cambios en la legislación). DevOps pone énfasis en la mejora continua para optimizar el rendimiento, los costes y la velocidad de entrega.

## Fomentar la automatización

Para hacer frente a los numerosos ciclos de desarrollo y responder de inmediato a los comentarios de los clientes, las organizaciones deben implementar procesos automatizados. Afortunadamente, en los últimos años han surgido variadas y numerosas herramientas de automatización que agilizan la gestión de procesos en gran medida.

## Dar prioridad a las necesidades del cliente

DevOps requiere que las organizaciones tengan la misma flexibilidad que las *start-up lean*, ya que estas pueden innovar continuamente, cambiar de estrategia cuando sea necesario e invertir en la creación de nuevas características (*features*) para aumentar la satisfacción de sus clientes. Los equipos de DevOps deben estar al día de todo lo que ocurre en el mercado para satisfacer constantemente las necesidades y demandas cambiantes de los consumidores. Los datos recopilados de los procesos automatizados deben revisarse con frecuencia para garantizar que se cumplen los objetivos de rendimiento establecidos por el área de negocio.

## Aceptar el fracaso y aprender de él

La aceptación del fracaso fomenta un clima de aprendizaje que tendrá un **impacto positivo en la cultura organizacional**. Cuando los equipos se sienten psicológicamente seguros para actuar y tienen la capacidad de transformar radicalmente su trabajo, pueden incurrir en fallos. Resulta vital convertir esos fallos en oportunidades de aprendizaje.

## Formar equipos multifuncionales

Es necesario que los equipos de DevOps se involucren en cada etapa del ciclo de vida del desarrollo de software, desde la planificación, la construcción, la implementación, y la retroalimentación hasta la mejora continua. Para ello, es necesario que el equipo sea multifuncional y equilibrado, y que cada miembro posea

un conjunto habilidades idóneas para su rol en TI.

Además de estos principios, existen otros que se centran, más específicamente, en el área de TI y que abogan por una mejora integral de sus procesos. Sin embargo, tienen un enfoque holístico para DevOps y organizaciones de todos los tamaños pueden adoptarlos. Estos principios son:

- ▶ **Trabajar en entornos similares a producción durante el desarrollo del software y la ejecución de pruebas.** Hace alusión al concepto DevOps de «desviación a la izquierda» o *shift left*. El objetivo que persigue este principio es el de permitir que los equipos de desarrollo y calidad trabajen en entornos similares a producción a fin de que puedan verificar el comportamiento y desempeño de la aplicación mucho antes de que esté lista para su despliegue. Desde una perspectiva de operaciones, este principio tiene un enorme valor porque permite que el equipo pueda ver, desde una fase temprana del ciclo, cómo se comportará su entorno cuando soporte a la aplicación.
- ▶ **Realizar despliegues mediante procesos repetibles y fiables.** Permite que el área de operaciones dé soporte a un proceso de desarrollo de software ágil e iterativo, durante todas las fases del ciclo de vida que atraviesa el código. La automatización es esencial para crear procesos que sean iterativos, frecuentes, repetibles y fiables, por lo que la organización debe crear una fuente o suministro de entregas que permita que el despliegue sea continuo, automatizado y validado a través de pruebas. Los despliegues frecuentes también permiten a los equipos poner a prueba sus propios procesos y, a largo plazo, hace que se reduzca el riesgo de fallos.
- ▶ **Supervisar y validar la calidad operativa.** Pone énfasis en la monitorización del código desde una fase temprana, al exigir que el código sea validado a través de pruebas automatizadas al comienzo del desarrollo y con una alta frecuencia, a fin de controlar las características funcionales y no funcionales de la aplicación. Esta supervisión frecuente proporciona una alerta temprana sobre cuestiones operativas o de calidad, que puedan ocurrir más tardíamente en producción.

- ▶ **Amplificar los bucles de retroalimentación.** Exige a las organizaciones la creación de canales de comunicación eficaces que permitan a todas las partes interesadas acceder y participar en un ciclo comunicativo. De ello se desprende que el desarrollo de código tendrá el foco puesto en las prioridades del proyecto y podrá ajustarse a los requisitos del proyecto si estos merman. También implica que los pases a producción se harán mediante la mejora de los entornos de producción y que los planes de entrega se modificarán en pos del negocio.

## 1.4. Metodologías y estrategias DevOps

DevOps tuvo sus orígenes en Agile. Para comprender mejor esta afirmación, es importante recordar que Agile promueve el desarrollo de software a través de iteraciones cortas, la entrega continua de nuevas características, así como correcciones de errores en ciclos rápidos, de entre dos a cuatro semanas. Por su parte, DevOps elimina los silos que separan a los equipos de desarrollo y operaciones con el objetivo de disminuir el tiempo de respuesta a sus clientes, y garantizar la entrega continua de software.

DevOps no solo involucra a todas las áreas de una organización en el proceso de desarrollo (líneas de negocio, proveedores involucrados en la entrega de software y propios consumidores), sino que además lo hace de una manera en la que acelera el desarrollo y mejora la calidad de todos sus procesos. Esto conduce a la creación de una cultura de la innovación, que permite a las organizaciones colaborar y reaccionar con agilidad ante los cambios del mercado.

Las metodologías de DevOps incluyen:

- ▶ Integración continua, que incluye a la creación de código, la construcción, la integración y las pruebas.
- ▶ Entrega continua, que incluye a la integración continua, pero se centra en las entregas.
- ▶ Despliegue continuo, que se centra en automatizar las entregas de la forma más rápida posible.
- ▶ Gestión de la configuración y monitorización continua.



## Estrategia DevOps

Como hemos visto, DevOps se basa en el concepto de continuidad. No es una casualidad que hayamos mencionado el concepto de integración, entrega y despliegue continuos. La estrategia de DevOps se centra en la capacidad empresarial para la entrega continua de software, que permite a los clientes aprovechar las oportunidades del mercado y, a la vez, reducir el tiempo de retroalimentación de estos.

Permite a los equipos trabajar de forma conjunta y esforzarse en solucionar problemas que puedan aparecer en cualquier parte del ciclo de vida del desarrollo de software. El cliente representa la prioridad la máxima de estos equipos y su bienestar es una responsabilidad compartida entre todos.

La estrategia DevOps acelera la entrega de software, ayuda a equilibrar la velocidad, los costes, la calidad y el riesgo, y aporta una mayor capacidad para innovar. Además, reduce el tiempo de retroalimentación de los clientes, lo que ayuda a mejorar su experiencia.

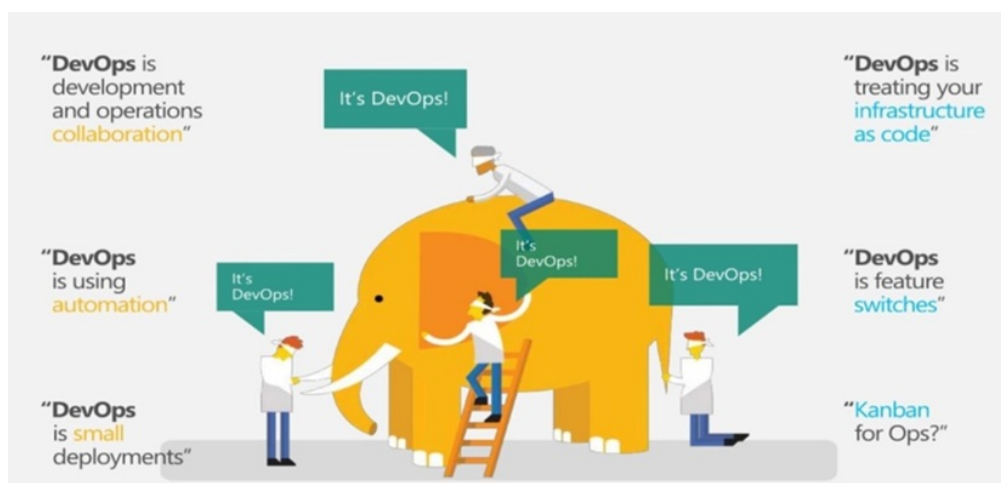


Figura 2. Cómo aplicar la cultura DevOps a tu empresa. Fuente: Bravent IT Consulting, 2019.

## 1.5. Las capacidades de DevOps

Las capacidades que conforman DevOps son un conjunto amplio que abarca todo el ciclo de vida de entrega del software. Cuando una organización comienza a trabajar con DevOps depende de sus objetivos y metas de negocio (cuáles son los desafíos que está intentando abordar y cuáles son los puntos de mejora en sus capacidades de entrega de software). DevOps tiene una arquitectura de referencia, es decir, un patrón o un conjunto de métodos y capacidades determinadas.

Estas capacidades son proporcionadas por un conjunto idóneo de personas, prácticas definidas y herramientas de automatización. La arquitectura de referencia de DevOps plantea los siguientes senderos o caminos de adopción:

- ▶ Planificación y medición.
- ▶ Desarrollo y pruebas.
- ▶ Lanzamientos y despliegues continuos.
- ▶ Supervisión y optimización.

### Planificación y medición

Este camino consiste en una práctica que se centra en las líneas de negocio y sus procesos de planificación: la planificación continua del negocio. Como ya hemos mencionado, las empresas tienen que ser ágiles y deben ser capaces de reaccionar rápidamente ante las necesidades y comentarios de los usuarios. En consecuencia, muchas empresas, hoy en día, emplean **técnicas de pensamiento Lean**. Estas técnicas comienzan a dar forma a sus proyectos a través de la identificación de los resultados deseados y los recursos necesarios para cumplir con sus metas de negocio y, luego, en base a la retroalimentación que reciben de sus clientes se adaptan y ajustan, según sea conveniente. En otras palabras, las organizaciones

modifican el curso de sus planes de negocio, lo que les permite tomar decisiones *trade-off* continuas en un entorno de recursos limitados.

### Desarrollo y pruebas

Este camino implica la adopción de dos prácticas: desarrollo colaborativo y pruebas continuas. Como tal, forma el núcleo de las capacidades de desarrollo y de garantía de calidad (en siglas, QA).

La entrega de software en una empresa involucra a una gran cantidad de equipos multifuncionales, que incluye a: los propietarios de las líneas de negocio, los analistas de negocios, arquitectos de software, desarrolladores, profesionales de control de calidad, el personal de operaciones, especialistas en seguridad, proveedores y socios. Los profesionales de estos equipos trabajan en múltiples plataformas y pueden estar dispersos en múltiples localizaciones. El desarrollo colaborativo permite que estos profesionales trabajen juntos y proporciona, así, un conjunto común de prácticas y una plataforma común que se puede utilizar para crear y desplegar software.

Una capacidad básica, incluida dentro del desarrollo de colaborativo, es la integración continua (una práctica en la que los desarrolladores de software integran, de forma frecuente, su trabajo con el de otros miembros del equipo de desarrollo). La integración continua se hizo popular gracias al **movimiento ágil (o Agile)**. Esta filosofía de trabajo tiene como objetivo que los desarrolladores integren regularmente su trabajo con el del resto de los desarrolladores de su equipo y luego ejecuten pruebas para comprobar que dicha integración ha sido exitosa.

En el caso de los sistemas complejos integrados por múltiples sistemas o servicios, los desarrolladores también integran regularmente su trabajo con otros sistemas y servicios. Esta integración, llevada a cabo con regularidad, es fundamental para la detección temprana de fallos y posibles riesgos derivados. Es necesario tener presente que, en los sistemas complejos, los principales riesgos se asocian a

problemas técnicos y, al tiempo, a una mala gestión en la planificación de las entregas.

Las pruebas sobre el código adquieren una especial relevancia debido a la práctica de la integración continua. Esta persigue varios objetivos:

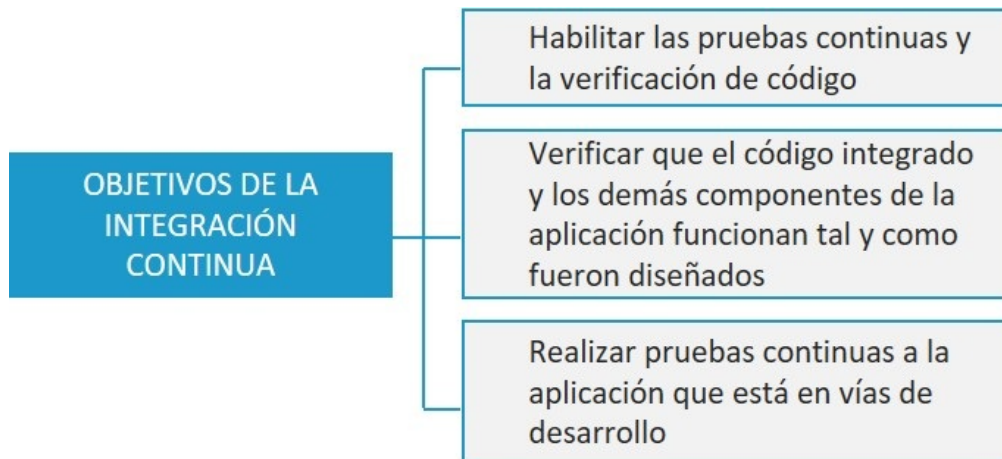


Figura 3. Los objetivos de la integración continua. Fuente: elaboración propia.

Quando hablamos de pruebas continuas, nos referimos a comenzar con la fase de pruebas desde el inicio del desarrollo y, de forma recurrente, a través del ciclo de vida de la aplicación. Esto da como resultado una reducción de costes y el acortamiento de los ciclos de *testing*, pero también sirve como retroalimentación de la calidad operativa. Estos objetivos son alcanzables mediante la implementación de pruebas automatizadas y la virtualización de servicios. Como ya hemos mencionado en otro apartado, el trabajo en entornos similares a producción hace factible la realización de pruebas continuas a nuestro código.

### Lanzamientos y despliegues continuos

**La entrega y el despliegue continuos** llevan el concepto de integración continua a otro nivel. La entrega continua permite a los desarrolladores automatizar las pruebas unitarias y verificar actualizaciones en las aplicaciones, antes de enviarlas a los clientes. Estas pruebas pueden ser sobre la interfaz de usuario o *user interface* (en

siglas, UI) de carga, de integración, de fiabilidad de la *Application Programming Interfaces* (en siglas, API), etc. De este modo, los desarrolladores pueden validar las actualizaciones de forma más exhaustiva y descubrir problemas por anticipado. Con la nube, resulta sencillo y rentable automatizar la creación y replicación de varios entornos de pruebas, algo que anteriormente era complicado en las instalaciones.

Una definición muy acertada es la que da Jez Humble (2010) en el libro *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*:

«La entrega continua es la habilidad de facilitar cambios de todo tipo (incluyendo nuevas características, cambios de configuración, soluciones de bugs y experimentos) a producción, o a los usuarios, de forma rápida, segura y sostenible».

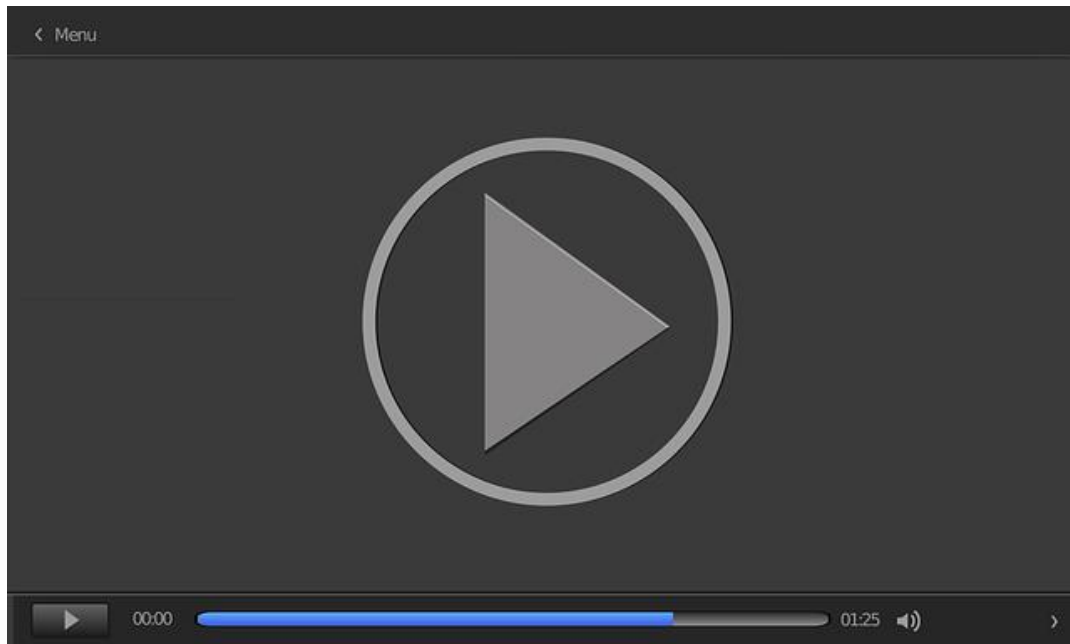
### **Supervisión y optimización**

La ruta de adopción, a través de la supervisión y optimización, permite a las empresas controlar cómo las aplicaciones están comportándose en producción y recibir retroalimentación de los clientes. La supervisión continua proporciona información y métricas sobre las aplicaciones durante las diferentes etapas del ciclo de entrega del software a los departamentos de operaciones, QA, desarrollo, el personal de las líneas de negocio y otras partes interesadas.

Las nuevas tecnologías permiten a las empresas capturar el comportamiento y los problemas que experimentan los clientes que utilizan la aplicación. Esta retroalimentación permite que las diferentes partes interesadas (accionistas o jefes de proyecto, por ejemplo) tomen medidas para mejorar las aplicaciones y la experiencia del cliente. Con esta información, las líneas de negocio podrán ajustar su estrategia, el departamento de desarrollo podrá ajustar las características que entrega y el departamento de operaciones podrá mejorar el entorno en el que se despliega la aplicación.

## Madurez en el desarrollo de software

Antes de continuar con el siguiente apartado, te invito a que veas la píldora *Modelos de madurez* en el desarrollo del software, donde analizaremos los diferentes niveles de madurez que han sido definidos por la industria y su conexión con DevOps.



---

Accede al vídeo: <https://unir.cloud.panopto.eu/Panopto/Pages/Embed.aspx?id=639043f1-f3c8-4cc9-a1e0-ad8d0104e260>

---

## 1.6. Mitos DevOps

El movimiento DevOps es joven y todavía emergente, especialmente en las organizaciones como conjunto. Al igual que cualquier nuevo movimiento o tendencia, este ha despertado mitos y falacias. Lo cierto es que algunos de estos mitos se han originado en empresas que probaron, sin éxito, la adopción de DevOps. A continuación, nombraremos algunos de ellos.

### **La metodología DevOps solo es adecuada para las organizaciones nacidas en la web**

Lo que generalmente se conoce como DevOps se originó en las empresas nacidas en la web, es decir, aquellas empresas que se originaron en Internet como Etsy, Netflix y Flickr. Sin embargo, empresas de gran renombre han estado implementando los principios y prácticas alineadas con DevOps para realizar entregas de software durante décadas. Por otra parte, **los principios DevOps actuales tienen un nivel de madurez que los hace aplicables a todo tipo de organizaciones que tengan tecnologías multiplataforma y equipos distribuidos.**

### **DevOps es operaciones aprendiendo a codificar**

Los equipos de operaciones siempre han gestionado los entornos y tareas repetitivas a través de *scripts*, pero con la evolución de la infraestructura como código, vieron la necesidad de gestionar estas grandes cantidades de código con prácticas de ingeniería del software como el control de código de versiones, o *el check-in y check out*, por ejemplo. Hoy en día, una nueva versión del entorno se lleva a cabo mediante una nueva versión del código que lo define. Esto no significa, sin embargo, que los equipos de operaciones necesiten aprender a codificar en Java o C#. La mayoría de las tecnologías de infraestructura como código utilizan idiomas como Ruby, que es relativamente fácil, especialmente para las personas que tienen experiencia con secuencias de comandos.

### **DevOps solo involucra a las áreas de desarrollo y operaciones**

Aunque este mito sugiere que DevOps únicamente involucra o afecta a los equipos de desarrollo y operaciones, como hemos visto, **la adopción de DevOps requiere que toda la organización se comprometa en la consecución de los objetivos.**

Todas las partes interesadas en la entrega software —líneas de negocio, profesionales, ejecutivos, socios, proveedores, etc.— deben participar y colaborar en un esfuerzo conjunto.

### **La metodología DevOps no es adecuada para organizaciones ITIL**

Algunas personas temen que las capacidades de DevOps, tales como la entrega continua, sean incompatibles con los controles y procesos prescritos por la *Information Technology Infrastructure Library* (en lo sucesivo denominado ITIL, por sus siglas en inglés), un conjunto documentado de prácticas recomendadas (del inglés *best practices*) para los servicios de gestión de TI. En realidad, **el modelo del ciclo de vida ITIL es compatible con DevOps y lo mismo ocurre con sus principios (se alinean muy bien con los de DevOps).** La ITIL, sin embargo, ha tenido una mala reputación en algunas organizaciones debido a que se ha implementado, en su mayor parte, en las tradicionales metodologías en cascada que, como sabemos, se caracterizan por tener procesos lentos que no admiten cambios ni mejoras rápidas.



## La metodología DevOps no es apta para industrias reguladas

Las industrias reguladas necesitan contar con procedimientos y prácticas fiables, comprobaciones y balances que aseguren el cumplimiento de las normas y sean susceptibles de superar auditorías con éxito. **La adopción de DevOps mejora el cumplimiento de estos requisitos, si se hace correctamente.** La automatización del flujo de procesos y el uso de herramientas de software, que tienen la capacidad de monitorizar y hacer seguimiento de la información, pueden ser un gran aliado a la hora de recoger los datos o paquetes de datos necesarios requeridos en una auditoría.

## DevOps no es compatible con equipos de desarrollo externalizado

Los equipos de outsourcing deben ser vistos como proveedores o prestadores de una capacidad en el canal de entrega de DevOps. Es por ello por lo que **las organizaciones deben garantizar que las prácticas y procesos de estos equipos proveedores sean compatibles con los de sus propios equipos internos.** La planificación de entregas conjuntas, la gestión de los elementos de trabajo y las herramientas adecuadas mejoran significativamente la comunicación y la colaboración entre las líneas de negocio, los proveedores y los equipos de proyectos, abriendo paso a la implementación de las prácticas DevOps. El uso de herramientas de gestión de entrega de aplicaciones puede mejorar significativamente la capacidad de una organización para definir y coordinar todo el proceso de entrega a través de todos los participantes.

## Si no hay nube, no hay DevOps

Cuando se piensa en DevOps, a menudo se lo asocia a la nube por su capacidad de proveer dinámicamente recursos de infraestructura para desarrolladores y *testers* (trabajadores dedicados a la realización de pruebas) a fin de que obtengan rápidamente entornos de prueba sin tener que esperar días o semanas. Sin embargo, **la nube no es un requisito necesario para adoptar las prácticas**

**DevOps**, siempre y cuando la organización tenga procesos eficientes para la obtención de recursos, despliegues y pruebas. La virtualización en sí misma es opcional.

### **DevOps no es aplicable a sistemas grandes y complejos**

Los sistemas complejos requieren del mismo nivel de disciplina y colaboración que ofrece DevOps, ya que dichos sistemas suelen tener varios componentes de software y, cada uno de ellos, a su vez, tiene sus propios ciclos y plazos de entrega.

**DevOps facilita la coordinación de estos ciclos y planificación de entregas, a nivel de sistema.**

### **DevOps es un cambio en la comunicación**

Algunos miembros de la comunidad DevOps han acuñado términos graciosos como ChatOps o HugOps, dando a entender que DevOps solo se centra en la colaboración y la comunicación. Como hemos visto, **la idea de que la comunicación y la colaboración resuelven todos los problemas es errónea**. Si bien DevOps se apoya en la comunicación, si esta viene acompañada de procesos desorganizados, equipos disfuncionales, o malas inversiones, no conduce a mejores resultados ni a la obtención de beneficios.

### **DevOps implica realizar cambios y despliegues a diario**

Esta idea errónea proviene de las organizaciones que solo despliegan aplicaciones en la web. Algunas de estas empresas afirman con orgullo, en sus sitios web, que realizan despliegues en producción a diario. Esta práctica, sin embargo, no resulta en absoluto provechosa en organizaciones de gran tamaño que deseen desplegar aplicaciones complejas, sino que incluso también puede resultar imposible debido a ciertas restricciones regulatorias o a la política interna de la empresa. **La adopción de DevOps permite a las organizaciones hacer un *push* a producción cuando sea justificado y necesario, y no se rige por una fecha concreta marcada en un calendario.**

## 1.7. Referencias bibliográficas

Bravent. (2019, junio 3). *Cómo aplicar la cultura DevOps a tu empresa*.

<https://www.bravent.net/como-aplicar-la-cultura-devops-a-tu-empresa/>

Buhering, S. (2020, marzo 20). *What is DevOps: Free ebook*. Knowledge Train.

<https://www.knowledgetrain.co.uk/it/devops/what-is-devops>

Puppet. (2015). 2015 State of DevOps Report.

<https://puppet.com/resources/report/2015-state-devops-report/>

Humble, J. y Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley Professional.

### ¿Qué tareas hace un profesional de DevOps?

Fundación Telefónica. (2018, noviembre 26). *¿Qué tareas hace un profesional de DevOps?* | #ConectaEmpleo [Vídeo]. YouTube. <https://www.youtube.com/watch?v=MagivhkTrGc>

Este breve vídeo es un resumen de las tareas que desempeña un profesional de DevOps en una organización. Te animo a que lo visualices para que puedas comprender cómo se aplican los contenidos que hemos revisado en este tema a la vida real de un trabajador del sector en España.

### What is DevOps?

IBM Cloud. (2019, septiembre 9). *What is DevOps?* [Vídeo]. YouTube. <https://youtu.be/UbtB4sMaaNM>

Andrea Crawford, ingeniera y CTO ( *Chief Technology Officer*) de DevOps en IBM, explica con sus propias palabras qué es DevOps, cuál es valor que aporta DevOps a la organización y cómo las prácticas y herramientas de DevOps intervienen en el ciclo de vida de una aplicación.

1. ¿Cuál de las siguientes prácticas se corresponde con las metodologías DevOps?
  - A. Integración continua, entrega continua, gestión de la configuración y monitorización.
  - B. Atraer y contratar a más trabajadores para contar con una fuerza de trabajo más fuerte y numerosa.
  - C. Rediseñar la estrategia de marketing para atraer a una mayor cantidad de clientes.
  - D. Reestablecer los objetivos de negocio para aumentar la facturación y extender las directrices necesarias al resto de la organización para alcanzar esta meta.
  
2. ¿Cuáles de las siguientes afirmaciones acerca de DevOps son falsas?
  - A. La filosofía, cultura y herramientas DevOps son exclusivas para las *start-ups*.
  - B. Para la adopción de DevOps, solo es necesario que las áreas de desarrollo y operaciones trabajen en equipo, mientras el resto de la organización persigue sus objetivos individuales.
  - C. Para la implementación de DevOps es necesario que se haga un *push* a producción a diario.
  - D. Todas las anteriores.

3. ¿Qué objetivos persigue la estrategia DevOps?
- A. Aumentar la capacidad de una organización para entregar aplicaciones y servicios a alta velocidad.
  - B. Mejorar sus productos de forma continua a un ritmo más rápido que las organizaciones tradicionales.
  - C. Ofrecer un mejor servicio a sus clientes y competir en el mercado de forma más eficiente.
  - D. Todas las anteriores.
4. ¿Cuáles de los siguientes enunciados representan beneficios directos de la implementación de DevOps en una organización?
- A. Impacto positivo en la cuenta de resultados debido a un aumento de la facturación anual.
  - B. Aumento de la satisfacción laboral.
  - C. Aceleración de la innovación.
  - D. B y C son correctas
5. ¿Cuáles son los principios fundamentales de DevOps?
- A. La colaboración, mejora continua, responsabilidad equitativa, automatización de procesos, foco en el cliente, equipos multifuncionales y el aprendizaje en base a errores.
  - B. La colaboración y la automatización.
  - C. La gestión de la configuración, la automatización y la colaboración.
  - D. La integración y entrega continuas.

6. ¿Cuál definición de DevOps es la más acertada?
- A. DevOps es una combinación de filosofías, prácticas y herramientas culturales que utiliza software ágil para mejorar, gestionar y automatizar sus procesos.
  - B. DevOps es una metodología basada en la tecnología.
  - C. DevOps es un enfoque basado en la metodología en cascada que aboga por la cooperación e interacción entre los miembros de los equipos IT.
  - D. DevOps es una mejora en la comunicación entre los equipos.
7. ¿Qué impulsa a las empresas hacia la adopción de DevOps?
- A. La reducción del *time-to-market*.
  - B. La mejora de procesos a través de la automatización.
  - C. La mejora de la comunicación, la interacción y la colaboración entre los equipos.
  - D. Todas las anteriores.
8. ¿Cuál de los siguientes enunciados es correcto?
- A. Los equipos de operaciones deben identificar y resolver los problemas de forma temprana mediante la monitorización.
  - B. La monitorización es opcional para el enfoque DevOps, solo la virtualización es obligatoria y necesaria.
  - C. La virtualización en sí misma es opcional para implementar DevOps, pero reportará enormes beneficios en caso de que se integre.
  - D. A y C son correctas.



9. Termina la frase: «Las herramientas automatizadas...»
- A. Ofrecen oportunidades adicionales para mejorar la eficiencia.
  - B. Ayudan a acortar los ciclos de desarrollo.
  - C. Aceleran la entrega del producto.
  - D. Todas las anteriores.
10. El término *shift left* hace alusión a...
- A. El trabajo en entornos similares a producción durante el desarrollo del software y la ejecución de pruebas.
  - B. La automatización de las pruebas antes del pase a producción.
  - C. El trabajo denominado *pair programming* (programación en parejas) que llevan a cabo los desarrolladores en ocasiones.
  - D. Ninguna de las anteriores.