

INFORME ACCELERATE

State of DevOps 2022



Patrocinado por



Índice

01	Resumen ejecutivo	03	05	Sorpresas	55
02	¿Cómo lo estáis haciendo?	08	06	Datos demográficos y emporiográficos	59
03	¿Cómo podéis mejorar?		07	Comentarios finales	67
	Introducción	19	08	Agradecimientos	68
	Nube	21	09	Autores	69
	SRE y DevOps	26	10	Metodología	73
	Prestaciones técnicas de DevOps	29	11	Más información	76
	Cultura	37			
04	Por qué es importante la seguridad de la cadena de suministro	42			

01

Resumen ejecutivo



Derek DeBellis



Claire Peters

Llevamos ocho años elaborando el informe Accelerate State of DevOps, con la participación de 33.000 profesionales. Nuestra investigación se centra en examinar la forma en que las funciones y las prácticas influyen en los resultados que consideramos esenciales para DevOps:

- Rendimiento del envío de software – Las cuatro métricas clave del rendimiento del envío de software: frecuencia de despliegue, plazo de entrega de los cambios, índice de fallos por cambios y tiempo para restaurar el servicio.
- Rendimiento operativo – La quinta métrica clave: fiabilidad.
- Rendimiento organizativo – Con qué eficacia cumple tu organización con sus objetivos de rendimiento y rentabilidad.

También nos centramos en los factores que subyacen en otros resultados, como el desgaste y la probabilidad de que los empleados recomienden a sus equipos.



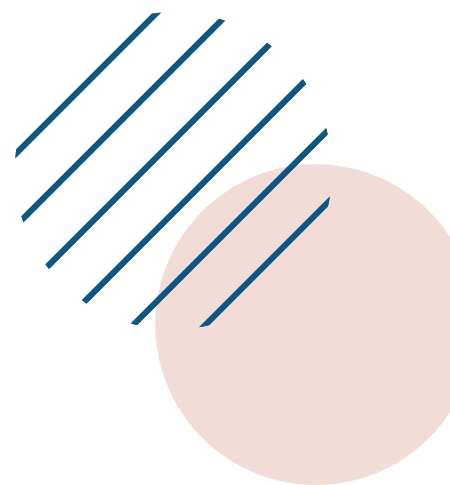
Proteger la cadena de suministro de software

En el 2021 descubrimos que proteger la cadena de suministro de software es esencial para alcanzar muchos resultados importantes.

Este año hemos ahondado en la seguridad de la cadena de suministro de software y es un tema central en nuestra encuesta y nuestro informe. Aprovechamos el framework [Supply Chain Levels for Secure Artifacts \(SLSA\)](#) para identificar las prácticas técnicas que facilitan el desarrollo de la seguridad de la cadena de suministro de software. También usamos el framework [Secure Software Development Framework \(SSDF\)](#) del Instituto Nacional de Normas y Tecnología (NIST) para identificar actitudes, procesos y prácticas genéricas relacionados con la seguridad de la cadena de suministro.

Descubrimos que los factores que mejor predicen las prácticas de seguridad para desarrollar aplicaciones de una organización son culturales y no técnicos: las culturas basadas en la confianza y no en la culpabilización tenían 1,6 veces más probabilidades de adoptar prácticas de seguridad de emergencia que la media, en comparación con aquellas más desconfiadas, culpabilizadoras y centradas en la jerarquía o las normas. También hallamos indicios que sugerían que hacer un análisis de seguridad antes del despliegue ayuda a encontrar dependencias vulnerables, lo que da como resultado menos vulnerabilidades en el código de producción.

La adopción de buenas prácticas de seguridad de desarrollo de aplicaciones suponía ventajas adicionales. Descubrimos que los equipos que se centran en establecer estas prácticas reducen el desgaste de sus desarrolladores, mientras que los equipos que no las priorizan tienen una probabilidad 1,4 veces mayor de desgastarse.¹ En consecuencia, los primeros suelen recomendar sus equipos a otras personas bastante más que los últimos. Además, las prácticas de seguridad relacionadas con SLSA predicen positivamente tanto el rendimiento organizativo como el del envío de software, pero para que tengan plena eficacia se requieren sólidas funciones de integración continua.



¹ En estas estadísticas, conceptualizamos "prioridad alta" como una desviación estándar de ≥ 1 en la puntuación (por ejemplo, seguridad), y "prioridad baja" como una desviación estándar de ≤ -1 .



Factores que impulsan el rendimiento organizativo

Además de las prácticas de seguridad mencionadas anteriormente, las variables clave que afectan al rendimiento organizativo suelen entrar en las siguientes categorías:

Cultura de organización y de equipo

Las culturas basadas en la confianza y no en la culpabilización, como las define [Westrum](#), suelen tener un mayor rendimiento organizativo. De modo parecido, las organizaciones cuyos equipos se sienten apoyados a través de patrocinios de financiación y liderazgo tienden a rendir mejor. Y lo mismo ocurre cuando los equipos son estables y sus integrantes los perciben de forma positiva (probabilidad de recomendarlo). Por último, las empresas que ofrecen modalidades de trabajo flexibles suelen tener altos niveles de rendimiento organizativo.

Fiabilidad

Tanto las prácticas de ingeniería que asociamos con la fiabilidad (por ejemplo, objetivos claros de fiabilidad, métricas de fiabilidad de las señales destacadas, etc.) como la medida en que los individuos afirman cumplir sus expectativas de fiabilidad son factores que predicen altos niveles de rendimiento organizativo.

Google Cloud

Descubrimos que el uso de la nube es un factor que predice el rendimiento organizativo. Las empresas cuyo software se desarrolló originalmente en la nube y para ella suelen tener un mayor rendimiento organizativo. El uso de nubes privadas, públicas, híbridas o una mezcla de varias nubes se corresponde con un mayor rendimiento organizativo que el uso exclusivo de servidores on-premise. Aquellos que usan varias nubes públicas tienen 1,4 veces más probabilidades de tener un rendimiento empresarial por encima de la media que los que no las usan.

El uso de la nube también parece influir en el rendimiento organizativo a través de otros factores que aparecen en nuestro conjunto de datos. Un ejemplo es la seguridad de la cadena de suministro, donde descubrimos que las organizaciones que usan nubes públicas tienen más probabilidades de implantar prácticas de SLSA, quizá porque los proveedores de servicios en la nube promueven y ofrecen los fundamentos de muchas prácticas de SLSA, como la automatización de compilaciones y despliegues.^{2,3} La conclusión general es que las plataformas en la nube permiten a los equipos adquirir muchas funciones y prácticas que acaban favoreciendo el rendimiento organizativo".

²Jung, Sun Jae. "Introduction to Mediation Analysis and Examples of Its Application to Real-world Data". Journal of preventive medicine and public health = Yebang Uihakhoe chi vol. 54,3 (2021): 166-172. doi:10.3961/jpmph.21.069


³ Carrión, Gabriel Cepeda, Christian Nitzl, and José L. Roldán. "Mediation analyses in partial least squares structural equation modeling: Guidelines and empirical examples". Partial least squares path modeling. Springer, Cham, 2017. 173-195.

El contexto es importante

Durante mucho tiempo, Dora ha tenido en consideración que el impacto depende del contexto global del equipo. Creemos que es importante comprender las características del equipo (procesos, puntos fuertes, limitaciones y objetivos) y el entorno de trabajo. Por ejemplo, una función técnica que es beneficiosa en un contexto puede ser perjudicial en otro distinto. Este año nos hemos centrado en modelar de manera explícita estas condiciones hipotéticas en forma de interacciones; muchas de dichas hipótesis están respaldadas por los datos de este año:

- Un rendimiento alto del envío de software solo es beneficioso para el rendimiento organizativo si el rendimiento operativo también es alto. Es posible que no importe que tengas un envío rápido si tu servicio no es capaz de cumplir las expectativas de fiabilidad de los usuarios.
- Implementar controles de seguridad de la cadena de suministro de software, como los recomendados por el framework SLSA, tiene un efecto positivo en el rendimiento del envío de software cuando la integración continua está bien establecida. Sin las funciones de integración continua, el rendimiento del envío de software y los controles de seguridad podrían entrar en conflicto.
- El impacto de las prácticas de Site Reliability Engineering (SRE) sobre la capacidad de un equipo de alcanzar sus objetivos de fiabilidad no es lineal. La práctica de SRE no tienen un efecto positivo en la fiabilidad hasta que el equipo no alcanza un cierto grado de madurez con SRE. Antes de que un equipo alcance dicho grado de madurez con SRE, no se detecta una relación entre la práctica de SRE y el logro de los objetivos de fiabilidad. Sin embargo, a medida que el equipo usa cada vez más la SRE, se alcanza un punto de inflexión en el que esta empieza a predecir atinadamente la fiabilidad. Y entonces, la mejora de la fiabilidad incide sobre el rendimiento organizativo.
- Las prestaciones técnicas se sustentan entre sí. La entrega continua y el control de versiones amplifican recíprocamente su capacidad de fomentar altos niveles de rendimiento del envío de software. La combinación de entrega continua, arquitectura con bajo acoplamiento, control de versiones e integración continua potencia un rendimiento del envío de software mayor que la suma de sus partes.





Tras considerar las condiciones de las que depende el envío y la necesidad de comprender el contexto global del equipo, llegamos a una conclusión similar a esta idea del 2021:

"Para lograr mejoras significativas, los equipos deben adoptar una filosofía de mejora continua. Usar los puntos de referencia para medir tu estado actual, identificar las restricciones según las funciones investigadas por el estudio y experimentar con mejoras para reducir dichas restricciones. La experimentación implica una combinación de victorias y fracasos, pero, en ambos casos, los equipos pueden adoptar medidas significativas como resultado de las lecciones aprendidas".

Efectivamente, este año descubrimos un efecto que está estrechamente ligado a esta filosofía global: los equipos que reconocen que deben seguir mejorando continuamente suelen tener un rendimiento organizativo más alto que los que no lo hacen.

En resumen, los equipos deben adaptarse continuamente y probar distintas prácticas de desarrollo de software.

Esto lo sabemos porque, por lo general, los equipos que lo hacen tienen un mayor rendimiento organizativo. Esto no es siempre así (lo que funciona para una organización no tiene por qué funcionar para otra), pero se aplica a la mayoría de los casos. A medida que vas experimentando con las prácticas de DevOps, prepárate para fracasar ocasionalmente hasta que des con lo que funciona para tu equipo.

Este año descubrimos varias [sorpresas](#) en los datos, pero tendrás que seguir leyendo para saber cuáles fueron.

Los equipos que reconocen que deben seguir mejorando continuamente suelen tener un rendimiento organizativo más alto que los que no lo hacen.

02

¿Cómo lo estáis haciendo?



Derek DeBellis

¿Tienes curiosidad por comparar tu equipo con el del resto de profesionales del sector? En esta sección se incluye la evaluación comparativa más reciente del rendimiento de DevOps. Examinamos la forma en que los equipos desarrollan, entregan y operan los sistemas de software y, a continuación, segmentamos a los encuestados en diferentes grupos que reflejan las combinaciones más comunes de rendimiento de DevOps.

Este año hemos incluido dos métodos de agrupación diferentes. El primero se basa en precedentes históricos. Este método de agrupación se centra en crear grupos con base en cuatro métricas que reflejan el rendimiento del envío de software: el plazo de entrega, la frecuencia de despliegue, el tiempo para restaurar el servicio y el índice de fallos por cambios. A continuación, resumimos cada una de ellas. El objetivo de esta metodología es ayudarte a cuantificar el rendimiento actual de tu equipo para que puedas compararlo con el de otros equipos.

El segundo método de agrupación incluye una quinta métrica, la fiabilidad, que utilizamos para entender el rendimiento operativo. ¿Por qué añadir una nueva métrica al análisis de grupos? Porque hemos comprobado sistemáticamente la importancia de esta métrica. Efectivamente, hemos visto indicios que sugieren que el rendimiento del envío puede perjudicar el rendimiento organizativo si no está acompañado de un sólido rendimiento operativo. A diferencia de nuestro método de agrupación tradicional, este es un ejercicio descriptivo que trata de exponer los resultados de los equipos en términos de rendimiento de entrega y operativo. Por tanto, no siempre es evidente qué grupo es mejor.

Primero, vamos a presentar brevemente las cinco métricas que utilizamos para entender el rendimiento del envío de software y el rendimiento operativo.

Rendimiento del envío de software y rendimiento operativo

Para satisfacer las necesidades de su sector en constante evolución, las organizaciones deben enviar y operar software de forma rápida y fiable. Cuanto más rápido puedan cambiar el software los equipos, antes podrás ofrecer valor a los clientes, realizar experimentos y recibir comentarios valiosos. Durante siete años de recopilación e investigación de datos, hemos desarrollado y validado cuatro métricas que miden el rendimiento del envío de software. Desde el 2018 incluimos una quinta métrica para reflejar las

funciones operativas. Los equipos que destacan en las cinco métricas presentan un rendimiento organizativo excepcional. Estas cinco métricas se denominan **rendimiento del envío de software y operaciones (SDO)**. Ten en cuenta que estas métricas se centran en los resultados a nivel de sistema, lo que ayuda a evitar los errores habituales en la medición de las métricas de software, que pueden resultar en enfrentar funciones entre sí y en hacer optimizaciones locales en detrimento de los resultados globales.

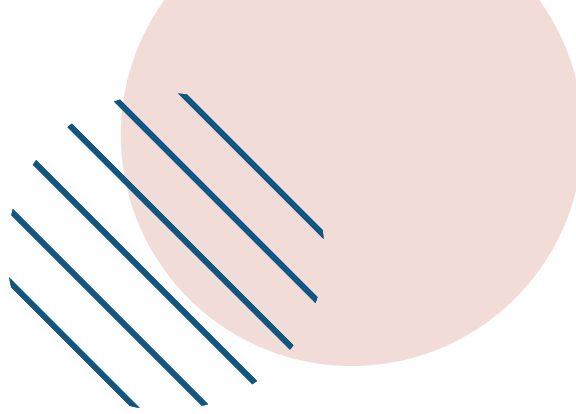


Las cinco métricas de rendimiento del envío y del rendimiento operativo

En términos de rendimiento y estabilidad, se pueden considerar cuatro métricas de rendimiento del envío de software. Medimos el rendimiento mediante el **tiempo de entrega de los cambios en el código** (es decir, el tiempo que transcurre desde que el código se confirma hasta que se lanza en producción) y la **frecuencia de despliegue**. Medimos la estabilidad usando el **tiempo para restaurar un servicio** después de un incidente y el **índice de fallos por cambios**.

La quinta métrica representa el **rendimiento operativo** y mide las prácticas operativas modernas. Basamos **el rendimiento operativo** en **la fiabilidad**, que es la medida en que tus servicios cumplen las expectativas del usuario, tales como la disponibilidad y el rendimiento. Antiguamente medíamos la disponibilidad en lugar de la fiabilidad, pero, dado que la disponibilidad se centra específicamente en la ingeniería de fiabilidad, en el 2021 incluimos la fiabilidad en nuestra medición para poder representar mejor la disponibilidad, la latencia, el rendimiento y la escalabilidad. En concreto, pedimos a los encuestados que evaluaran su capacidad para cumplir o superar sus objetivos de fiabilidad. Hemos observado que los equipos con diversos grados de rendimiento de envío obtienen mejores resultados (por ejemplo, menos desgaste) cuando también priorizan el rendimiento operativo.





Método de agrupación histórico: agrupación del rendimiento del envío

Este año, al evaluar el método de cuatro grupos que llevamos usando desde el 2018, nos dimos cuenta de que los datos mostraban un grupo con un bajo rendimiento evidente y otro con un alto rendimiento evidente. Sin embargo, los dos grupos que usábamos tradicionalmente para distinguir el rendimiento medio del alto no estaban lo suficientemente diferenciados como para justificar una división. Por otro lado, los indicios que usamos para seleccionar el método de agrupación adecuado

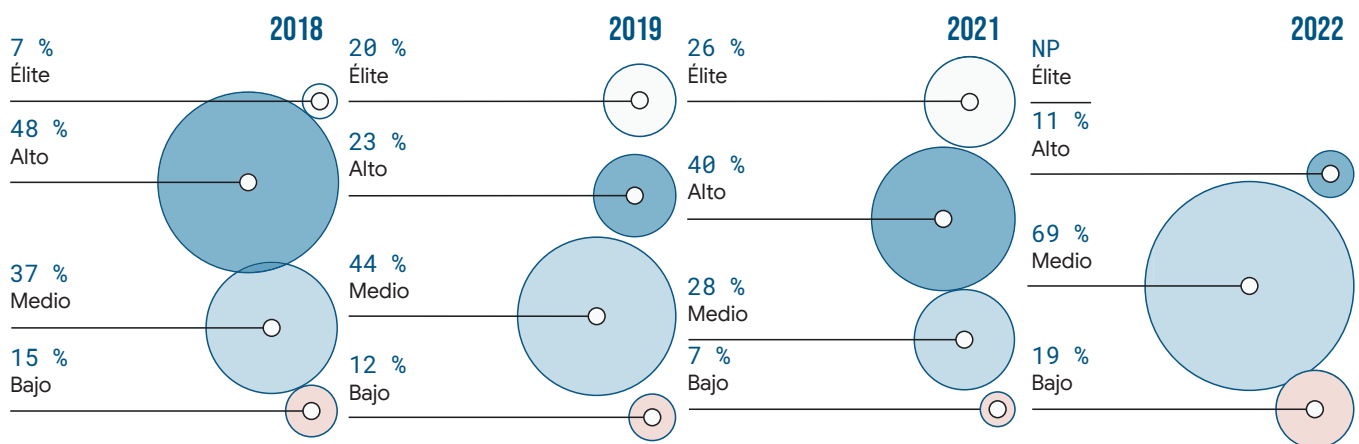
sugerían invariablemente que usar tres grupos refleja mejor los datos, independientemente del método de agrupación aplicado. En la siguiente tabla se describen las características de rendimiento del envío de cada grupo.

La mayor diferencia con respecto al año pasado es que este año no tenemos un grupo de élite. El grupo alto de este año es una mezcla de los grupos alto y de élite del año pasado. Decidimos omitir el grupo de élite porque el grupo con el rendimiento más alto no muestra suficientes de las características del grupo de élite del año pasado.

Métrica de rendimiento del envío de software	Bajo	Medio	Alto
Frecuencia de despliegue Respecto a la aplicación o servicio principal en los que trabajas, ¿con qué frecuencia despliega código tu organización en la fase de producción o lo lanza a los usuarios finales?	Entre una vez al mes y una vez cada seis meses	Entre una vez a la semana y una vez al mes	Bajo demanda (varios despliegues al día)
Plazo de entrega de los cambios Respecto a la aplicación o servicio principal en los que trabajas, ¿cuál es tu plazo de entrega de cambios (es decir, ¿cuánto se tarda desde que se confirma el código hasta que se publica con éxito en producción)?	Entre uno y seis meses	Entre una semana y un mes	Entre un día y una semana
Tiempo para restaurar el servicio Respecto a la aplicación o servicio principal en los que trabajas, ¿cuánto se suele tardar en restaurar el servicio cuando se produce un incidente o defecto que afecta a los usuarios (por ejemplo, una interrupción o alteración imprevistas del servicio)?	Entre una semana y un mes	Entre un día y una semana	Menos de un día
Índice de fallos por cambios Respecto a la aplicación o servicio principal en los que trabajas, ¿qué porcentaje de los cambios en la fase de producción o lanzados a los usuarios perjudican el servicio (por ejemplo, lo alteran o interrumpen) y, en consecuencia, requieren una solución inmediata (como un paquete de revisiones, una restauración, una corrección inmediata o un parche)?	46-60 %	16-30 %	0-15 %

Sugiere que esta muestra no representa a equipos u organizaciones cuyos empleados sienten que han avanzado. Una posible hipótesis, que en este momento no podemos fundamentar con datos, es que ha habido menos innovación en el campo del desarrollo de software en materia de prácticas, herramientas e intercambio de información. Esto puede ser consecuencia de la pandemia actual, que dificulta la capacidad de intercambiar conocimiento y prácticas entre equipos y organizaciones. Es posible que haya habido menos oportunidades de reunirse y aprender unos de otros, lo cual, a su vez, puede haber ralentizado la innovación. Esperamos ahondar más para averiguar las causas de este hallazgo.

De todos modos, si comparamos los grupos bajo, medio y alto de este año con los del año pasado, vemos que se ha producido un cambio hacia un rendimiento ligeramente más alto. Parece que los grupos de este año están entre dos de los del año pasado. El grupo alto del 2022 está entre los grupos alto y de élite del 2021. El grupo bajo del 2022 está entre los grupos bajo y medio del 2021. El cambio ascendente en el grupo de bajo rendimiento sugiere que, aunque el límite superior del rendimiento del envío ha bajado, el límite inferior ha subido.

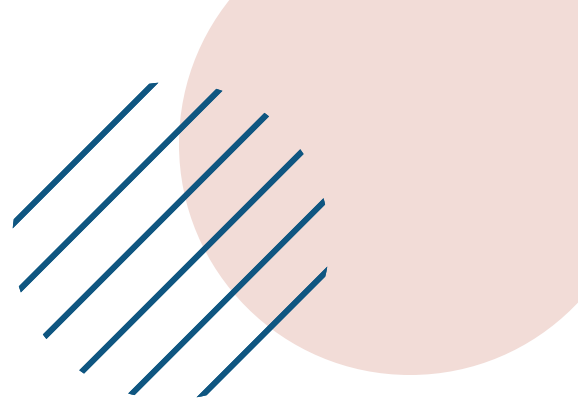


Los datos que aparecen en la tabla anterior muestran que el porcentaje de equipos con rendimiento de élite es el más bajo de los últimos cuatro años, mientras que el porcentaje de los de bajo rendimiento aumentó de forma espectacular: del 7 % en el 2021 al 19 % este año. Más de dos tercios de los encuestados este año entran en el grupo medio. La evidente caída en los equipos de alto rendimiento y de élite puede indicar que muchos de los encuestados este año están en organizaciones o equipos que, o bien no han establecido aún la cultura de DevOps que vemos surgir en muchos equipos modernos, o bien están en proceso de establecerla.

Puede que estemos centrándonos demasiado en las diferencias entre el 2021 y el 2022, en lugar de subrayar las similitudes. Los grupos del 2021 y el 2022 comparten muchas características, entre otras, una amplia separación entre los equipos de alto rendimiento y los de bajo rendimiento. Por ejemplo, se calcula¹ que los equipos de alto rendimiento hacen 417 veces más despliegues que los de bajo rendimiento.

¹ Consulta la sección "Metodología" para entender cómo se ha hecho este cálculo.





Agrupación del rendimiento del envío y del rendimiento operativo

Decidimos realizar un análisis de grupos en las tres categorías que representan las cinco métricas: **el rendimiento** (una combinación entre el plazo de entrega de los cambios en el código y la frecuencia de despliegue), **la estabilidad** (una combinación entre el tiempo para restaurar el servicio y el índice de fallos por cambios) y **el rendimiento operativo** (fiabilidad). La razón detrás de esta decisión es el papel fundamental que desempeña el rendimiento operativo

en nuestros modelos. En las organizaciones que no muestran un rendimiento operativo sólido, el rendimiento y la estabilidad tienen un impacto menor sobre el rendimiento organizativo. Sentimos que describir el panorama del rendimiento de DevOps sin tener en cuenta el rendimiento operativo deja fuera una parte crucial de dicho panorama.

Al analizar los datos llegamos a una solución de cuatro grupos. A continuación, presentamos un desglose de los cuatro grupos y sus nombres:

Grupo	Estabilidad		Rendimiento operativo	Rendimiento		% de encuestados
	Tiempo para restaurar el servicio	Índice de fallos por cambios	Fiabilidad	Plazo de entrega	Frecuencia de despliegue	
Empezando	Entre un día y una semana	31-45 %	A veces cumple las expectativas	Entre una semana y un mes	Entre una vez a la semana y una vez al mes	28 %
Fluyendo	Menos de una hora	0-15 %	Suele cumplir las expectativas	Menos de un día	Bajo demanda (varios despliegues al día)	17 %
Afrojando	Menos de un día	0-15 %	Suele cumplir las expectativas	Entre una semana y un mes	Entre una vez a la semana y una vez al mes	34 %
Ultimando	Entre uno y seis meses	46-60 %	Suele cumplir las expectativas	Entre uno y seis meses	Entre una vez al mes y una vez cada seis meses	21 %

No obstante, es posible que dos equipos del mismo grupo resulten ser muy diferentes y que nuestra tabla no lo refleje. Esta tabla es un intento de aplicar la experiencia que tenemos trabajando con diversos equipos para presentar patrones de forma inteligible. Además, si se analiza el mismo equipo en dos momentos diferentes, cabe la posibilidad de que no permanezca en el mismo grupo. Una posible razón es que el equipo haya mejorado o empeorado; otra, que haya cambiado su enfoque de despliegue por uno más adaptado a la situación actual de su aplicación o servicio. Por ejemplo, al empezar a desplegar su aplicación o servicio, un equipo determinado podría haberse centrado en explorar (grupo "Empezando"), pero, a medida que va encontrando su nicho, comienza a priorizar la fiabilidad (grupos "Fluyendo" o "Aflojando").

Cada grupo tiene unas características únicas y representa una proporción considerable de las respuestas.

El grupo **Empezando** no tiene un rendimiento ni bueno ni malo en ninguna de nuestras dimensiones. Es posible que se encuentre en las fases iniciales del desarrollo de su producto, función o servicio. Puede que esté menos centrado en la fiabilidad y más en recibir comentarios, entender cómo adaptar su producto al mercado y, a rasgos más generales, explorar.

El grupo **Fluyendo** tiene un buen rendimiento en todas las características: gran fiabilidad, estabilidad y rendimiento. Solo el 17 % de los encuestados se encuentran en este estado de fluidez.

Los encuestados en el grupo **Aflojando** no realizan despliegues muy a menudo, pero, cuando lo hacen, es probable que lo hagan con éxito. Más de un tercio de los encuestados entran en este grupo, por lo que esta muestra es la más grande y representativa. Este patrón suele ser típico (aunque en absoluto exclusivo) de un equipo que mejora progresivamente, pero que, al igual que sus clientes, está satisfecho por la mayor parte con el estado actual de su aplicación o producto.

Por último, el grupo **Ultimando** representa a los equipos que trabajan en un servicio o aplicación que sigue siendo valioso para ellos y sus consumidores, pero que ya no está en fase de desarrollo.

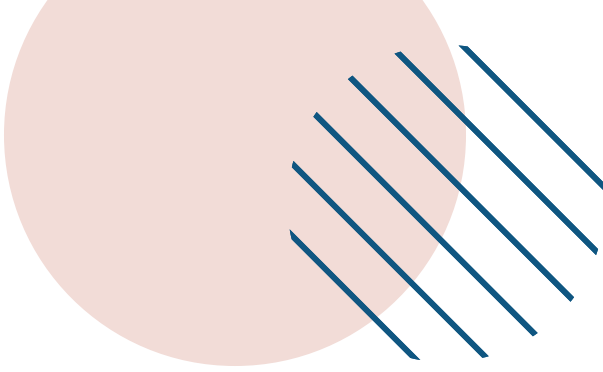
Estos grupos difieren considerablemente en sus prácticas y en sus prestaciones técnicas. Dado el alto rendimiento que demuestra el grupo **Fluyendo** en el envío de software y en sus operaciones, decidimos investigar qué diferencia a sus prácticas y prestaciones de las de otros grupos. Observamos que, en comparación con otros grupos, el grupo **Fluyendo** se centra más en:

- Arquitecturas con bajo acoplamiento: medida en que los equipos son capaces de realizar cambios a gran escala en el diseño de su sistema sin depender de otros equipos
- Flexibilidad: grado de flexibilidad que ofrece una empresa en cuanto a modalidades de trabajo
- Control de versiones: cómo se gestionan los cambios en el código de la aplicación, la configuración del sistema, la configuración de la aplicación, etc.
- Integración continua (CI): frecuencia con la que se integran las ramas en la troncal
- Entrega continua (CD): funciones que se centran en trasladar los cambios a la fase de producción de manera segura, sostenible y eficiente

Curiosamente, el grupo **Fluyendo** tiende a centrarse menos en la documentación. El año pasado observamos que las prácticas de documentación son clave tanto para el rendimiento de envío como para el rendimiento operativo (SDO). ¿Cómo es que el grupo **Fluyendo** tiene un alto rendimiento de SDO si no hace gran hincapié en la documentación? En primer lugar, hay muchas formas de obtener un alto rendimiento de SDO aparte de documentándose. Además, puede ser que el grupo **Fluyendo** refactorice su código continuamente para crear un proceso de documentación más automático y, de esa manera, tener menos necesidad de contar con documentos como los que describimos en la encuesta.

El grupo **Afrojando**, que representa la mayor parte de nuestros encuestados, suele estar compuesto de encuestados que pertenecen a grandes organizaciones que tienden a ser menos nativas de la nube que otros grupos. Imagina una empresa muy madura que, pese a tener algunos procesos anquilosados, sigue ofreciendo a sus usuarios finales experiencias estables, fiables y valiosas. Este grupo manifiesta una cultura generativa orientada hacia el rendimiento.² Una de las características más interesantes del grupo "Afrojando" es que tiene un bajo rendimiento y una cultura positiva de trabajo (una cultura de trabajo "generativa", según Westrum); una combinación poco habitual. Es más común observar un rendimiento proporcional a la cultura de trabajo (alto/alta o bajo/baja). Esperamos seguir investigando para comprender mejor la relación entre el rendimiento y la cultura.

² (Westrum)



También comparamos los diferentes grupos en cuanto a tres factores: desgaste, rendimiento organizativo y trabajo imprevisto. Lo que descubrimos echó por tierra nuestras expectativas. El grupo **Ultimando** tenía mejor rendimiento organizativo que el resto de grupos. Al observar las características de este grupo (poca estabilidad y bajo rendimiento), esto parecía contradecir la mayoría de los hallazgos previos de DORA. Pero, en lugar de juzgar este hecho como una anomalía aleatoria (muy posible), queremos investigar las posibles causas.

A la hora de discernir estos hallazgos, es importante tener en cuenta otro dato. El grupo **Ultimando** logra un alto rendimiento organizativo a un gran precio: sus equipos tienen el índice más alto de desgaste, mayor propensión a cometer errores y una mayor carga de trabajo imprevisto. En conjunto, estos resultados apuntan a que la fiabilidad puede ser suficiente para lograr un alto rendimiento organizativo; no obstante, sin velocidad ni estabilidad, tu equipo pagará el precio de desgastarse y cargar con trabajo imprevisto.

Tenemos otras hipótesis para explicar por qué el grupo **Ultimando** tiene un mayor rendimiento organizativo que el resto, en particular, que el grupo **Fluyendo**. A continuación, ofrecemos una lista resumida.

- Hay características que subyacen a estos cuatro grupos que tal vez no se vean en nuestros datos. Por ejemplo, el tamaño de la organización puede ser un buen indicador de su madurez. En el grupo **Fluyendo** suelen estar las empresas más pequeñas, lo que puede indicar que sus productos se encuentran en etapas de desarrollo.
- Los miembros del grupo **Fluyendo** suelen pertenecer a empresas más pequeñas, que pueden estar menos sujetas a procesos e infraestructuras antiguos y, en consecuencia, cuentan con procesos de DevOps más modernos. Dicho esto, los datos demuestran que existe una relación positiva entre el tamaño de la organización y su rendimiento operativo, por razones que pueden tener poco que ver con la tecnología.
- El grupo **Fluyendo** suele pasar por alto los principios descritos en la cultura generativa de Westrum. Hemos observado que, a menudo, esto perjudica el rendimiento organizativo.
- Las organizaciones pueden entender y supervisar las expectativas de fiabilidad de forma diferente, así como tener un concepto distinto de los objetivos de rendimiento organizativo.

- Puede que el grupo **Ultimando** tenga mayor rendimiento organizativo a corto plazo, pero cabe preguntarse cómo será a largo plazo. ¿Se traducirá el desgaste en mayor rotación de personal? ¿Serán capaces de escalar sus procesos?
- Planteamos estas preguntas a diversos niveles. Las preguntas sobre prestaciones técnicas (por ejemplo, arquitectura con bajo acoplamiento) se plantean a nivel de equipo; y las preguntas sobre rendimiento organizativo, a nivel de organización. A menudo las organizaciones cuentan con muchos equipos, y es posible que el encuestado reconozca que, aunque su organización funciona bien, su equipo no.

Además, el grupo **Fluyendo** tiene la segunda puntuación más alta en rendimiento organizativo, por detrás del grupo **Ultimando**, y sus niveles de desgaste y trabajo imprevisto están entre los más bajos. Tal y como demuestran los grupos **Fluyendo** y **Aflojando**, la filosofía de DevOps funciona mejor cuando hay fiabilidad.

Queremos seguir explorando nuevas formas de describir variaciones en el sector. En el futuro queremos seguir teniendo en cuenta el rendimiento operativo como una dimensión pertinente para entender estas variaciones. También queremos evitar los grupos muy prescriptivos y evaluadores (por ejemplo, el de élite) y, en su lugar, centrarnos en el ejercicio descriptivo de identificar semejanzas entre los rendimientos de SDO.



03

¿Cómo podéis mejorar?



Derek DeBellis

¿Cómo podéis mejorar muchos de los resultados?

El informe "State of DevOps" está diseñado para ofrecer una guía factual que ayude a tu equipo a centrarse en las prácticas y funciones de DevOps que te otorgan los resultados que te importan. Este año, ampliamos nuestra investigación para incluir la seguridad y el conjunto de resultados deseados por los equipos. Antes nos centrábamos en los resultados del rendimiento del envío de software y de operaciones (SDO) y del rendimiento organizativo. Seguimos haciéndolo, pero también queremos estudiar el desgaste, la probabilidad de recomendar al equipo, el trabajo imprevisto y la propensión a cometer errores, no solo como vía para mejorar el rendimiento de SDO y organizativo, sino también como un fin en sí mismos. Por lo tanto, este año nos aseguramos de identificar las prácticas y prestaciones que parecen incidir en estos resultados.



El modelo de investigación se reorientó para reflejar mejor una teoría sobre la que se basa DORA: no existe un único planteamiento para DevOps. En la práctica, hemos observado que para hacer recomendaciones hay que conocer el contexto global del equipo. Una práctica beneficiosa para un equipo podría ser perjudicial para otro. Por ejemplo, llevamos mucho tiempo con la hipótesis de que las prestaciones técnicas (como la arquitectura con bajo acoplamiento, el desarrollo basado en troncales y la integración continua) favorecen mucho más el rendimiento del envío de software cuando hay entrega continua. Este año hemos modelado explícitamente esta y otras interacciones. El objetivo es mejorar nuestro entendimiento del simple "¿qué afecta a qué?" para incluir "¿bajo qué circunstancias existen, se amplifican o se atenúan estos efectos?". Comprender todas estas condiciones ha resultado ser un proceso complicado y desafiante, pero por fin podemos compartir contigo algunos resultados iniciales.

Puedes consultar los modelos de investigación de este año y años anteriores por Internet en [nuestro sitio web](#).



Más allá de las cuatro claves

¿Cómo mejoran las métricas de DORA el rendimiento en materia de desarrollo y operaciones? Un equipo multidisciplinario de ingenieros de software de Liberty Mutual Insurance revisa regularmente el rendimiento usando las ["cuatro métricas clave" de DORA](#). Por ejemplo, Jenna Dailey, Scrum Master sénior en Liberty Mutual, comentó que un equipo aprovechó la investigación de DORA para ayudarles a identificar un cuello de botella, adoptar un enfoque basado en la realización de pruebas y mejorar su rendimiento general.

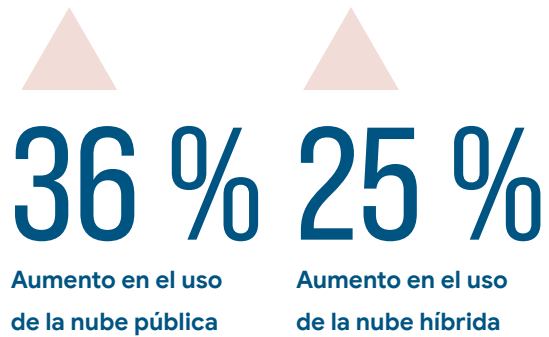
Tienes más información sobre el planteamiento de Liberty Mutual de emplear los datos y las métricas de DORA para mejorar la calidad y el envío del software en sus [Tomorrow Talks](#).



Eric Maxwell

Google Cloud

A partir del impulso que hemos visto en los últimos años, el uso del cloud computing sigue acelerándose. De hecho, el porcentaje de personas que afirman usar la nube pública, incluso varias nubes, ha subido del 56 % en el 2021 al 76 % ahora. El número de personas que confiesan no usar la nube en absoluto, incluidas las que no usan la nube privada, ha bajado del 21 % el año pasado al 10,5 %. El uso de varias nubes públicas aumentó del 21 % al 26 % y el uso de la nube híbrida ha subido del 25 % al 42,5 %. También hemos observado un ligero aumento en el uso de la nube privada, del 29 % el año pasado al 32,5 % ahora.



El uso del cloud computing ha tenido un efecto positivo en el rendimiento organizativo general. Los encuestados que usaban la nube tenían **un 14 % más de probabilidades de lograr** sus objetivos de rendimiento organizativo que los que no la utilizaban.

Como hemos mostrado en años anteriores y seguimos confirmando en este informe, el uso del cloud computing tiene un efecto positivo en el rendimiento organizativo global. Los encuestados que usaban la nube tenían **un 14 % más de probabilidades de lograr** sus objetivos de rendimiento organizativo que los que no la utilizaban. Según nuestros estudios, los equipos que usan el cloud computing mejoran la seguridad y fiabilidad de la cadena de suministro de software, que a su vez favorecen al rendimiento organizativo.

Sorprendentemente, los usuarios de todo tipo de nubes —pública, privada, híbrida y multinube— mostraban una relación negativa con el índice de fallos por cambios, que era superior. Es necesario investigar más al

respecto. En lugar de especular acerca de las razones, seguiremos investigando en los próximos estudios. Sin embargo, pese a contadas excepciones, el uso de aplicaciones nativas de la nube (las que se diseñaron y desarrollaron originalmente para la nube) arrojó indicios positivos en todo lo que investigamos.

El uso de cualquier plataforma de cloud computing, ya sea pública o privada, favorece la cultura y el entorno de trabajo (como una cultura generativa, menos desgaste, más estabilidad y mayor satisfacción del personal). Los usuarios que usaban la nube obtuvieron unos resultados culturales un 16 % mejores.

El uso de la nube sigue incrementando anualmente

	2022	% de cambio en el 2021
Nube híbrida	42,47 %	25 %
Una o varias nubes públicas	76,08 %	36 %
Nube privada	32,55 %	12 %
Ninguna nube	10,55 %	-50 %

La nube híbrida y el entorno multinube (así como la nube privada) parecen tener un efecto **negativo** en los indicadores de rendimiento del envío de software (MTTR, plazo de entrega y frecuencia de despliegue), **a no ser que** los encuestados tengan un nivel alto de **fiabilidad**.

Los entornos híbrido y multinube mejoran el rendimiento organizativo

Seguimos viendo claros indicios de que usar la nube híbrida o varias nubes públicas tiene un efecto positivo en las organizaciones. Los profesionales que usaban varias nubes mostraron un rendimiento organizativo 1,4 veces mayor que los que no usaban ninguna. Sin embargo, la nube híbrida y el entorno multinube (así como la nube privada) parecen tener un efecto negativo en varios indicadores de envío de software (MTTR, plazo de entrega y frecuencia de despliegue), a no ser que los encuestados tengan un nivel alto de fiabilidad. Este hallazgo respalda la importancia de contar con una

práctica sólida de SRE y del papel que desempeña la fiabilidad en el envío de software.

En el 2021 preguntamos a los encuestados cuál era su motivo *principal* para usar varias nubes públicas, mientras que en el 2022 les pedimos que nos contaran las ventajas que les reporta utilizar varios proveedores de servicios en la nube. La mayor ventaja según la mayoría es la disponibilidad, lo cual coincide con la atención e hincapié puestos en la fiabilidad que hemos visto en el sector; los servicios no pueden ser fiables si no están disponibles. Más del 50 % de los profesionales señalaron que utilizaron diversos proveedores de servicios en la nube para aprovechar sus ventajas únicas.

Ventajas de usar varios proveedores de servicios en la nube

Disponibilidad	62,61 %
Se aprovechan las ventajas únicas de cada proveedor	51,59 %
No se deposita toda la confianza en un solo proveedor	47,54 %
Recuperación tras fallos	43,48 %
Cumplimiento de los requisitos legales	37,97 %
Requisito de táctica o aprovisionamiento de negociación	19,13 %
Otra	4,06 %

> 50 %

Encuestados que afirmaron usar varios proveedores de servicios en la nube

Aprovechar las cinco características del cloud computing es un primer paso crucial en una larga cadena causal que conduce a un buen rendimiento organizativo.

Las cinco características del cloud computing

Para seguir con nuestro método de investigación anterior, no queríamos saber solo si los participantes usan tecnologías de cloud computing, sino también cómo las usan. Esto lo conseguimos preguntando por las cinco características esenciales del cloud computing, tal como las define el Instituto Nacional de Normas y Tecnología.

El autoservicio bajo demanda: los consumidores pueden aprovisionar los recursos de computación según sea necesario, de forma automática y sin intervención humana por parte del proveedor.

Amplio acceso a la red: las prestaciones están muy disponibles y los consumidores pueden acceder a ellas a través de varios clientes, como teléfonos móviles, tablets, portátiles y estaciones de trabajo.

Agrupamiento de recursos: los recursos de los proveedores se agrupan en un modelo multiciente, con recursos físicos y virtuales, que se asigna y reasigna de forma dinámica bajo demanda.

Por lo general, el cliente no tiene control directo sobre la ubicación exacta de los recursos proporcionados, pero puede especificarla en un nivel superior de abstracción, como el país, la provincia o estado, o el centro de datos.

Elasticidad rápida: las funciones se pueden aprovisionar y lanzar de forma elástica para escalarse rápidamente hacia dentro o hacia fuera con la demanda. Las funciones de los consumidores disponibles para el aprovisionamiento parecen ser ilimitadas y pueden usarse en cualquier cantidad y momento.

Servicio medido: los sistemas en la nube controlan y optimizan de forma automática el uso de los recursos mediante una función de medición en un nivel de abstracción adecuado para el tipo de servicio, como almacenamiento, procesamiento, ancho de banda y cuentas de usuario activas. El uso de recursos se puede monitorizar, controlar y evaluar para garantizar la transparencia.

El informe valida los tres años de investigación de DORA y concluye que la presencia de estas cinco características en una organización tiene un impacto positivo sobre el rendimiento del envío de software y el de operaciones. También observamos que estas características conducen a un mejor rendimiento organizativo, poniendo en marcha procesos que tienen un efecto positivo en la organización. Presentar las cinco características del cloud computing es el primer paso en un largo camino que conduce a un mayor rendimiento organizativo.

En el 2022 observamos que los equipos se benefician cada vez más de los factores de diferenciación del cloud computing. Por cuarto año consecutivo, estamos observando una adopción creciente de las cinco características del cloud computing. El agrupamiento de recursos tuvo un aumento del 14 %, el mayor, y la elasticidad rápida, que fue la segunda característica más usada el año pasado, vio el aumento más bajo, del 5 %.

NIST	2021	2022	Cambio porcentual
Amplio acceso a la red	74 %	80 %	8
Elasticidad rápida	77 %	81 %	5
Autoservicio bajo demanda	73 %	78 %	7
Servicio medido	78 %	83 %	7
Agrupamiento de recursos	73 %	83 %	14





Dave Stanke

SRE y DevOps

Un equipo tecnológico exitoso hace más por su organización que enviar código; más incluso que enviar código de calidad. También garantiza que sus servicios permanezcan disponibles, rindan adecuadamente y cumplan las expectativas de los usuarios a lo largo del tiempo. La fiabilidad es una métrica polifacética que mide con qué eficiencia mantiene un equipo estos compromisos y, este año, seguimos indagando en la fiabilidad como factor en el envío y las operaciones de software.

Site Reliability Engineering (SRE) es un enfoque operativo influyente que se originó en Google y que ahora adoptan muchas organizaciones. SRE prioriza el aprendizaje empírico, la colaboración multidisciplinaria, la automatización y el uso de técnicas de medición como los objetivos de nivel de servicio. Otras prácticas operativas modernas emplean métodos similares, pero aplican diferentes convenciones de nomenclatura. Por tanto, para evaluar la magnitud de estas prácticas de la manera más objetiva posible, nos aseguramos de usar lenguaje neutro y descriptivo en nuestra encuesta. También recopilamos datos sobre los resultados de ingeniería de fiabilidad: la medida en que los equipos

logran alcanzar sus objetivos de fiabilidad. Tanto las técnicas como los resultados (prácticas de SRE y resultados de fiabilidad) están reflejados en nuestro modelo predictivo, junto con otras funciones de DevOps.

La fiabilidad es esencial

Todos los equipos a los que encuestamos adoptan la SRE: la mayoría de ellos aplican una o varias de las prácticas sobre las que preguntamos. Entre esta gran variedad de equipos, los datos revelaron una relación sutil entre la fiabilidad, el envío de software y los resultados: si la fiabilidad es baja, el envío de software no influye tanto en el éxito de la organización. Por el contrario, si la fiabilidad es alta, un buen envío de software influye positivamente en los resultados de la empresa.

Sin fiabilidad, el rendimiento del envío de software no predice el éxito de la organización.

Invertir en SRE ofrece mejoras en la fiabilidad, pero solo cuando se ha alcanzado un determinado umbral de adopción.

Este fenómeno es coherente con el uso del framework de "presupuesto de errores" de SRE: si un servicio no es fiable, a los usuarios no les ayudará enviar código más rápido dentro de un contexto frágil.

Como los ingenieros de Site Reliability Engineering llevan tiempo afirmando, la fiabilidad es la "característica" más importante de cualquier producto. Nuestra investigación confirma la observación de que mantener las promesas a los usuarios es una condición necesaria para que un envío de software mejorado beneficie a la organización.

Reconocer la curva en J

¿Qué desafíos te aguardan en el camino hacia la fiabilidad? En "Enterprise Roadmap to SRE",¹ publicado en O'Reilly, James Brookbank y Steve McGhee, colaboradores en las encuestas de DORA, reflexionan sobre sus experiencias a la hora de implementar SRE en organizaciones consolidadas y recomiendan "reconocer la curva de cambio en J". La "curva en J", ya descrita en el informe "State of DevOps" del 2018, consiste en que las transformaciones organizativas suelen dar buenos resultados al principio y, luego, pasar por periodos de menor, o incluso peor, rendimiento.

¹ <https://sre.google/resources/practices-and-processes/enterprise-roadmap-to-sre/>

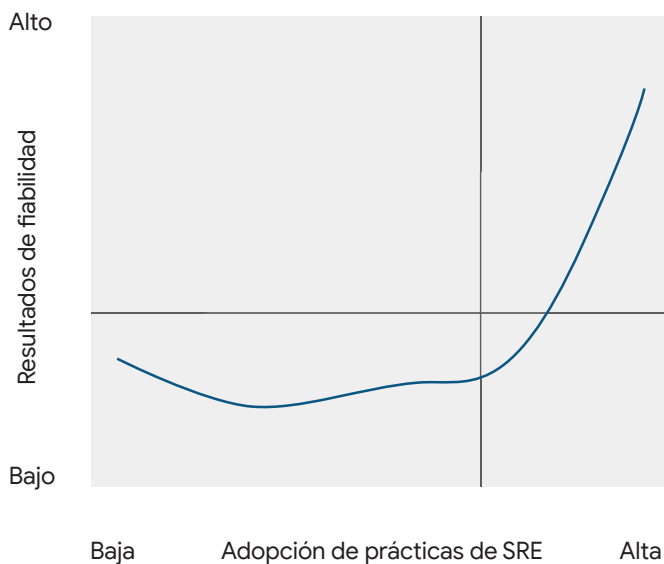
Sin embargo, los que superan esos desafíos iniciales suelen acabar logrando sus objetivos de forma renovada, constante y más fructífera.

Este año, nuestro estudio revela un patrón de curva en J en todos los equipos que analizamos: cuando los equipos emplean menos prácticas de ingeniería de fiabilidad (lo que sugiere que se encuentran en la fase inicial de adopción de SRE), estas prácticas no predicen mejores resultados de fiabilidad. Sin embargo, a medida que los equipos adoptan SRE en mayor medida, llegan a un punto de inflexión en el que el uso de SRE comienza a predecir firmemente la fiabilidad y, a su vez, el rendimiento organizativo.



Los equipos fiables producen servicios fiables: una cultura de trabajo generativa predice una mejor fiabilidad.

Los equipos que se encuentran en la primera etapa de la adopción de prácticas de SRE tienen que estar preparados para sufrir contratiempos en su trayectoria. Puede llevar tiempo para que la cultura, los procesos y las herramientas se readapten a los nuevos principios rectores. Pero pueden estar seguros de que, con tiempo e inversión continua, es probable que tengan éxito.



Los equipos que continúan adoptando la SRE tras la fase inicial aprecian mejores resultados a raíz de una mayor fiabilidad

Invertir en personas, procesos y herramientas

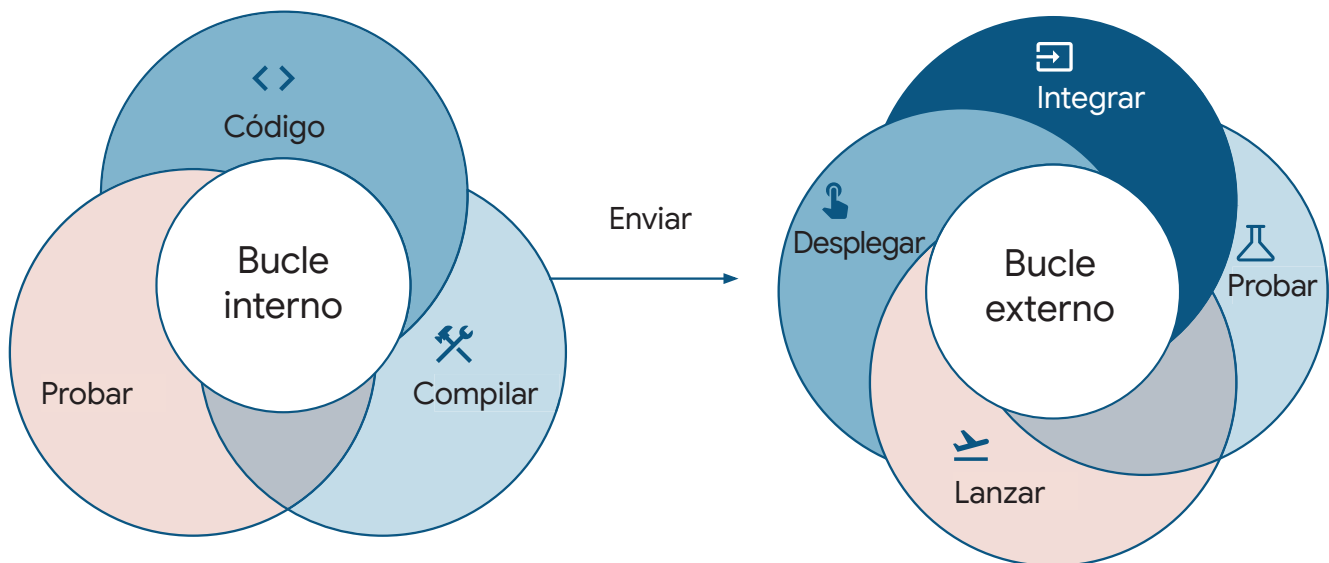
La fiabilidad es una labor humana y, de muchas maneras, el planteamiento de SRE así lo ejemplifica. Uno de los principios centrales de la SRE es que el verdadero indicador de fiabilidad es la percepción del usuario, no la monitorización interna de los datos. Por lo que quizá no es de extrañar que las dinámicas de equipo positivas fomenten la fiabilidad. Observamos que los equipos que tenían una cultura "generativa" en la que existía confianza y colaboración tenían más probabilidades de practicar SRE y de lograr buenos resultados de fiabilidad. Los equipos estables, cuya permanencia se mantiene a lo largo del tiempo, también prestan servicios más fiables a los usuarios. Y, en cuanto a DevOps en su conjunto, los esfuerzos humanos de ingeniería de fiabilidad mejoran con el apoyo adicional de procesos y herramientas. Prácticas como el uso del cloud computing y la integración continua predicen mejores resultados de fiabilidad.

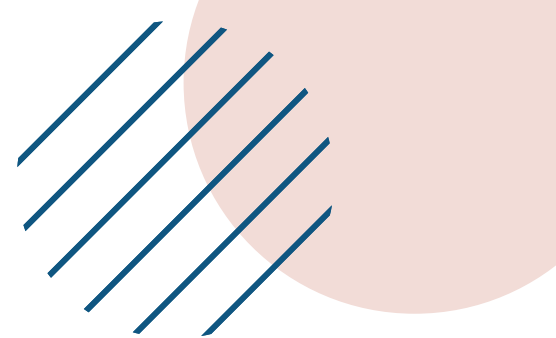


Eric Maxwell

Prestaciones técnicas de DevOps

Este año analizamos varias prestaciones técnicas para entender los resultados basados en las diferentes prácticas técnicas. Tuvimos en cuenta dos fases amplias de desarrollo del software: el "bucle interno", que abarca tareas como la programación, las pruebas y el envío al control de versiones; y el "bucle externo", que incluye actividades como la combinación de código, la revisión automatizada de código, la ejecución de pruebas, el despliegue y el lanzamiento.





Los equipos con alto rendimiento que cumplen sus objetivos de fiabilidad tienen **1,4 veces más probabilidades de usar la integración continua.**

Nuestro estudio muestra que las empresas que destacan en el desarrollo de bucle interno y externo son capaces de enviar código más rápido y con mayores niveles de fiabilidad. Las funciones que más contribuyen a tener un rendimiento alto son el control de versiones, la integración continua, la entrega continua y la arquitectura con bajo acoplamiento.

Los equipos con alto rendimiento que cumplen sus objetivos de fiabilidad son:

33 % más propensos a usar el control de versiones

39 % más propensos a practicar la integración continua

46 % más propensos a practicar la entrega continua

40 % más propensos a tener sistemas basados en una arquitectura con bajo acoplamiento

De hecho, los encuestados que usan las funciones mencionadas más que la media tienen **un rendimiento organizativo 3,8 veces más alto** que aquellos que no las usan.

Integración continua

La integración continua, o CI, es una parte del proceso de desarrollo del bucle externo que compila automáticamente un artefacto y ejecuta una serie de pruebas automatizadas para cada confirmación de código, con el fin de evaluar si el código está listo para ser desplegado. El proceso ofrece una evaluación automatizada al desarrollador, lo que le permite operar con mayores niveles de confianza. La CI es una parte clave en el proceso de trasladar el código de la estación de trabajo del desarrollador al entorno de producción. Como en los años anteriores, la CI ha demostrado impulsar el rendimiento de envío. **Los equipos con alto rendimiento que cumplen los objetivos de fiabilidad tienen 1,4 veces más probabilidades de usar CI que otros.**

Este año, ahondamos en la otra parte del proceso de desarrollo del bucle externo: la entrega continua, que describiremos más adelante. Pero antes vamos a hablar de un componente complementario de la integración continua: el desarrollo basado en troncales.

Desarrollo basado en troncales

El desarrollo basado en troncales es la práctica de fusionar el código continuamente con la troncal y evitar ramas de larga duración. Se considera que esta práctica es un complemento de la integración continua y durante años ha demostrado que acelera el envío de software.

Debido al cambio en los datos demográficos de este año relativos a la experiencia laboral, hemos podido comprobar que la experiencia importa a la hora de implementar el desarrollo basado en troncales. El año pasado, el 40 % de los encuestados afirmaban llevar más de 16 años trabajando y, este año, esa categoría representaba solo el 13 %. En sintonía con nuestro tema de "dependencias del envío", hemos observado que los desarrolladores con menor experiencia tienen resultados menos positivos en el desarrollo basado en troncales y que observan:

- ▼ **Una reducción** general en el rendimiento del envío de software
- ▲ **Un incremento** de la cantidad de trabajo imprevisto
- ▲ **Un incremento** de la propensión a cometer errores
- ▲ **Un incremento** del índice de fallos por cambios

Los individuos con más de 16 años de experiencia que usan el desarrollo basado en troncales reconocen las ventajas de esta práctica y observan:

- ▲ **Un incremento** del rendimiento general del envío de software
- ▼ **Una reducción** de la cantidad de trabajo imprevisto
- ▼ **Una reducción** de la propensión a cometer errores
- ▼ **Una reducción** del índice de fallos por cambios

Probablemente se deba a las prácticas adicionales que son necesarias para implementar con éxito el desarrollo basado en troncales. Los equipos que no aplican normas rigurosas para no ignorar nunca una troncal rota, que no usan ramas de código restringido o que restauran automáticamente el código que rompe la troncal, seguramente tendrán dificultades a la hora de desarrollar soluciones en la troncal.

Por otro lado, cuando se practica el desarrollo basado en troncales, la repercusión sobre el rendimiento organizativo general es positiva.



Frank Xu

Entrega continua

La entrega continua es una práctica de desarrollo de software que:

1. Permite al equipo desplegar el software en la fase de producción o a los usuarios finales en cualquier momento.
2. Garantiza que el software esté en un estado apto para el despliegue a lo largo de su ciclo de vida, incluso cuando se trabaja en nuevas características.
3. Establece un rápido bucle de retroalimentación que permite al equipo comprobar la calidad y la capacidad de despliegue del sistema, y prioriza la resolución de errores que bloquean el despliegue.

Ten en cuenta que la entrega continua no implica necesariamente un despliegue continuo, consistente en desplegar automáticamente cada compilación de software. La entrega continua solo requiere que una compilación pueda desplegarse en cualquier momento.

El año pasado analizamos las prestaciones técnicas de DevOps que predicen la probabilidad de que un equipo practique CD y descubrimos que la arquitectura con bajo acoplamiento y las pruebas e integración continuas eran algunos de los factores que más influían. Este año, además de analizar los factores que impulsan el uso de CD, analizamos e identificamos los efectos de la CD en solitario, así como en sus interacciones con otras funciones de DevOps, sobre los resultados de desarrollo.

La CD impulsa el rendimiento del envío de software

Al igual que comprobamos en años anteriores, el uso de la CD es un factor que predice un mayor rendimiento del envío de software, tanto en solitario como en combinación con otras funciones de DevOps. Los equipos con mayor puntuación en CD tienen más probabilidades de desplegar código en la fase de producción con más frecuencia y de tener un menor plazo de entrega de cambios y de restauración del servicio.

Los equipos que combinan el control de versiones y la entrega continua tienen una probabilidad 2,5 veces mayor de tener un rendimiento alto del envío de software que los equipos que solo se centran en una de las dos.

Además, los encuestados tienen una probabilidad 2,5 veces mayor de tener un rendimiento más alto del envío de software si su equipo adopta también las prácticas de control de versiones.

La CD puede incrementar el trabajo imprevisto

Los datos sugieren que la entrega continua hace que los desarrolladores tengan que dedicar más tiempo a tareas de reelaboración o trabajo imprevisto. Una hipótesis que explica este hallazgo es que los desarrolladores son más propensos a compilar aplicaciones de forma iterativa si hay bucles de retroalimentación más estrictos. Como resultado, puede que consideren algunos cambios iterativos como trabajo imprevisto en la misma parte del sistema. Este trabajo puede ser imprevisto y, a la vez, estar motivado por la retroalimentación de un despliegue previo.

Prácticas técnicas y CD

Nuestra investigación ha demostrado una y otra vez que la CD se sustenta sobre un amplio conjunto de prestaciones técnicas. Este año investigamos qué ocurre cuando algunas de esas funciones individuales se usan junto a la CD. Descubrimos que el desarrollo basado en troncales y la arquitectura con bajo acoplamiento, junto con la CD, podrían tener un impacto negativo en el rendimiento del equipo. Por ejemplo, vimos indicios de que los equipos que adoptan arquitecturas con bajo acoplamiento en combinación con la CD tienen un 43 % más de probabilidades de anticipar propensión a cometer errores por encima de la media (como interrupciones, vulnerabilidades de seguridad y una disminución significativa del rendimiento de sus servicios), en comparación con los equipos que adoptaban solo la CD. Es necesario seguir investigando estos efectos, que parecen señalar una posible fricción en los equipos que están mejorando. Puede que esta fricción se deba a la curva de transformación en J, que explica que los equipos que logran mejoras al principio empiezan a estancarse después de haber logrado los objetivos más fáciles. Para alcanzar el máximo potencial es necesario comprometerse a mejorar. A la hora de mejorar cualquier función, como puede ser la CD, asegúrate de observar los efectos que tiene en el equipo y el rendimiento general.



David Farley

Arquitectura con bajo acoplamiento

Los sistemas con bajo acoplamiento son importantes para la eficacia de los equipos y las organizaciones. Esto no se refiere solo a los servicios en la nube o a los basados en microservicios: tiene que ver con la capacidad de una organización para hacer cambios. La facilidad de una organización para hacer cambios en su software con seguridad y confianza es un indicador de la calidad del software.

Gracias a la arquitectura con bajo acoplamiento, los equipos pueden:

- Hacer cambios de gran envergadura en el diseño de su sistema sin depender de que otros equipos tengan que hacer cambios en sus sistemas
- Conseguir retroalimentación más rápido mediante pruebas independientes bajo demanda con menor coste de coordinación
- Desplegar código con un periodo de inactividad insignificante

En el informe de este año hemos preguntado a los encuestados si el software que compilan está basado en una arquitectura con bajo acoplamiento o no. El resultado fue muy interesante y mostraba una variedad de asociaciones, en su mayoría positivas, entre la existencia de arquitectura con bajo acoplamiento y el rendimiento del equipo en varias dimensiones.



Las ventajas de una arquitectura con bajo acoplamiento

Los equipos que se centran en compilar software con arquitecturas con bajo acoplamiento tienen más posibilidades de lograr buena estabilidad, fiabilidad y rendimiento. Estos equipos también tienen más probabilidades de recomendar su lugar de trabajo a un amigo o compañero.

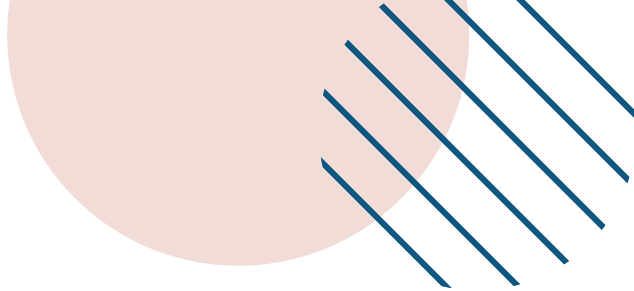
A menudo observamos que los equipos que despliegan en la nube y adoptan una estrategia de arquitectura de microservicios utilizan software con arquitectura con bajo acoplamiento para gestionar

Los equipos que se centran en compilar software con arquitecturas con bajo acoplamiento tienen más posibilidades de obtener buena estabilidad, fiabilidad y rendimiento.

cientos de servicios. No obstante, el bajo acoplamiento es más que una métrica para contabilizar los servicios de un sistema. En una arquitectura con bajo acoplamiento, los componentes se pueden desplegar de manera independiente. Esta independencia permite a los equipos desarrollar, hacer pruebas y desplegar sus servicios sin tener que invertir mucho para coordinarse con otros equipos.

En la vida real, el bajo acoplamiento no está limitado a un solo estilo de arquitectura, sino que se trata fundamentalmente de la capacidad de realizar cambios en una parte del sistema sin que otras partes se vean afectadas. Esto permite a las organizaciones dividir su trabajo para que los equipos puedan avanzar sin tener que coordinarse con otros equipos.

Según nuestra experiencia, los equipos que necesitan realizar pruebas de integración profunda en otros servicios como forma de aumentar la confianza del software antes del despliegue no han alcanzado aún el nivel de bajo acoplamiento. Para lograrlo, les ayudaría mejorar las interfaces y el aislamiento entre sistemas. Una forma eficaz de mejorar las interfaces y el aislamiento es mejorar la capacidad de hacer pruebas en los servicios y componentes. Si tu diseño te permite hacer pruebas en tu servicio de forma aislada, entonces la interfaz tiene por definición bajo acoplamiento.



También observamos que los equipos estables y bien integrados que usan arquitectura con bajo acoplamiento tienen más probabilidades de usar prácticas de desarrollo de software que impulsan y fomentan una mejora continua. Por ejemplo, las prácticas de SRE, como establecer objetivos de fiabilidad para priorizar el trabajo o revisar dichos objetivos regularmente de manera empírica, fomentan la arquitectura con bajo acoplamiento.

Las arquitecturas con bajo acoplamiento también facilitan que la organización contrate a más empleados, ya que los equipos independientes que no se tienen que coordinar con otros tienen más libertad para aumentar el tamaño de su equipo de forma independiente.

En resumen, el impacto del bajo acoplamiento en los servicios de software va más allá del aspecto técnico. También influye en los aspectos sociotécnicos del desarrollo de software. El acoplamiento es la base de la ley de Conway: la idea de que los sistemas de diseño de una organización reflejan su propia estructura de comunicación. Un número mayor de sistemas con bajo acoplamiento se traduce en organizaciones "con un acoplamiento más bajo", con un enfoque de desarrollo más distribuido y escalable.



Hallazgos sorprendentes

La investigación de este año reveló que la arquitectura con bajo acoplamiento puede contribuir al desgaste de los equipos. Este hallazgo es sorprendente y contradice los resultados de años anteriores. Nuestro análisis muestra que los equipos estables en los que la información fluye libremente tienen menores niveles de desgaste. Tanto la cultura generativa de Westrum como la estabilidad del equipo favorecen la arquitectura con bajo acoplamiento y reducen el desgaste, por lo que esto es una clara contradicción. Es preciso seguir investigando antes de extraer una conclusión definitiva.

Al mismo tiempo, si los requisitos de seguridad están definidos y controlados por una organización de seguridad consolidada, puede costarles más a los equipos desacoplar su software del de los otros equipos. Esto confirma las ventajas de trasladar las cuestiones de seguridad al equipo con más responsabilidad sobre la aplicación (consulta también: [Por qué es importante la seguridad de la cadena de suministro](#)). Este caso es una de las formas de acoplamiento más sutiles en las organizaciones y, aunque nosotros hemos recopilado datos sobre seguridad, es probable que esta situación se dé también en otras funciones centralizadas. Permitir a los equipos que tomen sus propias decisiones en materia de seguridad y en otras funciones frecuentemente centralizadas es una forma de recoger los frutos del uso que la arquitectura con bajo acoplamiento puede aportar a tu organización.



Daniella Villalba

Cultura

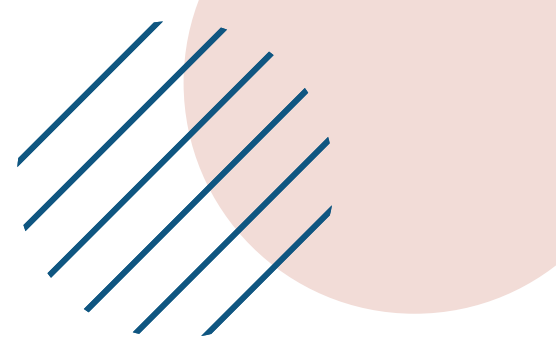
"Es que aquí hacemos las cosas así".

Esta frase se habrá utilizado en innumerables ocasiones y en muchos sectores para describir cómo aborda una organización los retos y las oportunidades.

La cultura de cada organización es única y nuestra investigación ha demostrado continuamente que es un elemento fundamental para el éxito de una organización y el bienestar de sus empleados.

La cultura también es un aspecto necesario de DevOps, ya que, al nivel más básico, DevOps se basa en herramientas, prácticas y **formas de colaborar de las personas** para desarrollar y entregar software de manera rápida, fiable y segura. Conocer los factores que afectan a la cultura de la organización puede ayudar a la dirección de la empresa a abordar los desafíos culturales resueltamente. Por ello, debería ser prioritario para las organizaciones fomentar una cultura sana. Si no se abordan estos desafíos culturales, pueden llegar a obstaculizar la implementación de las prácticas de DevOps.





Este año hemos vuelto a usar la tipología organizativa de Westrum para medir la salud de la cultura de las organizaciones. Además, hemos ampliado nuestro conocimiento sobre la cultura midiendo la tasa de abandono de los empleados, las modalidades de trabajo flexibles, la percepción del compromiso de la empresa y el desgaste.

Los datos de la investigación de este año confirman los hallazgos previos que señalan que **el rendimiento organizativo** se ve influido por el tipo de cultura de la organización. En concreto, una **cultura generativa** se asocia con mayores niveles de rendimiento organizativo, en comparación con las organizaciones que tienen una cultura burocrática o patológica. Los empleados que

trabajan en organizaciones donde hay una cultura generativa tienen más probabilidades de pertenecer a equipos estables, producir documentación de mayor calidad y dedicar la mayor parte del tiempo a realizar trabajo significativo.

Abandono del equipo

Hemos investigado el abandono del equipo y observamos **que los equipos estables**, aquellos cuya composición no ha cambiado mucho en los últimos doce meses, tenían más probabilidades de estar en organizaciones con alto rendimiento. El abandono constante puede afectar a la productividad y a la moral, ya que el proceso de incorporación de nuevos empleados lleva tiempo. Además, aquellos que permanecen en el equipo

Cultura empresarial de Westrum

Patológica <i>Centrada en las relaciones de poder</i>	Burocrática <i>Centrada en las normas</i>	Generativa <i>Centrada en el rendimiento</i>
Poca cooperación	Cooperación moderada	Mucha cooperación
Se culpa al mensajero	Se ignora al mensajero	Se forma al mensajero
Se elude la responsabilidad	Responsabilidad limitada	Se comparte la responsabilidad
No se recomienda establecer vínculos	Se tolera establecer vínculos	Se recomienda establecer vínculos
Los fallos llevan a buscar chivos expiatorios	Los fallos llevan a injusticias	Los fallos llevan a indagaciones
Se evita la innovación	La innovación lleva a problemas	Se implementa la innovación

tendrían que adaptarse a los cambios en su carga de trabajo y en la dinámica de equipo. Por otro lado, según nuestra investigación, los equipos estables afirmaban producir documentación de calidad más que los equipos donde había mayor tasa de abandono. A los equipos que tienen que adaptarse constantemente a cambios les cuesta más mantenerse al día de las prácticas que ayudan a producir documentación de calidad.

Las organizaciones con un alto rendimiento tienen más probabilidades de disponer de modalidades de trabajo flexibles.

Modalidades de trabajo flexibles

Dado que muchas organizaciones han adoptado modalidades de trabajo flexibles desde la pandemia del COVID-19, hemos investigado si ofrecer a los empleados la libertad de trabajar a distancia,

en persona o una modalidad híbrida se asociaba con un mayor rendimiento organizativo. Según los resultados, las organizaciones que ofrecen más **flexibilidad a sus empleados** tienen un rendimiento organizativo mayor que las que tienen una modalidad de trabajo más rígida. Estos resultados demuestran que ofrecer flexibilidad a los empleados para que adapten su entorno de trabajo según lo necesiten conlleva beneficios tangibles y directos para la organización.

Desgaste

El desgaste es un sentimiento de temor, apatía y escepticismo hacia el trabajo. No solo agota y desmotiva, sino que también acarrea insatisfacción laboral, lo que puede aumentar la rotación de personal. El desgaste se ha relacionado con muchos y muy diversos problemas de salud física y mental, como un mayor riesgo de sufrir depresión y ansiedad, padecer cardiopatías y tener ideas de suicidio¹.

El año pasado medimos el desgaste dentro del contexto del COVID-19 y observamos que la cultura generativa estaba vinculada a un índice menor de desgaste en los empleados.

¹ Maslach C, Leiter MP. Understanding the burnout experience: recent research and its implications for psychiatry. World Psychiatry. 2016 Jun;15(2):103-11. doi: 10.1002/wps.20311. PMID: 27265691; PMCID: PMC4911781.

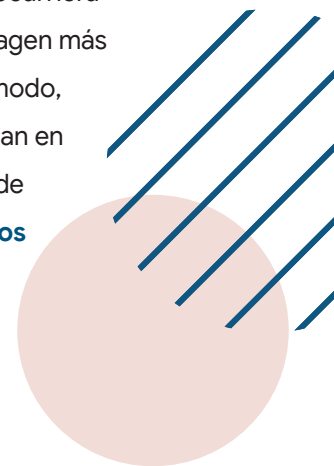
Las modalidades de trabajo flexibles están asociadas con una reducción del desgaste del personal y aumentan la probabilidad de que los empleados recomienden su lugar de trabajo.

Este año replicamos este resultado y ampliamos nuestro conocimiento sobre el desgaste mostrando que los equipos estables y las modalidades de trabajo flexibles también estaban asociadas con un menor desgaste. Además, este año hemos incluido el Net Promoter Score (NPS), que indica si una persona recomendaría su equipo a amigos o compañeros. Observamos que el NPS del equipo estaba ligado al compromiso del equipo directivo que perciben los empleados. Y, al reflejar los resultados del desgaste, observamos que factores como una cultura generativa, un equipo estable y una modalidad de trabajo flexible hacen que sea más probable que los empleados recomienden su equipo a otras personas.

Cómo perciben los empleados su organización

Por último, investigamos la percepción del compromiso del equipo directivo pidiendo a los encuestados que predijeran la cantidad de apoyo que esperaban recibir durante los doce meses siguientes. Los resultados mostraron que una mejor percepción del compromiso del equipo directivo (como apoyo financiero, más asignación de recursos, patrocinios) estaba asociada con organizaciones con alto rendimiento.

También pedimos a los encuestados que predijeran la probabilidad de que ocurriera una infracción de seguridad o una interrupción total del servicio en los próximos doce meses. Los resultados mostraron que las personas que trabajan en organizaciones con alto rendimiento esperaban menos que ocurriera un error importante; estas tenían una imagen más positiva de su organización. Del mismo modo, observamos que las personas que trabajan en organizaciones con un alto rendimiento de software y de envío **se planteaban menos** la necesidad de cambiar sus prácticas actuales para mejorar los resultados empresariales.



Hablemos de representación

Según nuestros resultados, los empleados de grupos infrarrepresentados en general dedicaban más tiempo a trabajo imprevisto, independientemente de que pertenecieran a organizaciones con alto o bajo rendimiento. También observamos que los empleados de grupos infrarrepresentados se desgastaban más que los que no pertenecían a esos grupos. Los jefes de equipo deben estar al tanto del riesgo que supone el desequilibrio en la carga de trabajo y garantizar que las tareas se distribuyan de manera justa entre los integrantes del equipo.

Conjuntamente, estos resultados subrayan la importancia de fomentar un entorno laboral saludable e inclusivo, tanto a nivel de organización como a nivel de equipo.

Aunque no dejamos de hacer hincapié en la importancia de la cultura, reconocemos que no es fácil cambiar o mejorar siquiera la cultura de una organización. Recomendamos a las organizaciones que primero procuren entender las experiencias de sus empleados y, después, inviertan en recursos para abordar los problemas culturales, como parte de las iniciativas de transformación de DevOps.



04

Por qué es importante la seguridad de la cadena de suministro

En noviembre del 2020, relativamente pocos profesionales del sector de la tecnología sospechaban que se avecinaba una crisis de seguridad de la cadena de suministro de software. La [Open Source Security Foundation](#) se fundó para ocuparse de la seguridad del software de código abierto, continuando los esfuerzos previos, y, aunque se produjeron [avances positivos](#) para resolver el problema, el tema no apareció en la portada de ningún periódico importante. Un ataque importante [a SolarWinds](#)

lo cambió todo. Cuando los atacantes logran infiltrarse silenciosamente en miles de redes de empresas importantes y del Gobierno a través de actualizaciones de software con troyanos, las cosas cambian rápido.

Hoy en día, la seguridad de la cadena de suministro de software es un tema por lo general apremiante; tal vez no en la cena, pero sí en la sala de reuniones.



John Speed Meyers



Todd Kulesza



Aparte de las numerosas iniciativas que hay en marcha, muchas empresas de software se han comprometido a reformar sus propias prácticas de seguridad de la cadena de suministro de software y a mejorar la seguridad del código abierto compartido.

En este capítulo nos centraremos en dos iniciativas: Supply Chain Levels for Software Artifacts ([SLSA](#), pronunciado "salsa" en inglés), y el Secure Software Development Framework del NIST ([SSDF](#)). Cada una ofrece una serie de medidas defensivas para asegurar que los atacantes no pueden manipular los procesos de producción del software y esquivar a los defensores de la red mediante actualizaciones maliciosas de software.

Pero ¿en qué medida se usan las prácticas de seguridad de cadena de suministro de software asociadas con SLSA y SSDF? ¿Qué prácticas necesitamos ayudar a implantar y cuáles están ya muy extendidas? Hasta la fecha, no encontramos respuestas sistemáticas a estas preguntas. Tras encuestar a cientos de profesionales de software sobre qué prácticas adoptan para salvaguardar la cadena de suministro, podemos ofrecer algunas respuestas. En particular, destacan cuatro hallazgos principales:

01 La adopción ya ha empezado: Las prácticas de seguridad de la cadena de suministro de software están integradas en SLSA y ya existe cierto grado de adopción del SSDF, pero todavía se puede hacer más.

02 Las culturas más sanas tienen ventaja: La cultura de la organización es el factor principal que promueve las prácticas de seguridad de desarrollo de software. Las culturas más dadas a confiar que a culpar tienen más probabilidades de aplicar las prácticas de SLSA y del SSDF que las culturas organizativas más recelosas.

03 Hay un elemento clave de integración: La adopción de los aspectos técnicos de la seguridad de la cadena de suministro de software parece cimentarse en el uso de integración continua / entrega continua (CI/CD), que a menudo ofrece la plataforma de integración para muchas prácticas de seguridad de la cadena de suministro.

04 Proporciona ventajas inesperadas: Además de reducir los riesgos de seguridad, unas buenas prácticas de seguridad conllevan más ventajas, como un menor desgaste profesional.

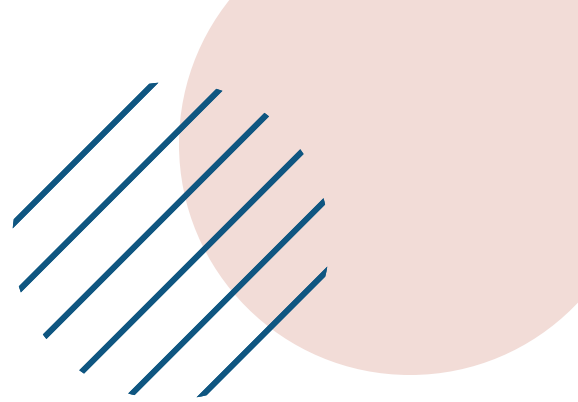
¿Qué hacen las empresas actualmente para evitar vulnerabilidades de seguridad?

Para saber mejor lo que hacen las organizaciones hoy en día con el fin de identificar y resolver las vulnerabilidades de seguridad en el software que compilan, hemos añadido más de una veintena de preguntas a la encuesta de este año. Estas preguntas entran generalmente dentro de dos categorías:

- Preguntas que piden a los encuestados que indiquen si están o no de acuerdo con una afirmación (por ejemplo: "mi organización tiene un método eficaz para abordar las amenazas de seguridad" o "tengo acceso a las herramientas necesarias para ejecutar pruebas de seguridad").
- Preguntas que piden a los encuestados que indiquen la medida en que se aplican las prácticas de seguridad en la organización (por ejemplo: las compilaciones están definidas por las secuencias de comandos y nada más" o "los lanzamientos en la fase de producción se compilan usando un sistema centralizado de CI/CD, nunca en la estación de trabajo del desarrollador"). Usamos la escala "se aplican/no se aplican" porque las pruebas iniciales mostraban que los encuestados se inclinaban más a estar de acuerdo con algunas de las preguntas sobre seguridad. Por otro lado, las preguntas acerca del SSDF se formularon de manera más natural, con la escala de "estoy de acuerdo / no estoy de acuerdo".

El **framework de SLSA**, en v0.1 en estos momentos, describe una serie de prácticas de integridad de la cadena de suministro de software asociadas con los "niveles" de SLSA. Los niveles más altos se correspondían con unos mayores niveles de garantía de seguridad de la cadena de suministro de software. Preguntamos a los encuestados acerca de muchas de las prácticas relacionadas concretamente con SLSA. En particular, en la encuesta se preguntaba: "¿En qué medida se aplican las siguientes prácticas en el servicio o aplicación principal en los que trabajas?". En la Tabla 1 se enumeran las prácticas relacionadas con SLSA que aparecen en la encuesta.

El **SSDF**, actualmente en v1.1, se centra en las prácticas que ayudan a las organizaciones a enviar software con menos vulnerabilidades y minimizar el posible impacto del resto de vulnerabilidades. En lugar de usar los "niveles" de SLSA, las prácticas del SSDF están agrupadas en cuatro categorías: preparar a la organización, proteger el desarrollo del software, producir un software seguro y responder eficazmente a las vulnerabilidades encontradas. En la encuesta se pedía a los encuestados que indicaran en qué medida estaban de acuerdo (o de desacuerdo) con las afirmaciones que describían varias de las prácticas de SSDF. Estas preguntas se resumen en la Tabla 2.



Práctica de SLSA	Definición de la encuesta
CI/CD centralizada	Los lanzamientos se compilan usando un sistema centralizado de CI/CD, nunca en la estación de trabajo del desarrollador
Conservación del historial	Las revisiones y su historial de cambios se conservan indefinidamente
Secuencia de comandos de la compilación	Las compilaciones están plenamente definidas por la secuencia de comandos de compilación y nada más
Aislamiento	Las compilaciones están aisladas; no pueden interferir con las compilaciones simultáneas ni consecutivas
Archivos de texto de la compilación	Las definiciones y configuraciones de la compilación están definidas en archivos de texto almacenados en el sistema de control de versiones
Parámetros en los metadatos	Los metadatos de la compilación (como las dependencias, el proceso de compilación o el entorno de desarrollo) de un artefacto incluyen todos los parámetros de la compilación
Dependencias en los metadatos	En los metadatos de la compilación (como las dependencias, el proceso de compilación o el entorno de desarrollo) de un artefacto están documentadas todas las dependencias
Metadatos generados	Los metadatos de la compilación (como las dependencias, el proceso de compilación o el entorno de desarrollo) los genera, o bien el servicio de compilación, o bien un generador de metadatos de la compilación que lee el servicio de compilación
Impedir entradas	Cuando se ejecuta una compilación, se impide que los pasos de compilación carguen cualquier entrada de compilación dinámica (por ejemplo, todas las fuentes y dependencias necesarias se obtienen por adelantado)
Los usuarios no pueden hacer cambios	Los usuarios de los servicios de compilación no pueden cambiar los metadatos de la compilación (como las dependencias, el proceso de compilación o el entorno de desarrollo) de un artefacto
Metadatos disponibles	Los metadatos de la compilación (como las dependencias, el proceso de compilación o el entorno de desarrollo) están disponibles para quienes los necesiten (como a través de una base de datos central) y se proporcionan en un formato que ellos acepten
Revisión de dos personas	Todos los cambios del historial de revisión deben revisarse y aprobarse de forma independiente por dos personas de confianza antes de enviarse
Metadatos firmados	Los metadatos de la compilación (como las dependencias, el proceso de compilación o el entorno de desarrollo) acerca de cómo se produjo un artefacto los firma mi servicio de compilación

Tabla 1. Preguntas de seguridad relacionadas con SLSA

Nota: Los encuestados tenían cinco respuestas posibles para cada pregunta: "no se aplica nunca", "se aplica a veces", "se aplica con relativa frecuencia", "se aplica a menudo" y "se aplica siempre".

Prácticas de SSDF	Definición de la encuesta
Revisiones de seguridad	Se lleva a cabo una revisión de seguridad en todas las características principales de las aplicaciones en las que trabajo
Análisis o pruebas del código continuos	Realizamos constantemente análisis y pruebas automatizados o manuales con el código de todas las publicaciones admitidas, con el fin de identificar o confirmar la presencia de vulnerabilidades que no se hayan detectado antes
Pruebas de seguridad iniciales	Yo u otro miembro de mi equipo realizamos pruebas de seguridad cuando se empieza a desarrollar el software
Abordar las amenazas eficazmente	Mi organización tiene un método eficaz para abordar las amenazas de seguridad
Integración en el equipo de desarrollo	Las funciones de seguridad están integradas en el equipo de desarrollo del software
Requisitos de documentación	Nuestra organización tiene procesos para identificar y documentar todos los requisitos de seguridad del software que desarrolla o adquiere (incluido el software de terceros y de código abierto)
Revisión periódica de los requisitos	Los requisitos de seguridad se revisan periódicamente (una vez al año o más si es necesario)
Metadatos generados	Los metadatos de la compilación (como las dependencias, el proceso de compilación o el entorno de desarrollo) los genera, o bien el servicio de compilación, o bien un generador de metadatos de la compilación que lee el servicio de compilación
Integración en el ciclo de desarrollo	En mi empresa, el protocolo de seguridad de software está integrado eficientemente en nuestro proceso de desarrollo
Proceso estándar en todos los proyectos	En mi empresa tenemos un proceso estandarizado para abordar la seguridad del software en todos los proyectos
Monitorizar los informes de seguridad	Actualmente monitorizamos la información que proviene de fuentes públicas relativa a posibles vulnerabilidades del software que usamos y de sus componentes de terceros
Disponer de las herramientas necesarias	Tengo acceso a todas las herramientas necesarias para ejecutar pruebas de seguridad

Tabla 2. Preguntas de seguridad relacionadas con SSDF

Nota: Los encuestados tenían siete respuestas posibles para cada pregunta: "totalmente en desacuerdo", "en desacuerdo", "algo en desacuerdo", "ni de acuerdo ni en desacuerdo", "algo de acuerdo", "de acuerdo" y "totalmente de acuerdo".

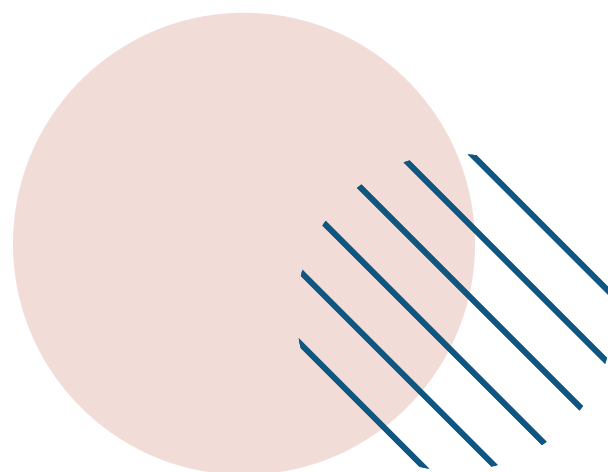


En general, observamos niveles de adopción relativamente amplios de las prácticas emergentes del sector, aunque todavía se puede hacer más por consolidarlas. Por ejemplo, aunque el 66 % de los encuestados estaban de acuerdo con la afirmación "en mi empresa el protocolo de seguridad de software está integrado eficientemente en nuestro proceso de desarrollo", solo el 18 % estaba totalmente de acuerdo. En las figuras 1 y 2 se resumen las respuestas de los encuestados a nuestras preguntas sobre la seguridad.

Observamos que el uso de sistemas de CI/CD en los lanzamientos era la práctica que más se aplicaba, y el 63 % de los encuestados respondió que se aplicaba "a menudo" o "siempre". El hecho de que CI/CD esté en lo alto de la lista concuerda con la investigación previa sobre seguridad, según la cual [la mayoría de las organizaciones implementa un análisis de seguridad a nivel de aplicación como parte de su proceso de CI/CD](#). Además, otro grupo distinto de entrevistas cualitativas sobre seguridad sugería que la mayoría de los desarrolladores eran incapaces de ejecutar tales herramientas localmente durante el proceso de desarrollo. El framework SLSA se basa igualmente en sistemas de CI como punto central de integración en la seguridad de la cadena de suministro. Según nuestro análisis de modelos, descrito en la siguiente sección, la presencia de CI en una organización es un factor

que predice la madurez de sus prácticas de seguridad. Por lo tanto, creemos que, sin esta pieza clave de infraestructura, es muy difícil que una organización pueda garantizar que se ejecuten una serie de análisis y pruebas en los artefactos de software que crea.

Además de CI/CD, otras prácticas que se aplican frecuentemente son la conservación indefinida del historial del código (60 %), compilaciones que solo están definidas por secuencias de comandos (58 %), aislamiento de las compilaciones (57 %) y almacenamiento de las definiciones de compilación en el control de fuente (56 %). Las dos prácticas que menos se aplican son el requisito de que dos o varios revisores aprueben cada cambio que se hace en el código (45 %) y firmar los metadatos de la compilación para evitar o detectar manipulaciones (41 %).



Además de las preguntas sobre las prácticas implantadas, también pedimos a los encuestados que indicaran en qué medida estaban de acuerdo o en desacuerdo con una serie de afirmaciones sobre la seguridad en su organización. La afirmación con la más de acuerdo estaban los encuestados (81 %) es "actualmente monitorizamos la información que proviene de fuentes públicas relativa a posibles vulnerabilidades del software que usamos y de sus componentes de terceros". En el extremo opuesto, la afirmación con la que menos de acuerdo estaban los encuestados es la referente a los efectos negativos de las prácticas de seguridad en el desarrollo del software: el 56 % estaba de acuerdo con que "los procesos de seguridad del software que hay en mi empresa ralentizan el desarrollo de las aplicaciones en las que trabajo".

Aunque este resultado es positivo, el hecho de que la mayoría de los encuestados afirmaran que sus procesos actuales de seguridad ralentizan el desarrollo parece indicar que todavía se pueden mejorar las herramientas y los métodos de seguridad. Nuestro análisis de modelos también respalda esta afirmación, ya que muestra efectos desiguales (aunque de poca importancia) en el rendimiento del envío del software.



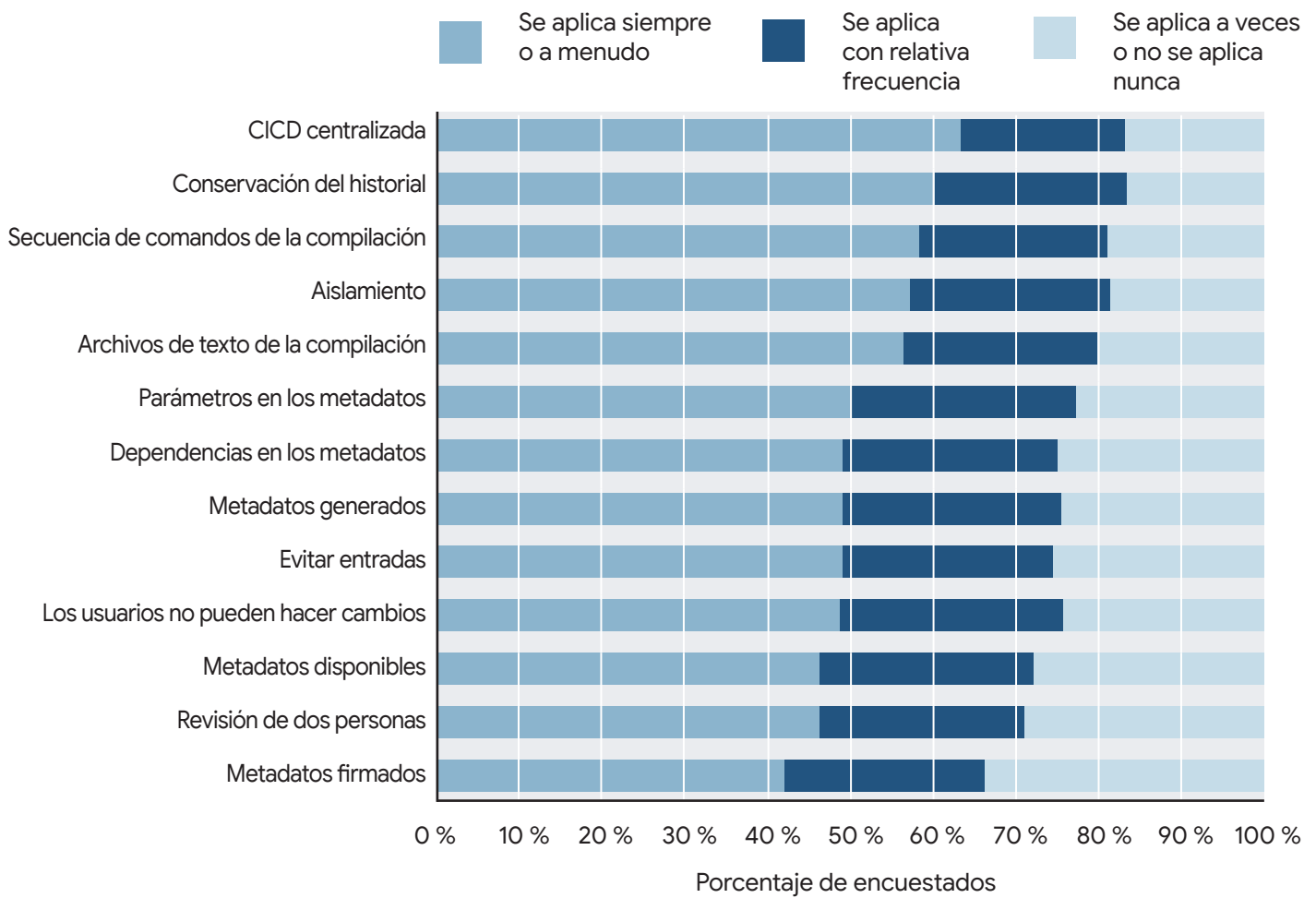


Figura 1. Aplicación de las prácticas de SLSA

Respuestas de la encuesta acerca de la aplicación de las prácticas de SLSA. La mayoría de los encuestados indicaron que todas estas prácticas se aplicaban en cierta medida, pero relativamente pocos afirmaron que se aplicaban "siempre".

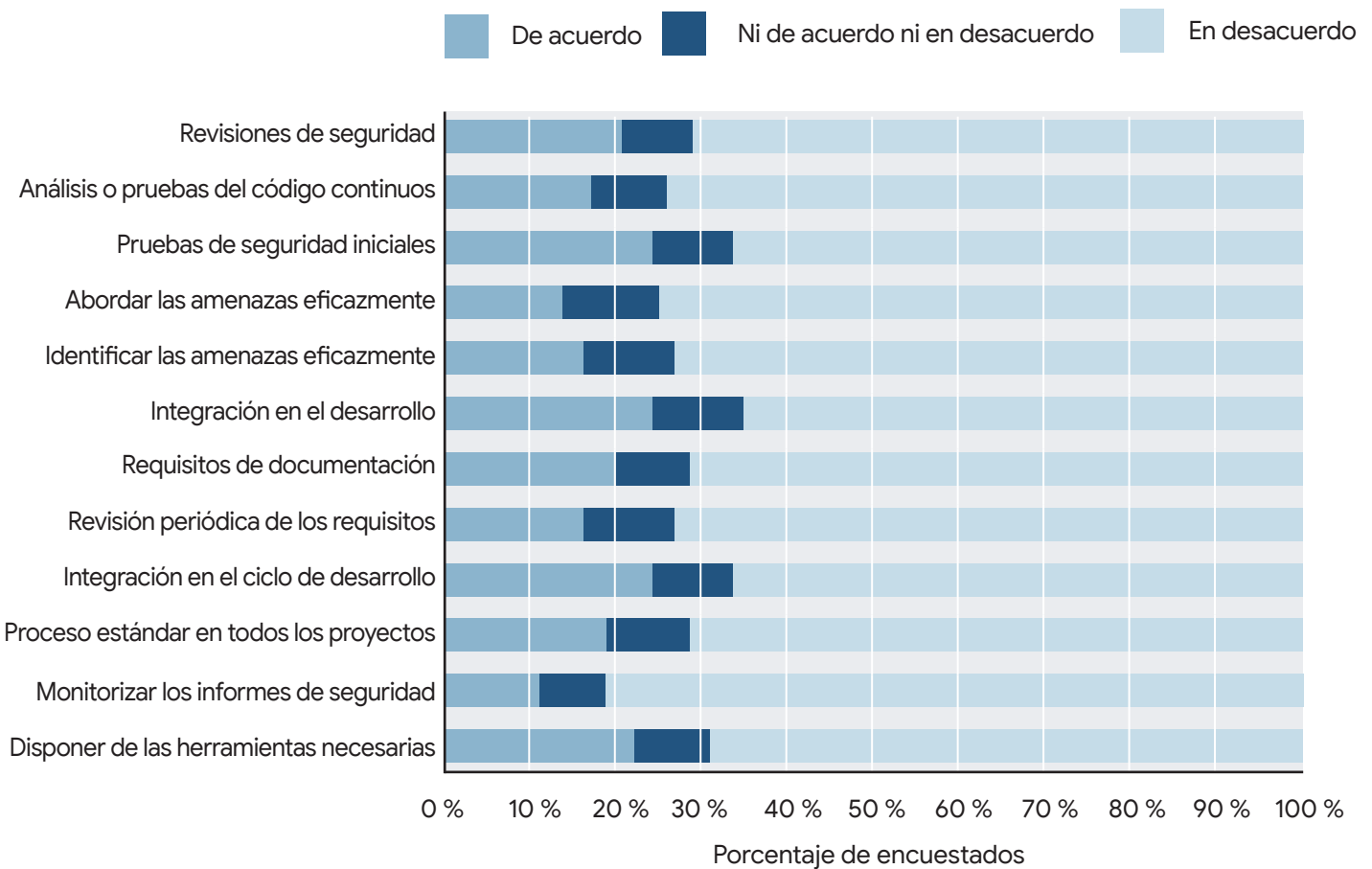


Figura 2. Aplicación de las prácticas del SSDL

Respuestas de la encuesta acerca de la aplicación de las prácticas del SSDL. Como en el caso de SLSA, la mayoría de los encuestados estaba de acuerdo con que su organización seguía todas estas prácticas.

¿Qué ayuda a las empresas a seguir buenas prácticas de seguridad?

La seguridad para aplicaciones es solo un aspecto del desarrollo de software y, por lo tanto, uno de los muchos factores a los que los desarrolladores tienen que dedicar tiempo y atención. Los sistemas de seguridad con mucha fricción pueden ser frustrantes para los desarrolladores e ineficaces en general, ya que se suelen evitar. Por ejemplo, en una serie de entrevistas de investigación con ingenieros de software profesionales, se observó que estos solo se comunicaban con los equipos de seguridad al comienzo o al final de los proyectos, y que no les resultaba fácil hablar con ellos. Según las palabras de uno de los participantes: "Tenemos un equipo de seguridad para aplicaciones, pero nunca han revisado mi código... Como la mayoría de ingenieros, suelo evitar tratar con ellos".

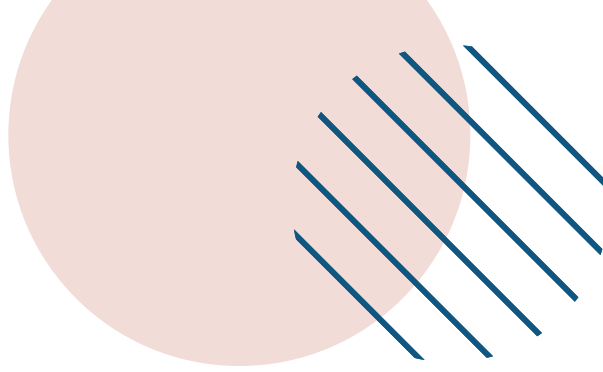
Una forma de mejorar la seguridad del software es reducir las dificultades que supone seguir las prácticas correspondientes. Los desarrolladores con los que hablamos querían hacer las cosas bien y a menudo mencionaban que les frustraba que enviar características o correcciones tenía siempre prioridad sobre abordar posibles problemas de seguridad. Por ejemplo, un encuestado en una encuesta de seguridad diferente explicaba que su mayor desafío de seguridad era "conseguir que la seguridad sea una prioridad. No interesa, no ayuda a vender el producto [y] no se considera importante hasta que hay un problema".

Los datos de nuestra encuesta nos autorizan a pensar que hay varios factores que pueden ayudar a los desarrolladores a "hacer las cosas de forma segura".

El factor más importante que observamos no era técnico, sino cultural: **las organizaciones que se acercaban más a la cultura "generativa" de Westrum solían señalar mucho más que habían aplicado a nivel general las prácticas de seguridad**, según se definen en el framework de SLSA¹. Las culturas generativas se distinguen por cooperar mucho, compartir riesgos y responsabilidades, y aprender de los errores cometidos. Según nuestra hipótesis, estos comportamientos se traducen en prácticas de seguridad más provechosas en muchos sentidos, como animar a los ingenieros de software a que sean más proactivos con la seguridad de la cadena de suministro, recompensar a los empleados por seguir las prácticas de seguridad independientemente de su puesto o reducir el riesgo de tener problemas de seguridad.

Desde el punto de vista tecnológico, tres de los factores que más refuerzan la seguridad tienen que ver con la infraestructura. Intuitivamente, esto tiene sentido: si la infraestructura facilita tareas como el análisis de vulnerabilidades o la revisión manual del código, es más probable que los ingenieros las pongan en práctica. Concretamente, observamos que **tener un sistema para realizar controles de fuente, integración continua y**

¹ Curiosamente, en general estos mismos encuestados no estaban más de acuerdo con las preguntas relativas al SSDF del NIST: Aunque los marcos de SLSA y del SSDF tratan diferentes aspectos de la seguridad del desarrollo de aplicaciones, esperábamos que las respuestas a ambos grupos de preguntas coincidieran. Como mencionamos antes, es posible que la escala del SSDF condujera más a marcar las respuestas de estar de acuerdo, lo cual explicaría esta diferencia.



entrega continua estaba relacionado con una aplicación más rigurosa de las prácticas de SLSA. Un factor clave es probablemente las situaciones en que los problemas de seguridad captan la atención de los desarrolladores; una encuesta distinta demostró que ocurría principalmente durante la CI. Normalmente, la CI precede a las revisiones del código y durante ese proceso se ejecutan las herramientas de análisis de vulnerabilidades y otro tipo de análisis, ya que garantiza que todas las confirmaciones de código estén sujetas a los mismos requisitos de seguridad. Si no existe un sistema de compilación centralizado, es mucho más difícil realizar esos análisis sistemáticamente y, al mismo tiempo, si no existe control del código fuente, es complicado tener un sistema de compilación centralizado en primer lugar.

Sin embargo, puede que el análisis de seguridad durante la CI/CD no se haga lo suficientemente pronto para los ingenieros de software. En una serie de entrevistas sobre seguridad realizada a desarrolladores de aplicaciones, estos coincidían en que realizar el análisis de seguridad desde su estación de trabajo les ahorraría tiempo y trabajo. Se citaron dos situaciones con frecuencia: 1) querer saber de antemano si estaban compilando código sobre una dependencia con vulnerabilidades conocidas para poder reevaluar el uso de esa dependencia antes de hacer la compilación, y 2) no tener que esperar tanto tiempo (a veces horas) con la CI solo para confirmar si los cambios han resuelto un problema de seguridad. En ambos casos, los ingenieros afirmaron que, aunque el "respaldo" de la CI era necesario, la capacidad de ejecutar las

mismas herramientas de seguridad localmente les ayudaría a trabajar más rápido y con más eficiencia.

Los factores culturales y tecnológicos mencionados anteriormente eran los mayores impulsores de seguridad, pero no los únicos. Otros factores importantes son:

- Modalidades de trabajo flexibles (por ejemplo, ¿ofrece la organización la posibilidad de trabajar desde casa?)
- Uso de la nube (ya sea pública o privada)
- Trabajar en una aplicación o servicio "nativos de la nube"
- Sentir que la empresa valora e invierte en el equipo
- Baja rotación en el equipo
- El tamaño de la organización; las organizaciones más grandes puntúan mejor en seguridad

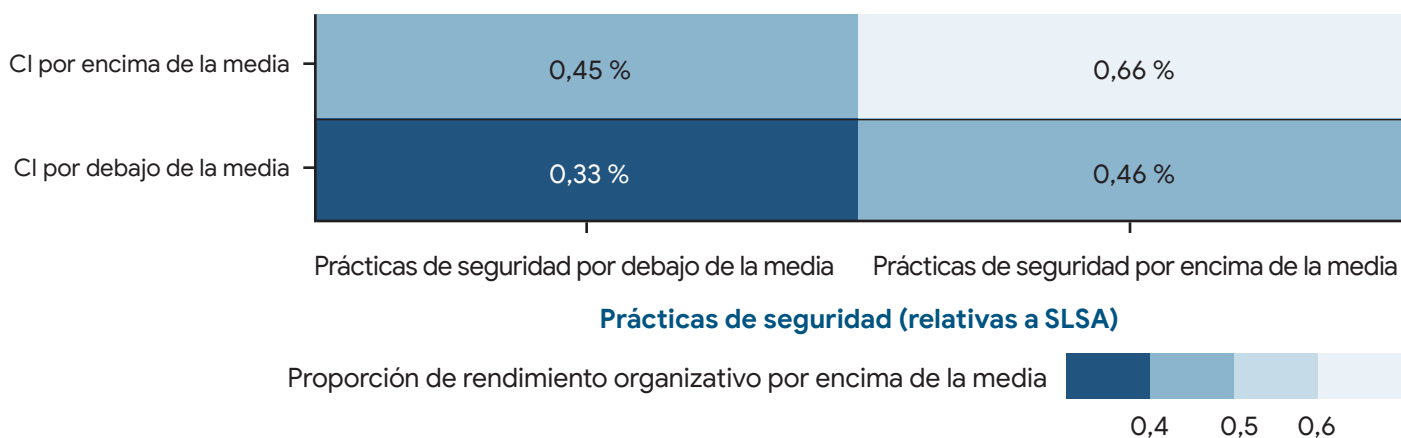
Sin embargo, estos factores parecen corresponderse sobre todo con la cultura generativa de Westrum (como tener modalidades de trabajo flexibles, sentir que la organización te valora o tener baja rotación en el equipo) o con el uso de CI/CD (como trabajar en una aplicación nativa de la nube o en una organización grande). Estos datos nos llevan a creer que la cultura de la organización y los procesos de desarrollo modernos (como la integración continua) son los factores que más contribuyen a la seguridad del desarrollo de aplicaciones en una organización y el mejor punto de entrada para las organizaciones que quieren mejorar su sistema de seguridad.

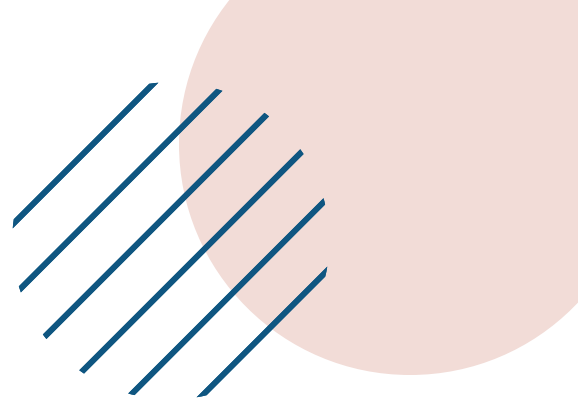
¿Qué resultados reportan las buenas prácticas de seguridad?

¿Qué ventajas pueden esperar obtener las organizaciones a medida que mejoran sus prácticas de seguridad en el desarrollo de software? Según los datos de nuestra encuesta, los encuestados anticipaban **una probabilidad menor de sufrir infracciones de seguridad, interrupciones del servicio y degradación del rendimiento a medida que las empresas consolidan la aplicación de las prácticas de seguridad de la cadena de suministro**. Asimismo, llevamos a cabo una investigación diferente durante la primera mitad del 2022 y observamos que ejecutar herramientas como el análisis de vulnerabilidades durante la CI aumentaba considerablemente la probabilidad de identificar vulnerabilidades en las dependencias del software: los encuestados que usaban dichas herramientas tenían casi el doble de probabilidades de identificar una vulnerabilidad de la seguridad en su propio código o en una de sus dependencias. En resumen, las prácticas de SLSA y SSDF parecen funcionar como está previsto. No afirmamos que puedan

eliminar las amenazas de seguridad, pero de nuestros datos se desprende que sí reducen los riesgos de seguridad de la organización.

Las prácticas de seguridad también pueden tener un efecto positivo en los resultados de rendimiento, pero hay una peculiaridad: la CI desempeña un papel clave. Si no se implementa la CI, las prácticas de seguridad no tienen ningún efecto en el rendimiento del envío de software. Pero si se implementa la CI, las prácticas de seguridad tienen un efecto positivo en el rendimiento del envío de software. Esto quiere decir esencialmente que la CI es necesaria para que las prácticas de seguridad tengan un efecto positivo en el rendimiento del envío de software. Además, las prácticas de seguridad suelen influir positivamente en el rendimiento organizativo y, cuando la CI está bien establecida, la repercusión es mayor. En la tabla que aparece a continuación se intenta reflejar este efecto.





Además de apreciar una reducción en los riesgos de seguridad, los encuestados también afirmaron notar **un menor desgaste** entre los integrantes del equipo y una mayor disposición a recomendar su organización **como un lugar de trabajo estupendo**. Ambos resultados confirman la naturaleza del "sí, y también" de la seguridad que consideran los ingenieros de software: es una cosa más que se acumula a su larga lista de tareas. Las herramientas y los procesos que les ayudan a incorporar prácticas de seguridad en su flujo de trabajo de desarrollo actual, en contraposición al trabajo imprevisto o a los "simulacros" cuando se descubre una amenaza, proporcionan un mecanismo para reducir los riesgos de seguridad y aumentan la satisfacción del desarrollador.

En conjunto, nuestros datos apuntan a que **los equipos sanos con alto rendimiento también suelen aplicar buenas prácticas de seguridad** a nivel general (aunque, como hemos mencionado antes, todavía se puede hacer más en este sentido). Puede que seguir metodologías como los frameworks de SLSA o del SSDF no mejore por sí solo todas las métricas de cultura y rendimiento que medimos, pero está claro que la seguridad no debe ir en detrimento de otras prioridades de desarrollo.

Las herramientas y los procesos que les ayudan a incorporar prácticas de seguridad en su flujo de trabajo de desarrollo actual, en contraposición con el trabajo imprevisto o a los "simulacros" cuando se descubre una amenaza, proporcionan un mecanismo para reducir los riesgos de seguridad y aumentan la satisfacción del desarrollador.

05

Sorpresas



Derek DeBellis

Aunque cada año el informe se centra en las respuestas a la encuestas que corresponden a ese año, hacemos lo posible por entender los datos en el contexto de todo el catálogo de informes de "State of DevOps" y de la investigación complementaria (como la investigación sobre el desgaste y la cultura). Poner a prueba la fiabilidad de estos efectos mediante tareas de replicación es un principio fundamental del programa de investigación. Esto nos ofrece la oportunidad de adaptar nuestras creencias a los datos y de comprender las tendencias emergentes o en evolución.

Este año tuvimos varias sorpresas que se explican por numerosas razones. Para empezar, este año la muestra incluía a más personas que se encontraban al principio de su carrera que en los informes previos. Una forma de interpretar este dato es que estamos recopilando más respuestas de personas que son directamente responsables de implantar prácticas y prestaciones técnicas, en contraposición con aquellas que

son responsables de supervisar o dirigir la implementación de esas prácticas. Otra posibilidad es que haya habido un cambio en el sector o en el mundo y que lo que funcionaba antes ya no tiene por qué funcionar en el futuro. Por ejemplo, las fuerzas macroeconómicas y, otro año más, la sombra de la pandemia del COVID-19, pueden haber transformado el panorama de DevOps. Por último, los cambios sutiles en nuestro modelo pueden haber transformado las relaciones entre las variables.¹



¹ El libro "Book of Why" de Judea Pearl y el trabajo de Robert McElreath, "Statistical Rethinking", nos ofrecen ejemplos increíbles de cómo lo que haces y no incluyes en tus modelos estadísticos puede afectar al resultado del modelo.

Los indicios inesperados o no sustentados por hipótesis ponen en un aprieto a los investigadores cuando tienen que redactar el informe. Dado que hay un riesgo de que el indicio sea falso o, como mínimo, que todavía tenga que establecer (o incluso contradiga) los resultados empíricos de varios estudios, lo responsable es hacer un seguimiento de la investigación para intentar replicar los resultados y entender su causa.² Al centrarse en las sorpresas, el investigador también corre el riesgo de subestimarla cantidad de efectos fiables que han surgido en años de investigación. En cada informe de "State of DevOps" llevamos a cabo un estudio estadístico de más de cien itinerarios. Al hacerlo, corremos el riesgo de obtener datos falsos solo por azar. Intentamos contrarrestar este efecto replicando el estudio anualmente.

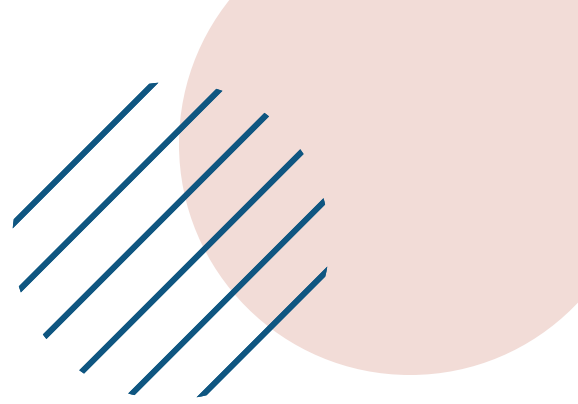
Por otro lado, no incluir tal resultado en el informe entraña el riesgo de crear un efecto de "archivador"³, que resulta en que lo esperado o lo aceptable se conozca y que lo inesperado o difícil de aceptar quede oculto. Buscamos un equilibrio: no queremos sensacionalizar resultados incipientes, pero también creemos que es crucial compartirllos. Estas son las cosas que más nos sorprendieron y lo que creemos que significan:

01 Hemos observado sistemáticamente que las prácticas del desarrollo basado en troncales tienen un efecto positivo en el rendimiento del envío de software. De hecho, este resultado aparece en el estudio todos los años desde el 2014. Las funciones del desarrollo basado en troncales exhibían un comportamiento fuera de lo normal este año. En primer lugar, estas funciones afectaban negativamente al rendimiento del envío de software. En informes de investigación anteriores se registraba el efecto contrario. Dado lo anómalo de este resultado, deseamos comprobar si el fenómeno se replica en los próximos estudios y averiguar si la comunidad puede explicarlo.

02 Observamos que el rendimiento del envío de software solo beneficia al rendimiento organizativo cuando el rendimiento operativo también es alto, y muchos de los encuestados no tenían un rendimiento operativo alto. Esto contradice las versiones anteriores de nuestro estudio, en las que era mucho más evidente la conexión entre el rendimiento del envío de software y el rendimiento organizativo.

² Kerr, N. L. (1998). HARKing: Hypothesizing after the results are known. *Personality and social psychology review*, 2(3), 196-217.

³ Rosenthal, R. (1979). The file drawer problem and tolerance for null results. *Psychological bulletin*, 86(3), 638.



03 Las prácticas de documentación tenían un impacto negativo en el rendimiento del envío de software. Este hallazgo contradice los informes anteriores. Una hipótesis es que cada vez se automatiza más la documentación, sobre todo entre los equipos con alto rendimiento. Hasta que no recopilamos más información, tenemos pocos datos que puedan respaldar o refutar esta idea.

04 Algunas prestaciones técnicas (por ejemplo, el desarrollo basado en troncales, la arquitectura con bajo acoplamiento, la CI o la CD) parecían predecir el desgaste. Como hemos mencionado anteriormente, en esta muestra había bastante más encuestados que se encontraban en el principio de su carrera que en las muestras de otros años. Por tanto, puede que hayamos estado encuestando a los individuos responsables de implementar la función, en lugar de a aquellos responsables de crear o supervisar la iniciativa. Es posible que implementarla sea bastante más complicado que supervisarla. Queremos seguir investigando para entender mejor este resultado.

05 Las prácticas de ingeniería de fiabilidad afectaban negativamente al rendimiento del envío de software. Una explicación sería que sus causas no están necesariamente interconectadas. En un nuevo análisis de grupo que realizamos este año (consulta "¿Cómo lo estáis haciendo?") observamos que un subgrupo de grupos parecía centrarse en la fiabilidad y no prestaba atención el rendimiento del envío de software. Nosotros creemos que ambos factores son independientes, en el sentido de que se puede abordar uno sin el otro, pero, al final, para que el rendimiento del envío de software contribuya al rendimiento organizativo, debe haber fiabilidad.

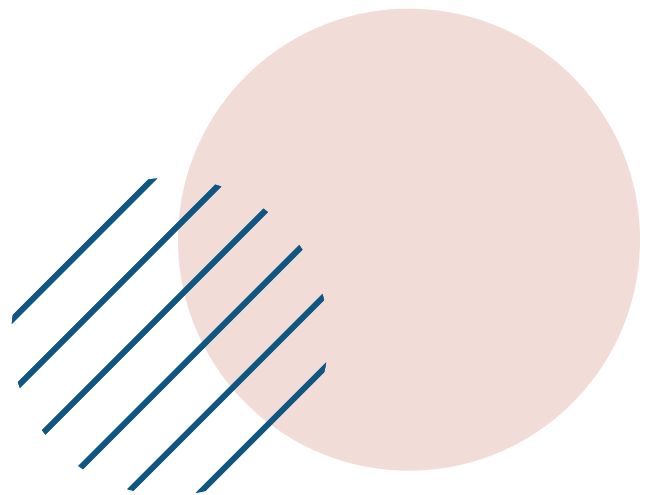
06 Añadimos las prácticas relacionadas con SLSA para averiguar si los equipos las adoptan para mantener una cadena de suministro del software segura. Aunque esperábamos que hubiera una relación entre la implementación de prácticas de seguridad y el rendimiento (por ejemplo, el uso de prestaciones técnicas, mejor rendimiento del envío de software y mejor rendimiento organizativo), nos sorprendió comprobar que las prácticas de seguridad eran en realidad el mecanismo mediante el cual las prestaciones técnicas influían en el rendimiento del envío de software y en el rendimiento organizativo.



La inclusión de las prácticas relacionadas con SLSA parecen explicar la mayor parte de los efectos de la integración continua, el control de versiones y la entrega continua, tanto en el rendimiento del envío de software como en el de la organización. Dicho de otra forma, existe una cadena causal que se detecta en los datos que explica que muchas prestaciones técnicas tienen un impacto positivo en las prácticas relacionadas con SLSA y que, a través de ese impacto, también influyen positivamente tanto en el rendimiento del envío de software como en el de la organización. Utilizamos el análisis de la mediación para detectar este resultado.⁴⁵ Esto nos conduce a seguir investigando si nuestra forma de medir las prácticas relacionadas SLSA está midiendo otras características del equipo (como el rendimiento general) y de qué manera contribuyen las prácticas de seguridad a mejorar el rendimiento del envío de software y el de la organización.

Contamos con volver a estudiar estos efectos el año que viene para comprobar si podemos reproducir y explicar estos nuevos patrones, o si deberíamos inclinarnos por considerarlos resultados atípicos (que también deberíamos intentar explicar). Como siempre, agradecemos los comentarios de la comunidad.

¡Únete a la [DORA Community](http://dora.community) (<http://dora.community>) para seguir debatiendo estas sorpresas y otros hallazgos del informe de este año!



⁴ Jung, Sun Jae. "Introduction to Mediation Analysis and Examples of Its Application to Real-world Data". Journal of preventive medicine and public health = Yebang Uihakhoe chi vol. 54,3 (2021): 166-172. doi:10.3961/jpmph.21.069

⁵ Carrión, Gabriel Cepeda, Christian Nitzl, and José L. Roldán. "Mediation analyses in partial least squares structural equation modeling: Guidelines and empirical examples". Partial least squares path modeling. Springer, Cham, 2017. 173-195.

06

Datos demográficos y emporiográficos



Derek DeBellis

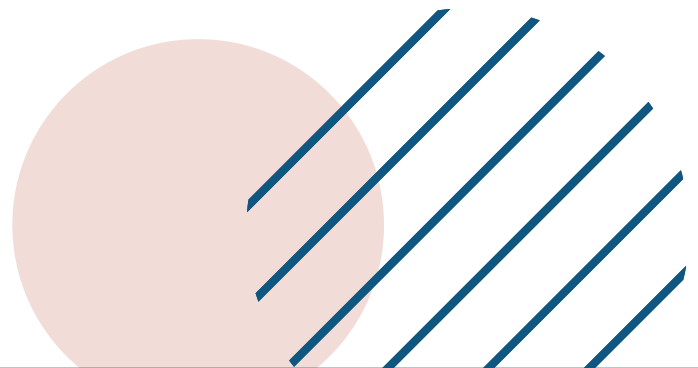
¡Gracias por contribuir a nuestra investigación y a nuestro sector!

¿Quién ha completado la encuesta?

Con más de ocho años de investigación y las respuestas de más de 33.000 profesionales del sector, el informe "State of DevOps" presenta las prácticas de DevOps y de desarrollo de software que contribuyen al éxito de los equipos y las organizaciones. Este año, más de 1350 profesionales de diversos sectores de todo el mundo compartieron sus experiencias para ayudarnos a conocer mejor los factores que mejoran el rendimiento. ¡Gracias por contribuir a nuestra investigación y a nuestro sector! En resumen, la representación de estadísticas demográficas y emporiográficas ha variado poquísimo.

Al igual que en años anteriores, hemos recopilado información demográfica de cada encuestado. Entre las categorías se incluyen las de sexo, discapacidad y grupos con poca representación.

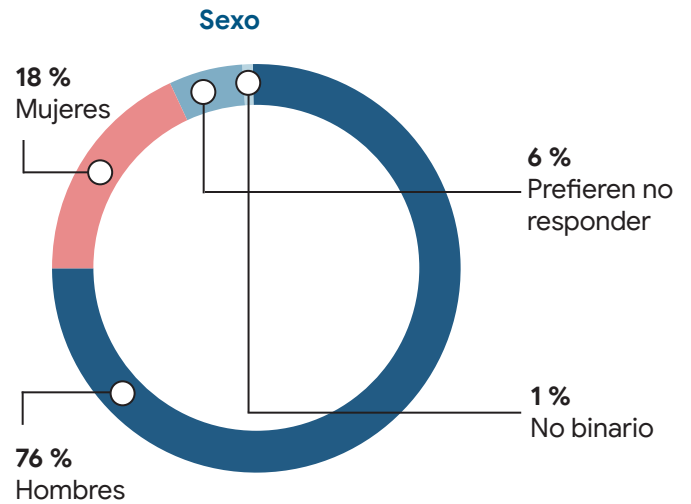
Este año, observamos una representación coherente con informes anteriores en cuanto a las categorías emporiográficas, como el tamaño, el sector y la zona de la empresa. De hecho, más del 60 % de los encuestados trabajan como ingenieros o supervisores, y una tercera parte trabaja en el sector tecnológico. Además, hay representación de los sectores de servicios financieros, minoristas y de empresas industriales y de fabricación.



Datos demográficos

Sexo

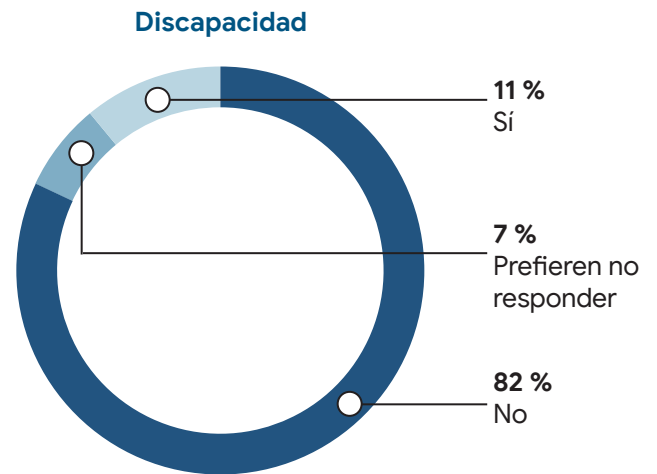
En comparación con el 2021, en la muestra de este año hay una mayor proporción de mujeres encuestadas (un 18 % frente a un 12 %). La proporción de hombres encuestados (un 76 %) es menor que en el 2021 (un 83 %). Los encuestados afirmaron que el 25 % de sus equipos son mujeres, igual que en el 2021 (un 25 %).



Porcentaje de mujeres: **mediana del 25 %**

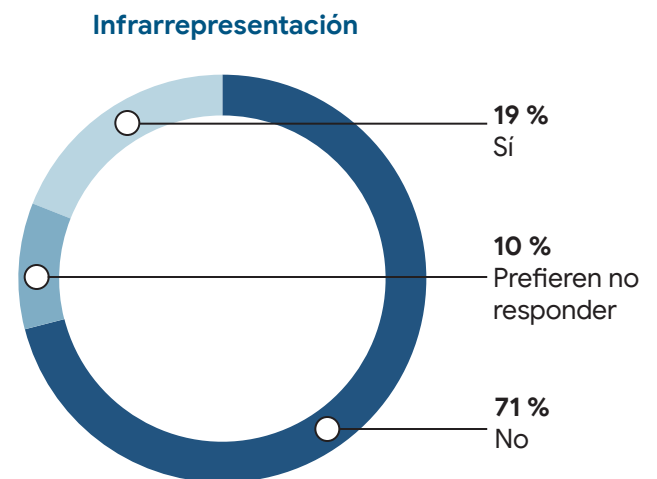
Discapacidad

Identificamos la discapacidad en seis dimensiones que siguen las pautas del [Washington Group Short Set](#). Este es el cuarto año en el que hemos preguntado sobre la discapacidad. El porcentaje de personas con discapacidades se mantuvo estable con respecto a nuestro informe del 2021 (un 11 %).



Infrarrepresentación

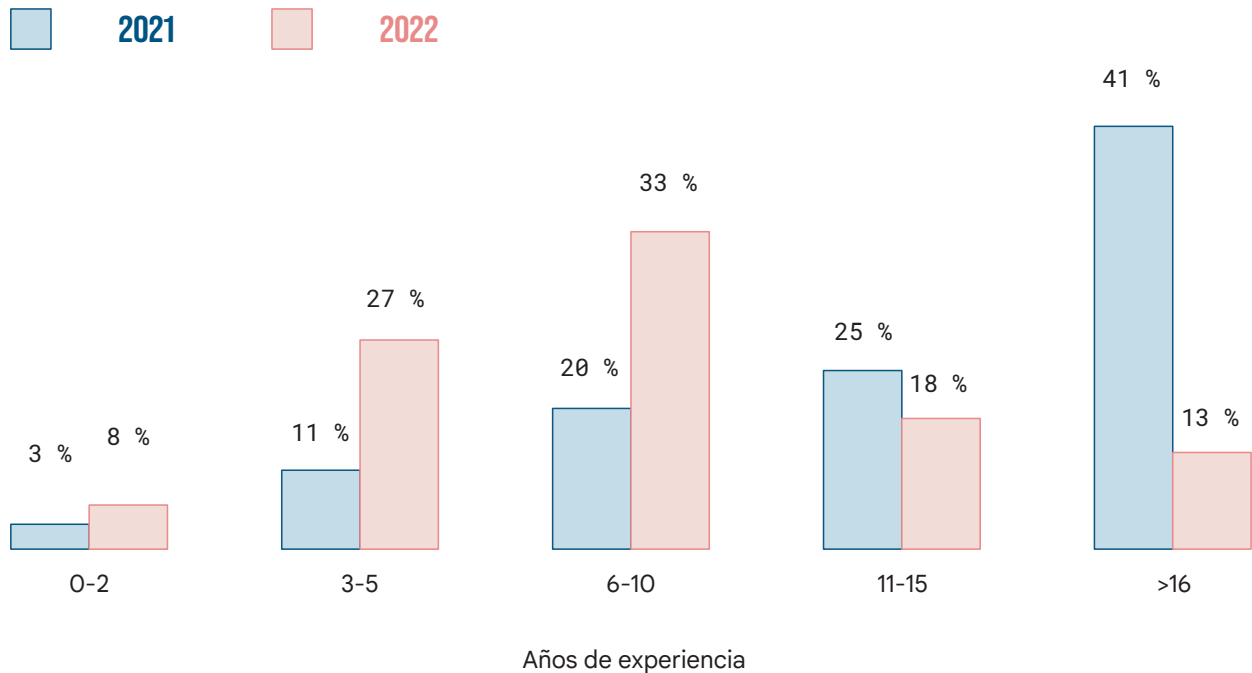
Identificarse como miembro de un grupo infrarrepresentado puede hacer referencia a la raza, sexo u otra característica. Este es el quinto año que hemos preguntado sobre la infrarrepresentación. El porcentaje de personas que se identifican como pertenecientes a un grupo infrarrepresentado ha aumentado ligeramente, del 17 % en el 2021 al 19 % en el 2022.

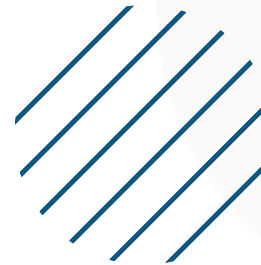


Años de experiencia

Cabe destacar que este año había más encuestados con cinco o menos años de experiencia (el 35 %) que en el 2021 (el 14 %). Por tanto, quizá no sorprende que la proporción de encuestados con más de 16 años de experiencia (el 13 %) fuera una fracción de la del 2021 (el 41 %).

Este cambio puede explicar algunos de los patrones que han asomado en los datos; creemos que es importante tenerlo en cuenta a la hora de interpretar los resultados, sobre todo al compararlos con los del año pasado.





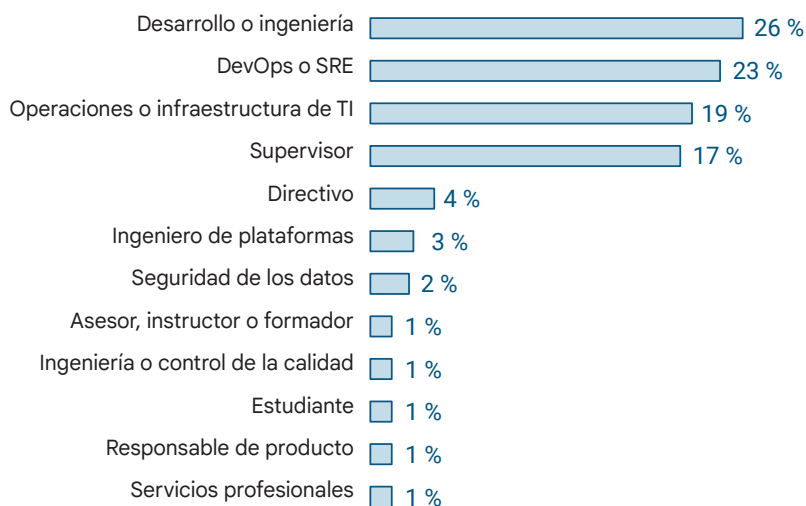
Datos emporgiográficos

Puesto

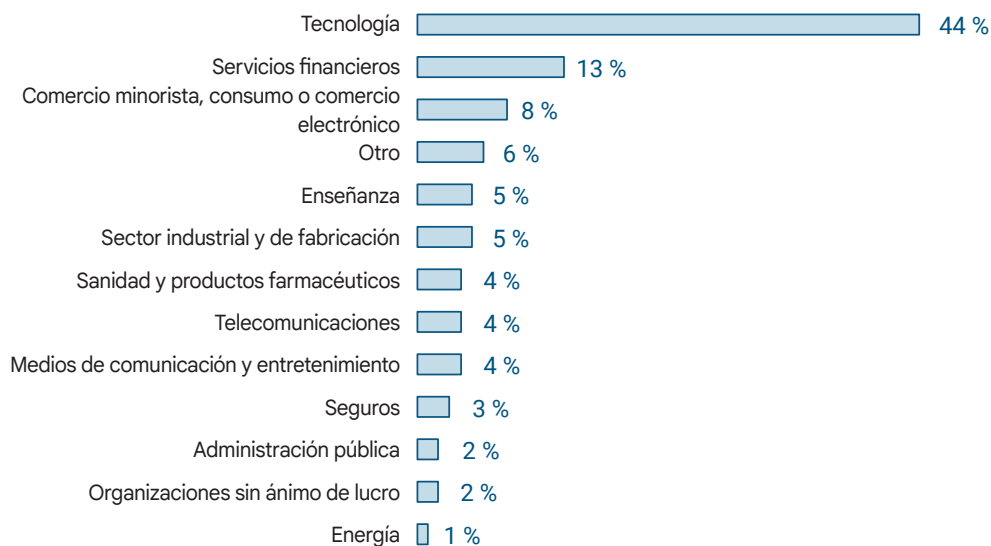
El 85 % de los encuestados trabajan en equipos de desarrollo o de ingeniería (el 26 %), en equipos de DevOps o SRE (el 23 %), en equipos de operaciones de TI o de infraestructura (el 19 %), o como supervisores (17 %). La proporción de encuestados que trabajan en equipos de operaciones de TI o de infraestructuras (el 19 %) es más del doble que el año pasado (el 9 %). Los directivos (del 9 % en el 2021 al 4 %) y los empleados de servicios profesionales (del 4 % el 2021 al 1 %) fueron los dos grupos que más disminuyeron en comparación con el año pasado.

Sector

Al igual que en los informes anteriores de "State of DevOps", observamos que la mayoría de los encuestados trabajan en el sector tecnológico, seguido por el de servicios financieros y el de comercio minorista.

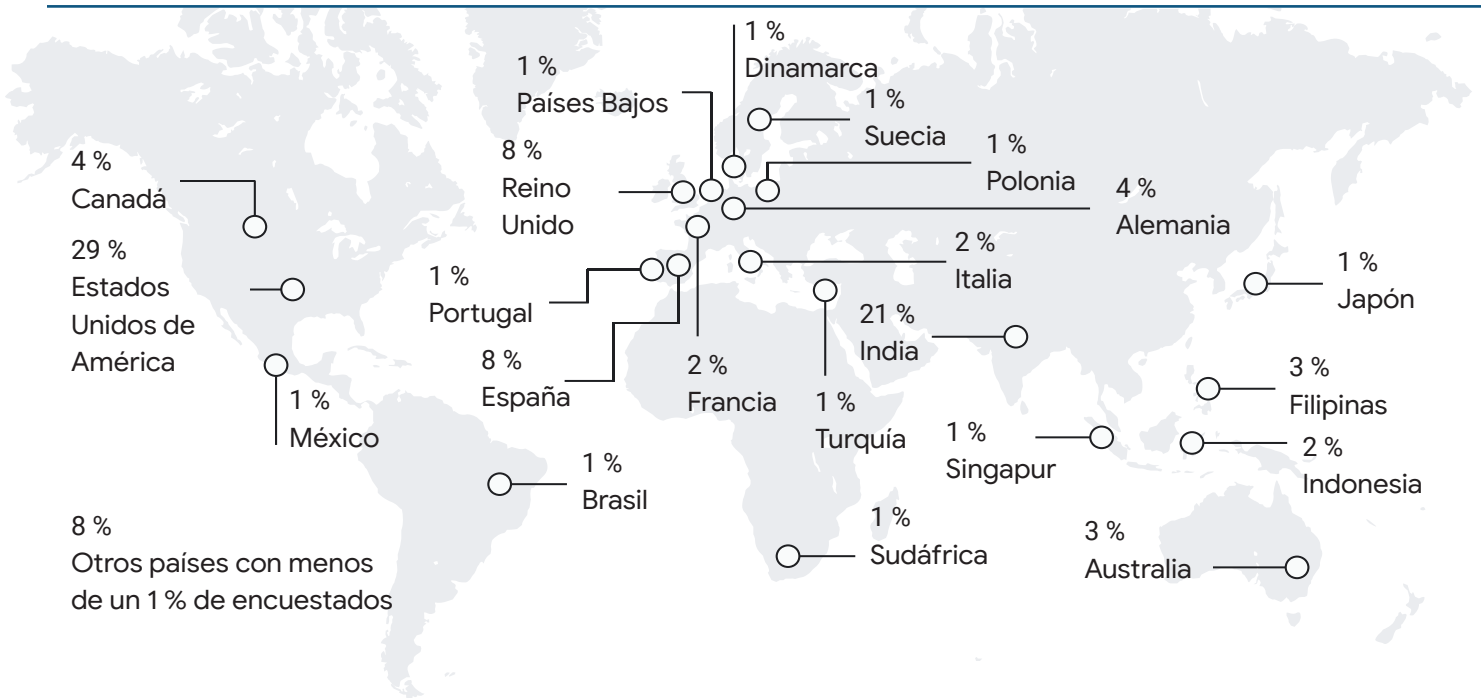


2022



País

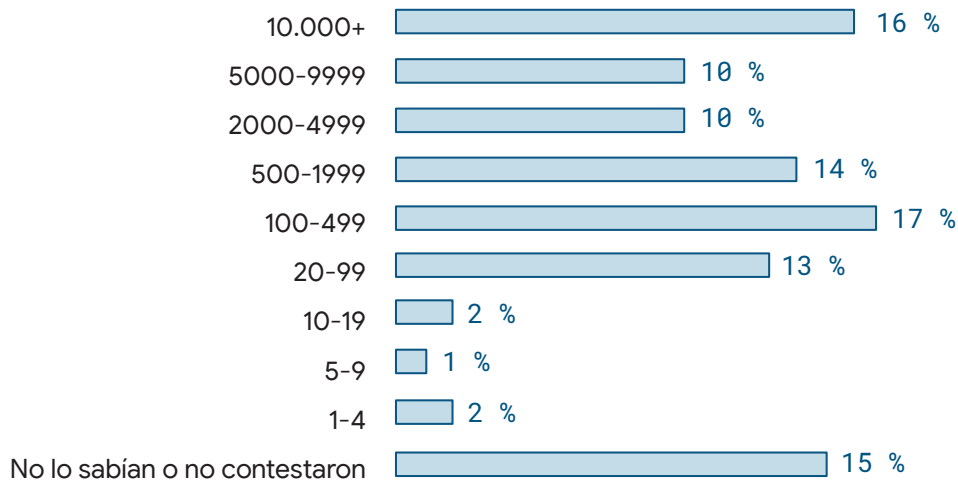
Este año pedimos a los encuestados que seleccionaran su país natal en lugar de su región. La región, que a menudo estaba representada por un continente, nos parecía demasiado general para entender la composición de los encuestados. Recibimos respuestas de participantes de más de 70 países; el 89 % de los encuestados procedían de 22 países.

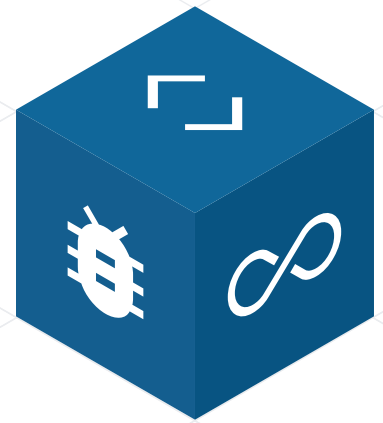


Número de empleados

Al igual que en las encuestas anteriores de "State of DevOps", los encuestados trabajan en organizaciones de diferentes tamaños. El 22 % de los encuestados trabaja en empresas con más de 10.000 empleados, y el 7 %, en empresas con entre 5000 y 9999 empleados. Otro 15 % de los encuestados trabajan en organizaciones con entre 2000 y 4999 empleados.

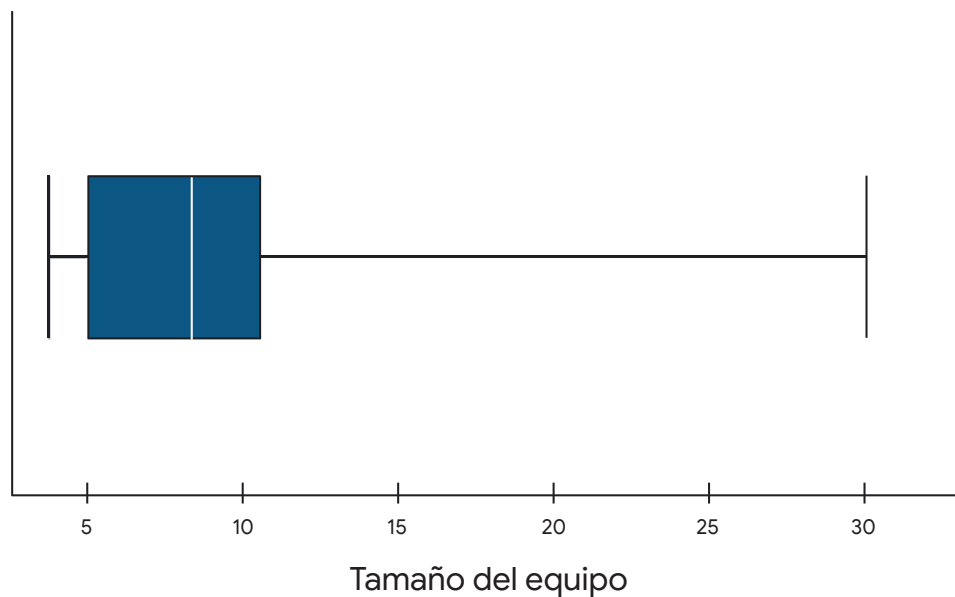
También observamos una representación equitativa de los encuestados que trabajan en organizaciones con entre 500 y 1999 empleados (un 13 %), entre 100 y 499 empleados (un 15 %) y, por último, entre 20 y 99 empleados (un 15 %). Este año también dimos la opción a los encuestados de responder "No lo sé" a la pregunta sobre el tamaño de su organización; el 15 % de encuestados indicaron que no lo sabían o no contestaron.





Tamaño del equipo

Este año pedimos a los encuestados que indicaran el tamaño aproximado de su equipo. El 25 % de los encuestados trabajaban en equipos de cinco o menos personas. El 50 % de los encuestados trabajaban en equipos de ocho o menos personas. El 75 % de los encuestados trabajaban en equipos de doce o menos personas.

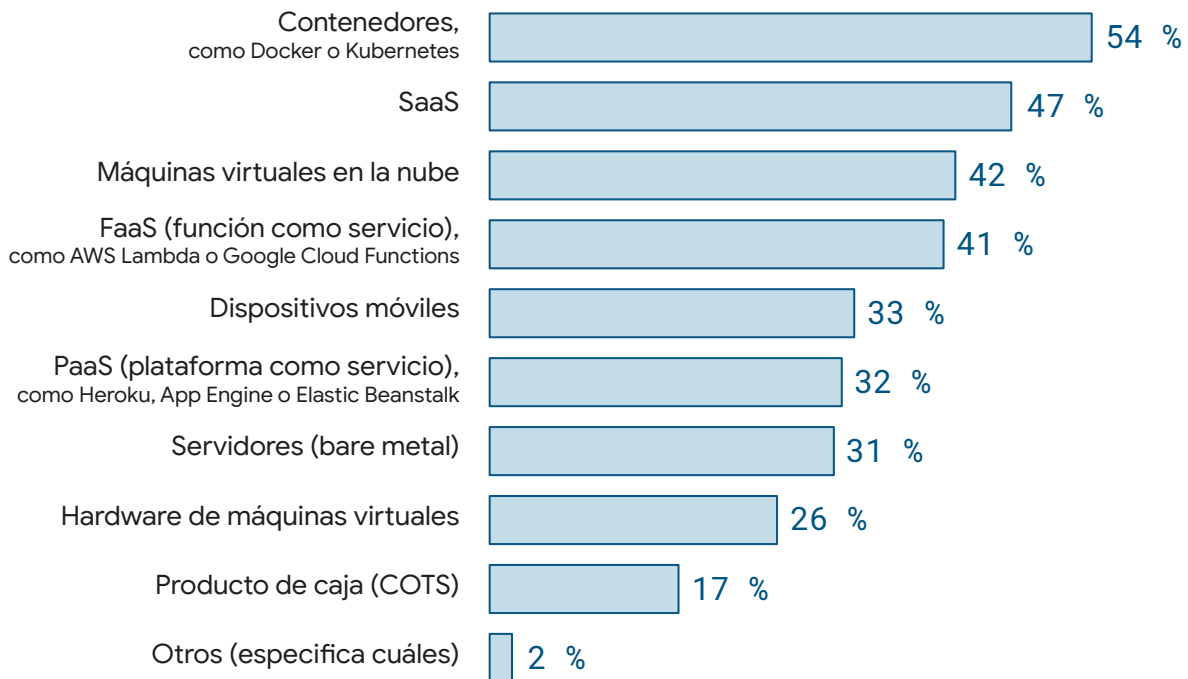


Destino de despliegue

El 2021 fue el primer año en el que decidimos preguntar a los encuestados dónde desplegaban su servicio o aplicación principal en los que trabajan. Para nuestra sorpresa, el principal destino de despliegue eran

contenedores. Este año no ha sido diferente, aunque con una menor proporción (el 54 %) que el año pasado (el 64 %).

También añadimos más opciones para ofrecer a los encuestados más posibilidades de reflejar dónde hacen despliegues.



07

Comentarios finales



Derek DeBellis

Cada año que elaboramos este informe nos esforzamos por ofrecer una representación rigurosa de cómo las prácticas y prestaciones producen resultados cruciales para las empresas, como el rendimiento organizativo. Estudiamos cómo se replican muchos de los efectos que aparecen en los informes anteriores y ampliamos nuestro estudio para incluir las prioridades emergentes del panorama de DevOps. Este año formulamos nuestra encuesta y nuestros análisis para analizar en profundidad las prácticas de seguridad; también modificamos nuestro sistema de modelos estadísticos para investigar la condicionalidad o las dependencias de determinados efectos. Asimismo, exploramos nuevas maneras de describir el panorama del envío de software y del rendimiento operativo.

En muchos sentidos, la narrativa que se materializó este año refleja la de años anteriores: las prestaciones técnicas se sustentan entre sí para mejorar el rendimiento, hay muchas ventajas inherentes en el uso de la nube, la cultura de oficina y la flexibilidad contribuyen a un mejor rendimiento organizativo y el desgaste de los empleados impide a las organizaciones alcanzar sus objetivos. El análisis de interacciones que añadimos explícitamente al modelo nos ayudó a entender las condiciones bajo las que se producen

determinados efectos. Por ejemplo, el rendimiento del envío de software solo influye positivamente en el rendimiento operativo (fiabilidad) si el rendimiento operativo es alto, lo que nos lleva a concluir que ambos factores son necesarios para que la organización prospere. También encontramos varias sorpresas, que señalamos en una sección específica.

Queremos agradecer a todos los que han contribuido a la encuesta de este año. Esperamos que nuestra investigación os ayude a ti y a tu organización a desarrollar mejores equipos y mejor software, así como a conciliar la vida laboral con la personal.

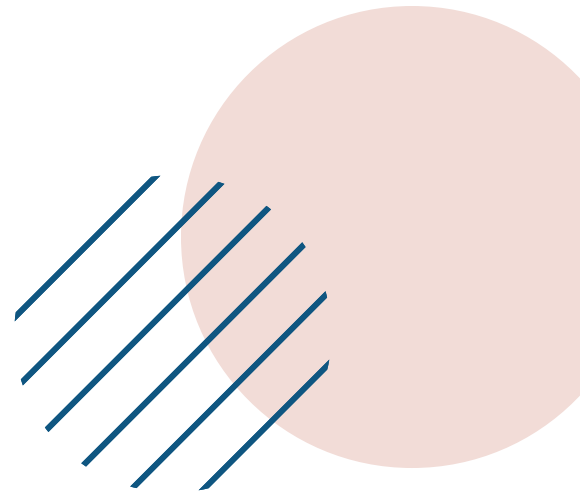


08

Agradecimientos

Una gran familia de apasionados colaboradores ha hecho posible el informe de este año. El diseño, análisis, redacción y revisión de las preguntas de la encuesta, así como el diseño del informe, son solo algunas de las formas en las que nuestros compañeros han ayudado a llevar a cabo esta gran labor. Los autores quieren dar las gracias a todas estas personas por sus contribuciones y asesoramiento en el informe de este año. Todos los apellidos aparecen por orden alfabético.

- | | |
|--------------------|------------------------|
| Scott Aucoin | Eric Maxwell |
| Alex Barrett | John Speed Meyers |
| James Brookbank | Steve McGhee |
| Kim Castillo | Jacinda Mein |
| Lolly Chessie | Alison Milligan |
| Jenna Dailey | Pablo Pérez Villanueva |
| Derek DeBellis | Claire Peters |
| Rob Edwards | Connor Poske |
| Dave Farley | Dave Stanke |
| Christopher Grant | Dustin Smith |
| Mahshad Haeri | Seth Vargo |
| Nathen Harvey | Daniella Villalba |
| Damith Karunaratne | Brenna Washington |
| Todd Kulesza | Kaiyuan "Frank" Xu |
| Amanda Lewis | Nicola Yap |
| Ian Lewis | |



09

Autores



Claire Peters

Claire Peters es investigadora de la experiencia de usuario en Google. Su trabajo abarca varios aspectos y expresiones de DORA en configuraciones aplicadas. Claire estudia el Cloud Application Modernization Program (CAMP) de Google, las interacciones con el cliente dirigidas por DORA y las herramientas basadas en las cuatro claves para ayudar a equipos e individuos a aplicar los principios de DORA con mayor eficacia en su trabajo diario. También es miembro del equipo de investigación principal de DORA y elabora la encuesta anual de DORA y el informe "State of DevOps". Tiene un máster en Análisis Cultural Aplicado por la Universidad de Copenhague.



Dave Farley

Dave Farley es el director ejecutivo y fundador de "Continuous Delivery Ltd", y el creador del canal de YouTube "Continuous Delivery". Es coautor del superventas "Continuous Delivery", así como autor del superventas "Modern Software Engineering: Doing What Works to Build Better Software Faster". Es coautor del "Reactive Manifesto" y ganador del premio Duke por su proyecto de código abierto "LMAX Disruptor". Dave es precursor de la entrega continua, formador de opinión y experto en CD, DevOps, desarrollo guiado por pruebas y diseño de software. Cuenta con una dilatada experiencia formando a equipos de gran rendimiento, encaminando organizaciones hacia el éxito y creando software excepcional. Descubre más datos sobre Dave en sus cuentas de [Twitter](#), [YouTube](#), [su blog](#) y [su sitio web](#).

Daniella Villalba



Daniella Villalba es investigadora de la experiencia de usuario y se entrega por completo al proyecto DORA. Se centra en identificar los factores que hacen que los desarrolladores estén satisfechos y sean productivos. Antes de trabajar en Google, Daniella estudió las ventajas de la formación en meditación, los factores psicosociales que afectan a las experiencias de los estudiantes universitarios, la memoria de los testigos y las confesiones falsas. Tiene un doctorado en Psicología Experimental por la Universidad Internacional de Florida.

Dave Stanke



Dave Stanke es ingeniero de relaciones con los desarrolladores en Google. Asesora a los clientes sobre las prácticas recomendadas para adoptar DevOps y SRE. A lo largo de su carrera ha ocupado todo tipo de puestos, como el de director de tecnología en una empresa emergente, responsable de producto, servicio de asistencia, desarrollador de software, administrador de sistemas y diseñador gráfico. Tiene un máster en Gestión Tecnológica por la Universidad de Columbia.

Derek DeBellis



Derek DeBellis es investigador de la experiencia de usuario cuantitativa en Google, donde lleva a cabo investigaciones mediante encuestas, analiza registros y estudia formas de medir conceptos clave en el desarrollo de productos. Derek ha publicado recientemente acerca de la interacción entre los humanos y la inteligencia artificial, el impacto del COVID-19 en el abandono del tabaquismo, el diseño para abordar los errores de PLN y el papel de la experiencia de usuario en los debates sobre la privacidad.

Eric Maxwell



Eric Maxwell dirige las prácticas de transformación digital de DevOps en Google, donde asesora a las mejores empresas del mundo sobre cómo mejorar aún más, poco a poco, de forma continua. Trabajó como ingeniero durante la primera mitad de su carrera, automatizando de todo y abogando por la empatía hacia otros profesionales. Colaboró en la creación del Cloud Application Modernization Program (CAMP) de Google. Además, es miembro del equipo central de DORA y autor de la DevOps Enterprise Guidebook. Antes de trabajar en Google, Eric estuvo cocinando maravillas con otros tios ingeniosos en Chef Software.

John Speed Meyers



John Speed Meyers es científico de datos de seguridad en Chainguard, una empresa emergente de seguridad de la cadena de suministro de software. Los proyectos de investigación de John abarcan temas como la seguridad de la cadena de suministro de software, la seguridad del software libre y la respuesta internacional ante el creciente poder militar de China. Anteriormente, trabajó en In-Q-Tel, la RAND Corporation y el Center for Strategic and Budgetary Assessments. John tiene un doctorado en Análisis de Políticas por la Escuela de Postgrado Pardee RAND, un máster en Asuntos Públicos (MPA) por la Escuela de Asuntos Públicos e Internacionales de Princeton y un grado en Relaciones Internacionales por la Universidad Tufts.

Kaiyuan "Frank" Xu



Kaiyuan "Frank" Xu es investigador de la experiencia de usuario cuantitativa en Google. Analiza datos de registro y de encuestas para interpretar los patrones de uso y los comentarios de usuarios acerca de los productos de Google Cloud y, así, mejorar la excelencia de los productos para desarrolladores. Antes de Google, Kaiyuan pasó años realizando estudios de usuario cualitativos y cuantitativos en Microsoft para los productos de Azure y Power Platform. Tiene un máster en Diseño Centrado en Humanos e Ingeniería por la Universidad de Washington.

Nathen Harvey



Nathen Harvey es ingeniero de relaciones con los desarrolladores en Google. A lo largo de su carrera, ha ayudado a desatar el potencial de diversos equipos y a adaptar la tecnología a los resultados empresariales. Ha tenido el privilegio de trabajar con algunos de los mejores equipos y comunidades de software libre, ayudándoles a aplicar los principios y las prácticas de DevOps y SRE. Nathen revisó y contribuyó a la publicación de *"97 Things Every Cloud Engineer Should Know"*, O'Reilly 2020.

Todd Kulesza



Todd Kulesza es investigador de la experiencia de usuario en Google, donde estudia cómo trabajan actualmente los ingenieros de software y cómo podrían trabajar mejor en el futuro. Tiene un doctorado en Informática por la Universidad Estatal de Oregón.

10

Metodología

Diseño de la investigación

El estudio emplea un diseño transversal de talante teórico, conocido como **predictivo inferencial**, uno de los tipos de diseño más habituales en la investigación empresarial y tecnológica de hoy en día. El diseño predictivo inferencial se usa cuando no es posible o práctico realizar un diseño totalmente experimental.

Población objetivo y muestreo

La población objetivo de esta encuesta eran profesionales y líderes que trabajan en el sector de la tecnología y las transformaciones, o bien en ámbitos estrechamente relacionados, sobre todo los que estaban familiarizados con DevOps. Promocionamos la encuesta mediante listas de correo electrónico, promociones online, un panel digital y redes sociales, y pedimos a los usuarios que compartieran la encuesta con sus contactos (es decir, muestreo por bola de nieve).

Crear estructuras latentes

Formulamos nuestras hipótesis y conceptos a partir de ideas previamente validadas siempre que nos fue posible. Desarrollamos nuevos conceptos basados en la teoría, definiciones y la opinión de expertos. A continuación, tomamos medidas adicionales para aclarar nuestra intención de asegurarnos de que los datos recopilados en la encuesta tuvieran una alta probabilidad de ser fiables y válidos.¹

Cómo calculamos las diferencias entre los equipos de alto y bajo rendimiento

En la sección "¿Cómo lo estáis haciendo?" comparamos los equipos de alto y bajo rendimiento según las cuatro métricas de rendimiento del envío de software. El método es muy sencillo. Tomemos la frecuencia de despliegue como ejemplo. El grupo de alto rendimiento despliega bajo demanda (por ejemplo, varias veces al día). Si de media realizan cuatro despliegues al día, eso supone 1460 despliegues al año (4×365). Por el contrario, los equipos de bajo rendimiento despliegan entre una vez al mes y una vez cada seis meses, con una frecuencia media de despliegue de una vez cada 3,5 meses, y un promedio de frecuencia de despliegue de unos 3,4 despliegues al año ($12 \div 3,5$). Cogemos el ratio (1460 despliegues en el grupo de alto rendimiento \div 3,4 despliegues en el grupo de bajo rendimiento) y nos salen 417 veces. Este método se ha generalizado en las otras métricas de rendimiento del desarrollo.

¹ Churchill Jr, G. A. "A paradigm for developing better measures of marketing constructs", Journal of Marketing Research 16:1, (1979), 64–73.

Métodos de análisis estadístico

Análisis de grupos

En ambas técnicas de agrupación reflejadas en la sección "¿Cómo lo estáis haciendo?", usamos la agrupación jerárquica con una versión del método aglomerativo de Ward² para evaluar cómo se ajustan las diferentes técnicas de agrupación a los datos.

En los primeros resultados de agrupamiento que presentamos, buscamos grupos de respuestas en los factores de frecuencia de despliegue, tiempo para restaurar el servicio e índice de fallos por cambios. Este año hemos encontrado tres grupos tras evaluar 14 técnicas de agrupación jerárquica, usando 30 índices diferentes para determinar el número de grupos.³

El segundo análisis de grupos que presentamos seguía la misma metodología que el primero, pero lo desplegamos en diferentes dimensiones de los datos.

Queríamos encontrar patrones comunes de respuestas (por ejemplo, grupos) en los factores de rendimiento (una combinación de la frecuencia de despliegue y el plazo de entrega), rendimiento operativo (fiabilidad) y estabilidad (una combinación del tiempo para restaurar el servicio y el índice de fallos por cambios). También probamos distintos algoritmos de agrupación para comprobar lo sensibles que eran nuestros resultados a nuestra metodología. Aunque no existe una forma establecida de cuantificar dicha sensibilidad (que sepamos), los grupos que aparecieron solían compartir características.

Modelo de medición

Antes de llevar a cabo nuestro análisis, identificamos conceptos que usaban el análisis de factores exploratorios con el análisis de componentes mediante la rotación varimax.⁴ Confirmamos las pruebas estadísticas para la validez y fiabilidad convergentes y divergentes mediante la varianza media extraída (AVE), la correlación, el valor alfa de Cronbach⁵, ρA^6_A , el criterio heterotrait-monotrait ratio^{5,7} y la fiabilidad compuesta.

² Murtagh, Fionn, and Pierre Legendre. "Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion?". *Journal of classification* 31.3 (2014): 274-295.

³ Charrad M., Ghazzali N., Boiteau V., Niknafs A. (2014). "NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set.", *Journal of Statistical Software*, 61(6), 1-36., "URL <http://www.jstatsoft.org/v61/i06/>"

⁴ Straub, D., Boudreau, M. C., & Gefen, D. (2004). Validation guidelines for IS positivist research. *Communications of the Association for Information systems*, 13(1), 24.

⁵ Nunnally, J.C. *Psychometric Theory*. Nueva York: McGraw-Hill, 1978

⁶ Hair Jr, Joseph F., et al. "Partial least squares structural equation modeling (PLS-SEM) using R: A workbook". (2021): 197.

⁷ Brown, Timothy A., and Michael T. Moore. "Confirmatory factor analysis". *Handbook of structural equation modeling* 361 (2012): 379.

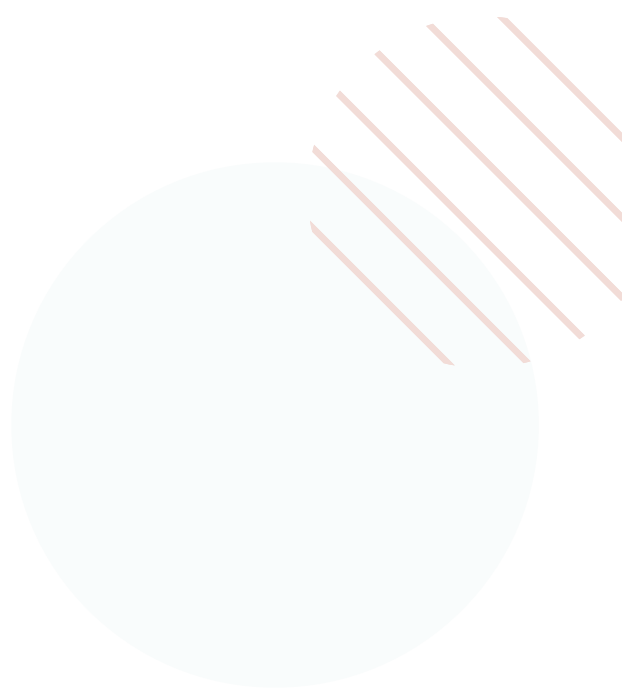


Modelización de ecuaciones estructurales

Probamos los modelos de ecuaciones estructurales (SEM) mediante el análisis de cuadrados parciales mínimos (PLS)⁸, que es un método modelo de ecuación estructural basado en la correlación.

Análisis del segundo modelo de agrupación

Para entender qué predice la pertenencia a un grupo, usamos la regresión logística multinomial.⁹ Usamos este sistema porque tratábamos de predecir la pertenencia a un grupo, que, en este caso, se trata de un conjunto de datos categóricos sin ordenar con más de dos niveles. Para entender los resultados que predecían la pertenencia a los grupos, usamos una regresión lineal en cada resultado (desgaste, trabajo imprevisto y rendimiento organizativo).



⁸ Hair Jr, J. F., Hult, G. T. M., Ringle, C. M., & Sarstedt, M. (2021). "A primer on partial least squares structural equation modeling (PLS-SEM)". Sage publications.

⁹ Ripley, Brian, William Venables, and Maintainer Brian Ripley. "Package 'nnet'". R package version 7.3-12 (2016): 700.

11

Más información

Únete a la DORA Community para debatir, aprender y colaborar a fin de mejorar el rendimiento del envío de software y de las operaciones

<http://dora.community>

Más información sobre las cuatro métricas clave

<https://goo.gle/four-keys>

Más información sobre las prestaciones de DevOps

<https://goo.gle/devops-capabilities>

Más información sobre cómo implantar las prácticas de DORA en tu organización con nuestra Enterprise Guidebook

<https://goo.gle/enterprise-guidebook>

Encuentra recursos sobre Site Reliability Engineering (SRE)

<https://sre.google>

<https://goo.gle/enterprise-roadmap-sre>

Completa la encuesta breve de DevOps

<https://goo.gle/devops-quickcheck>

Descubre el programa de investigación de DevOps

<https://goo.gle/devops-research>

Consulta qué otras empresas han implantado las prácticas de DORA en el libro electrónico ganador de los Premios Google Cloud DevOps

<https://goo.gle/devops-awards>

Más información sobre el Google Cloud Application Modernization Program (CAMP)


<https://goo.gle/3daLa9s>

Aprovechar los datos y las métricas de DORA para transformar los procesos tecnológicos

<https://goo.gle/3Doh8Km>

Lee el informe "The ROI of DevOps Transformation: How to quantify the impact of your modernization initiatives", de Forsgren, N., Humble, J., & Kim, G. (2018).

<https://goo.gle/3qECllh>



Lee el libro *Accelerate: The science behind devops: Building and scaling high performing technology organizations*. IT Revolution.

<https://itrevolution.com/book/accelerate>

Más información sobre el framework Supply chain Levels for Secure Artifacts (SLSA)

<https://slsa.dev>

Más información sobre el Secure Software Development Framework (NIST SSDF)

<https://goo.gle/3qBXLWk>

Más información sobre la cultura de DevOps: cultura empresarial de Westrum

<https://goo.gle/3xq7KBV>

Más información sobre la Open Source Security Foundation <https://openssf.org/>

Más información sobre in-toto

<https://in-toto.io/>

Más información sobre el Software Bill of Materials de NTIA.gov <https://www.ntia.gov/SBOM>

Ciberseguridad: respuesta federal a los incidentes de SolarWinds y Microsoft Exchange

<https://www.gao.gov/products/gao-22-104746>

Más información sobre el análisis de seguridad a nivel de aplicación como parte de la CI/CD

<https://go.dev/blog/survey2022-q2-results#security>

Por último, consulta los informes anteriores de "State of DevOps"; puedes encontrarlos todos en

<https://goo.gle/dora-sodrs>:

[Informe "Accelerate State of DevOps" del 2014](#)

[Informe "Accelerate State of DevOps" del 2015](#)

[Informe "Accelerate State of DevOps" del 2016](#)

[Informe "Accelerate State of DevOps" del 2017](#)

[Informe "Accelerate State of DevOps" del 2018](#)

[Informe "Accelerate State of DevOps" del 2019](#)

[Informe "Accelerate State of DevOps" del 2021](#)

