

# DevOps y la Medición de la Calidad del Producto de Software: Hallazgos Preliminares

Gabriel Alberto García-Mireles<sup>1</sup>, Nerina Peña Olivero<sup>2</sup>, Himer Avila-George<sup>2</sup>

[gabriel.garcia@unison.mx](mailto:gabriel.garcia@unison.mx); [nerinapolivero@gmail.com](mailto:nerinapolivero@gmail.com); [himer.avila@academicos.udg.mx](mailto:himer.avila@academicos.udg.mx)

<sup>1</sup> Departamento de Matemáticas, Universidad de Sonora, C. P. 83000, Hermosillo, México.

<sup>2</sup> Centro Universitario de los Valles, Universidad de Guadalajara, C. P. 46600, Guadalajara, México.

DOI: [10.17013/risti.53.37-52](https://doi.org/10.17013/risti.53.37-52)

**Resumen:** Las organizaciones de desarrollo de software están adoptando el enfoque DevOps como una estrategia para mantenerse competitivas. El impacto de estas iniciativas se relaciona con la reducción del tiempo de desarrollo mientras se mantiene la calidad del software. Sin embargo, pocos trabajos han señalado específicamente que aspectos de la calidad se han medido. En este trabajo, presentamos los resultados de un mapeo sistemático de literatura en donde identificamos las características de calidad del producto de software que se han estudiado en entornos DevOps. Los resultados se complementan con las iniciativas de medición del desempeño general de DevOps y aquellas que miden la calidad del producto de software. Dada la falta de trabajos en esta área, como primer paso en la definición de un proceso de medición de la calidad del producto en entornos DevOps, identificamos algunas métricas de mantenibilidad y factores contextuales por considerar en estas iniciativas.

**Palabras-clave:** DevOps, características de calidad de producto de software, medición, mantenibilidad.

## *DevOps and Software Product Quality Measurement: Preliminary Findings*

**Abstract:** Software development organizations are adopting the DevOps approach as a strategy to stay competitive. The impact of these initiatives is related to reducing development time while maintaining software quality. However, little research work has specifically indicated which aspects of quality have been measured. In this paper, we present the results of a systematic mapping study where we identify software product quality characteristics that have been studied in DevOps settings. The results are complemented by overall DevOps performance measurement initiatives and those that measure software product quality. Given the lack of research work in this area, as a first step in defining a product quality measurement process in DevOps settings, we identified some maintainability metrics and contextual factors to consider in these initiatives.

**Keywords:** DevOps, software product quality characteristics, measures, maintainability.

## 1. Introducción

El software impulsa la innovación y competitividad no solo de las industrias de las tecnologías de la información, sino de todos los sectores (Forsgren y Kernsten, 2018). Para muchas organizaciones, la adopción de enfoques ágiles y de DevOps se convierte en una estrategia para mantenerse competitivas (Forsgren y Kernsten, 2018). DevOps introduce un cambio organizacional en donde los grupos, tradicionalmente separados, de desarrollo y operaciones trabajan en entregas continuas del software (Debois, 2008; Ebert et al., 2016).

Los beneficios percibidos de la implementación de iniciativas basadas en DevOps reportan la reducción en el tiempo de entrega, menor número de defectos en los productos de software liberados, mejoras en el control de versiones, tiempo de comercialización más rápidos, mejoras en la productividad y mejora de la calidad del software, entre otras (Lazuardi et al., 2021). En una encuesta a desarrolladores de software, Offerman et al. (2022) reportan que los aspectos con mayor impacto positivo de la implementación de las prácticas de DevOps son la reducción del ciclo de desarrollo del producto, incremento en la colaboración, incremento en la efectividad del proceso de desarrollo y la detección temprana de defectos.

Aunque el impacto de las iniciativas DevOps se reporta positivo en diferentes indicadores organizacionales, el tema de la calidad del software se ha considerado desde la perspectiva del proceso (García-Mireles et al., 2015) dado que la calidad se mide en términos de la cantidad de defectos o fallas encontradas en las iniciativas DevOps que miden su impacto (Offerman et al., 2022; Visser y Hulstijn, 2023; Diamantopoulos, 2020; Bermon et al., 2022). Algunos estudios han señalado que la calidad del producto de software se ha estudiado poco y que existe la necesidad de abordarla en entornos DevOps (Domínguez-Acosta y García-Mireles, 2021; Mishra y Otaiwi, 2020). En particular, se recomienda estudiar el proceso de medición en las diferentes etapas de DevOps y cómo se podrían medir características de calidad como la usabilidad, eficiencia, mantenibilidad, portabilidad y seguridad (Mishra y Otaiwi, 2020).

Considerando lo expuesto anteriormente, el objetivo de este mapeo sistemático de literatura es conocer el estado actual de DevOps y la calidad del software, además identificar prácticas de DevOps informadas en la literatura que influyan específicamente en las características de calidad de producto que se encuentran representadas en el modelo de calidad de la norma ISO/IEC 25010 (ISO, 2011). Los resultados preliminares de este mapeo se reportaron en (Peña, Ávila & García-Mireles, 2023) en donde se describen las prácticas de DevOps que podrían estar correlacionadas con la calidad del producto de software y las características de calidad del software abordadas. Esta versión actualizada se centra en explorar cómo se realiza la medición en entornos DevOps y qué indicadores se recomiendan. Respecto de la calidad del producto de software se encontró poca evidencia de su medición. Así que, la investigación abordó la característica de mantenibilidad del producto de software para identificar cómo podría medirse en entornos DevOps. Sin embargo, la escasez de trabajos publicados en este tema llevó a recopilar un conjunto de métricas de mantenibilidad que se han aplicado en otros contextos como un primer paso para establecer un proceso de medición de la calidad del producto considerando el modelo ISO/IEC 25010 (ISO, 2011).

La estructura del documento es como sigue. En la sección 2 se presenta brevemente las principales revisiones de literatura que abordan la calidad del software en entornos DevOps. La sección 3 muestra los aspectos principales del proceso de estudio de mapeo sistemático realizado y cómo el estudio exploratorio permitió identificar algunas métricas de mantenibilidad que podrían ser un primer paso para establecer un proceso de medición de calidad del producto para dicha característica de calidad. En la sección 4 se mencionan los resultados principales del mapeo sistemático respecto de las características de calidad mencionadas en los estudios primarios, estándares utilizados y recomendaciones para abordar la calidad del producto. Además, se agregaron las sugerencias para medir el impacto de DevOps. En particular, la investigación se enfocó en identificar propuestas de la medición de la calidad del producto de software y cómo podría medirse la mantenibilidad. La discusión de los resultados se presenta en la sección 5. Finalmente, las conclusiones se presentan en la sección 6.

## 2. Antecedentes

En esta sección se presentan conceptos claves para la comprensión de la calidad del software y específicamente la calidad de producto de software, así como una visión general de algunos trabajos que abordan la relación de DevOps con la calidad del software.

La calidad del software se define como “el grado en el cual un producto de software satisface las necesidades enunciadas e implícitas cuando se usa en condiciones especificadas” (ISO, 2011). La calidad del software se puede estudiar desde distintas perspectivas y cada una de ellas enfoca su atención a diferentes aspectos que pueden ser medidos. Las dos perspectivas más utilizadas en la evaluación de la calidad del software son la basada en la vista de proceso y la vista de producto (ISO, 2011; García-Mireles et al., 2015).

La calidad de producto de software implica que hay un modelo de calidad subyacente que define un conjunto de características de producto (García-Mireles et al., 2015; ISO, 2011). La vista de producto de la calidad del software está basada en un modelo que define un conjunto de características de calidad de producto (Kitchenham & Pfleeger, 1996). En esta perspectiva, la calidad es vista como apropiada para su propósito y depende de las propiedades del producto que satisfacen las necesidades del usuario (Kitchenham & Pfleeger, 1996). La calidad de producto se mide considerando el contexto de la tarea o escenario (Kitchenham & Pfleeger, 1996). Para analizar y estudiar dichos escenarios, los modelos de calidad del software describen un conjunto de características de calidad que son útiles para identificar, medir y evaluar la calidad de producto de software (Mishra & Otaiwi, 2020; ISO, 2011).

Respecto de los modelos de calidad de producto de software, el ISO/IEC 25010 (ISO, 2011) proporciona una guía para identificar los requisitos de calidad de producto. Éste se compone por ocho características de calidad que debería tener un producto de software: adecuación funcional, eficiencia de desempeño, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad. Cada una de estas características contiene subcaracterísticas de calidad y, en conjunto, se tienen en cuenta al evaluar un producto de software.

Esta investigación aborda la característica de calidad denominada “mantenibilidad”, que se define como “el grado de efectividad y eficiencia con el cual los responsables del mantenimiento previsto pueden modificar un producto o sistema” (ISO, 2011). La mantenibilidad está compuesta por un conjunto de subcaracterísticas: modularidad, reusabilidad, analizabilidad, modificabilidad y comprobabilidad. La Tabla 1 presenta la traducción de las definiciones contenidas en el estándar y, entre paréntesis, se señala el nombre en inglés de cada una de las subcaracterísticas (ISO, 2011).

Las notas aclaratorias de dicha norma indican que las modificaciones pueden incluir correcciones, mejoras o adaptaciones del software a los cambios en el ambiente de operación. Además, dichas modificaciones incluyen aquellas que están bajo responsabilidad del personal de soporte especializado, personal de operaciones o usuarios finales. La mantenibilidad también incluye la instalación de actualizaciones y mejoras del software. Por lo tanto, es una característica clave de la calidad del producto que está estrechamente relacionada con el enfoque DevOps.

Por otra parte, el tema del impacto de DevOps en la calidad del producto de software se ha estudiado en distintos tipos de revisiones de literatura. Por ejemplo, en un estudio de mapeo sistemático Mishra y Otaiwi (2020) reportaron que la calidad del software tiene relación con aspectos de DevOps como la cultura, la colaboración, la automatización y el monitoreo. Además, una revisión exploratoria de literatura reporta diez prácticas que podrían influir la calidad del producto (Domínguez-Acosta y García-Mireles, 2021). Finalmente, Céspedes et al. (2020) señalan el interés por estudiar algunas características de calidad del producto, como confiabilidad y mantenibilidad. Además, las prácticas asociadas a DevOps, como el producto mínimo viable, la automatización de la implementación, la automatización de pruebas, cómputo en la nube y la cooperación en equipo, muestran una relación con la mejora en la calidad de producto de software. Sin embargo, ninguno de estos estudios muestra cómo se definen y miden las características de calidad del producto de software.

Subcaracterística	Definición
Modularidad (Modularity)	“Grado en el cual un sistema o programa de computadora se compone de componentes discretos tal que cambiar un componente tiene impacto mínimo en los restantes.” (ISO, 2011)
Reusabilidad (Reusability)	“Grado en el cual un activo puede ser usado en más de un sistema o en la construcción de otros activos.” (ISO, 2011)
Analizabilidad (Analyzability)	“Grado de efectividad y eficiencia con el cual es posible evaluar el impacto de un producto o sistema en un cambio previsto a una o más en sus partes, o diagnosticar un producto para determinar sus deficiencias o causas de fallas, o para identificar las partes por ser modificadas.” (ISO, 2011)
Modificabilidad (Modifiability)	“Grado en el cual un producto o sistema puede ser efectiva y eficientemente modificado sin introducir defectos o degradar la calidad del producto existente” (ISO, 2011)
Comprobabilidad (Testability)	“Grado de efectividad y eficiencia con el que se pueden establecer criterios de prueba para un sistema, producto o componente y se pueden realizar pruebas para determinar si esos criterios se han cumplido.” (ISO, 2011)

Tabla 1 – Definición de las subcaracterísticas de mantenibilidad  
(tomado del ISO/IEC 25010 (ISO, 2011))

### 3. Metodología

Con base en lo expuesto anteriormente, el objetivo de este estudio de mapeo sistemático de literatura es identificar los estudios empíricos que muestren la relación entre las prácticas de DevOps y la calidad del producto de software. Se tomará como referencia de la calidad del producto las ocho características descritas en el modelo ISO/IEC 25010 (ISO, 2011). Puesto que los resultados de dicha revisión se presentaron en Peña, Ávila & García-Mireles (2023), este documento se centrará en la relación entre DevOps y la calidad del producto de software y se extenderá la revisión exploratoria de literatura a los enfoques de medición en DevOps y cómo se podría abordar la medición de la mantenibilidad del producto de software.

Los resultados del mapeo sistemático presentado en este artículo consideraron las guías propuestas por Kitchenham & Charters (2007). La descripción completa del proceso, incluyendo la lista completa de preguntas de investigación, criterios de selección de artículos primarios, procedimiento de extracción de datos se encuentra en (Peña, Ávila & García-Mireles, 2023).

En el mapeo sistemático, la pregunta principal de investigación es: ¿Qué impacto tienen las prácticas de DevOps en la calidad de producto de software de acuerdo con el modelo de calidad descrito por la norma ISO/IEC 25010? De esta pregunta general se derivan varias preguntas más específicas. En este artículo extendido, sólo presentaremos un resumen de los resultados obtenidos para contestar las siguientes:

- RQ-a. ¿Qué características de calidad de producto se informan en los estudios relacionados con DevOps?
- RQ-b. ¿Qué recomendaciones se sugieren para mejorar la calidad de producto al adoptar DevOps?

En el mapeo sólo consideramos la búsqueda automática en los siguientes recursos: ACM Digital Library, IEEE Xplore, Scopus y Web of Science. La cadena de búsqueda utilizada fue:

```
(DevOps) AND (9126 OR 25010 OR usability OR maintainability OR security OR efficiency OR portability OR reliability OR functionality OR compatibility OR "non-functional requirements" OR "nonfunctional requirements" OR nfr OR "quality characteristics" OR "quality objectives" OR "quality factors" OR "quality requirements" OR "quality attributes" OR "software quality" OR "product quality").
```

Con base en los resultados del mapeo sistemático, realizamos un análisis exploratorio del papel que desempeñan las métricas en iniciativas DevOps y en la calidad del producto de software. En particular, la investigación se centró en la mantenibilidad del producto de software. Para ello se realizó una consulta en Scopus con la cadena: (devops AND maintainability AND ( measur\* OR metric\* ). Los resultados fueron verificados el 4 de abril de 2024 y Scopus devuelve sólo ocho registros. Uno de ellos, Tamburri et al. (2018), se enfoca en cómo se podría modificar el patrón Modelo-Vista-Controlador para medir sus eventos de forma automatizada. Sin embargo, nuestro interés es en evaluar el impacto de las prácticas de DevOps en la calidad del producto de software, por lo que este trabajo fue descartado.

Dada la escasez de estudios empíricos sobre este tema, la investigación se centró en métricas de la mantenibilidad del producto de software en cualquier contexto. Las mediciones encontradas se clasificaron con base en las subcaracterísticas de mantenibilidad del ISO/IEC 25010. Esta recopilación de métricas es el punto de referencia para elaborar un proceso de medición de la mantenibilidad del software que se abordará en la siguiente etapa de la investigación.

## 4. Resultados

La sección de resultados presenta los hallazgos principales del mapeo sistemático (Peña, Ávila & García-Mireles, 2023) relacionados con las características de calidad mencionadas en los estudios primarios y las recomendaciones para abordar la calidad del producto. Además, se presentan las principales iniciativas de medición en iniciativas DevOps. Por último, se presenta una lista inicial de métricas de la mantenibilidad que han sido utilizadas en distintos contextos como punto inicial para abordar un proceso de medición de la mantenibilidad del software.

### 4.1. Hallazgos principales del mapeo sistemático

Con base en los 19 estudios primarios seleccionados (Peña, Ávila & García-Mireles, 2023), la Figura 1 muestra las características de calidad de producto identificadas. Dos estudios primarios (Perera, Bandara, & Perera, 2016; Kato, Shimizu & Ishikawa, 2022) mencionan características de calidad como funcionalidad o adecuación funcional, fiabilidad, usabilidad, eficiencia de desempeño, mantenibilidad y portabilidad. Por su parte, hay referencias a la fiabilidad en (Bertolino et al., 2023) y eficiencia del desempeño en (Bezemer et al., 2019). Además, nueve estudios primarios abordan la seguridad del software con una tendencia hacia los estudios empíricos en el sector industrial (lista de referencias en Peña, Ávila & García-Mireles, 2023).

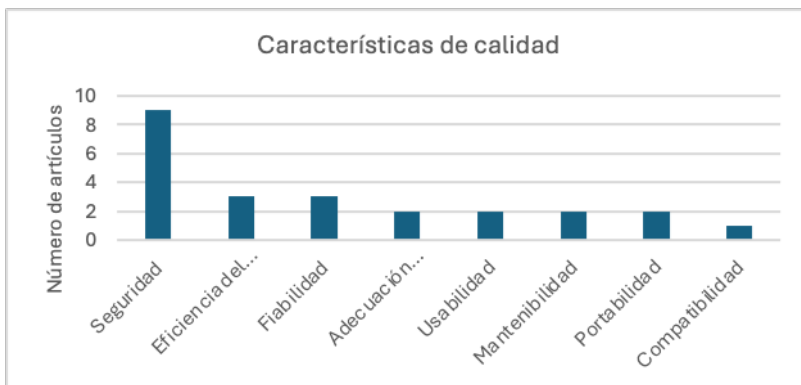


Figura 1 – Características de calidad mencionadas en los estudios primarios

Pocos estándares se citaron en los estudios primarios, entre ellos el IEC 62443 (Moyón et al., 2020) relacionado con la seguridad, el ISO/IEC 29110, mencionado por (Gunawan



& Budiardjo, 2021), y que está orientado a mejorar la calidad del software desde la perspectiva del proceso. Sin embargo, no se encontraron referencias a estándares como el ISO/IEC 25010 (ISO, 2011), cuyo modelo de calidad del producto puede usarse para definir los requisitos de calidad y evaluar el software en sus diferentes etapas de desarrollo.

Respecto de las recomendaciones para mejorar la calidad del producto de software, se sugiere implementar prácticas como el control de versiones y gestión de la configuración, pruebas automatizadas, integración continua automatizada y supervisión continua (Offerman et al., 2022; Kato, Shimizu & Ishikawa, 2022). Otros autores sugieren que todos los miembros del equipo sean conscientes de las responsabilidades asignadas a sus respectivos roles (Perera, Bandara, Perera, 2016). Además, se recomienda establecer los objetivos de calidad del producto con base en las características de calidad del software, y definir los logros de calidad en cada hito del proyecto (Kato, Shimizu & Ishikawa, 2022). Como resultado, se puede establecer un conjunto de pruebas apropiadas para los objetivos de calidad desde la planificación del proyecto y del aseguramiento de la calidad del software (Kato, Shimizu & Ishikawa, 2022).

Uno de los estudios primarios (Gunawan & Budiardjo, 2021) también proporciona recomendaciones para lograr mejoras en el producto, estas son: mapeo, diseño y separación de plataformas y partes de software para personalización mediante la utilización de mecanismos de línea de productos de software; aplicación de conceptos y modelos de diseño de software; y que se requieren pruebas como parte obligatoria en conjunto con el proceso de diseño de acuerdo con los requisitos del software. Por otra parte, las listas de entrada de productos deben incluir el registro de trazabilidad y el registro de riesgos como parte de los requisitos de cierres del proyecto realizados al comienzo del proyecto y actualizados continuamente a lo largo del proyecto.

El estudio de Bezemer et al. (2019), que aborda el rendimiento, recomienda asegurarse de que las herramientas se integren sin problemas en todos los procesos automatizados de DevOps, de manera que tanto el equipo de desarrollo y de operaciones puedan colaborar en la compilación e implementación de código en un entorno de producción. Dado que en su estudio observan que Jenkins CI es la herramienta de integración continua más popular recomiendan que los investigadores proporcionen complementos que permitan una fácil integración de sus herramientas de evaluación de rendimiento en Jenkins.

Algunos de los estudios que abordaron la seguridad también presentaron recomendaciones a tener en cuenta para lograr una mejora de la seguridad en DevOps. El trabajo de Tomas, Li & Huang (2019) recomienda que la seguridad se puede potenciar con un conjunto de herramientas adecuadas, que debe existir un estándar sobre cómo configurar pruebas de seguridad automatizadas, que podría ser beneficioso para las empresas obtener una descripción general de las características de seguridad que ofrecen los proveedores de servicios en la nube, pero también cuáles son las limitaciones y que lo más importante de la seguridad es que los desarrolladores estén capacitados en ella. Similarmente, (Rahman & Williams, 2016) recomiendan el monitoreo automatizado, las pruebas y la implementación automatizadas de software. Además, sugiere que actividades como la colaboración supervisada con el equipo de seguridad podría ayudar a que la implementación automatizada sea beneficiosa para la seguridad del sistema como una actividad de DevOps (Rahman & Williams, 2016).

Otras recomendaciones para mejorar la seguridad sugieren que el personal que diseñe los casos de prueba no sea el mismo programador del componente de software (Mohan, Othmane & Kres, 2018) ya que los casos de prueba funcionan como guardianes de los otros entornos. Además, recomiendan que las partes interesadas deben iniciar y aprobar una solicitud de cambio antes de la implementación en producción y que debe haber derechos de acceso separados para cada uno de los tres entornos: acceso de desarrollo, acceso de TIC y acceso de producción (Mohan, Othmane & Kres, 2018). Un desarrollador no deberá tener derecho de acceso a los entornos de prueba y producción. Finalmente, se propone agregar un rol de evaluador de seguridad independiente (Zeller, 2021).

#### 4.2. Iniciativas de medición en entornos DevOps

Aunque no hay una definición estándar de las etapas del proceso de DevOps (Cappizzi et al., 2019; Alnafessah et al., 2021), algunos autores las definen como la planeación, el desarrollo, la verificación, la prueba, la liberación, el despliegue, la operación y el monitoreo (Cappizzi et al., 2019; Bermon et al. 2022). De estas etapas, para medir la calidad del software podría considerarse el monitoreo. Esta etapa consiste en evaluar los datos relacionados con el despliegue continuo de las aplicaciones de software con la finalidad de encontrar fallas y proporcionar retroalimentación para mejorar el software y guiar su evolución (Alnafessah et al. 2021).

Para evaluar el desempeño de DevOps, las métricas más frecuentes abordan la frecuencia de despliegue, el cambio en el tiempo de espera, la restauración del servicio y el cambio en las tasas de falla (Bermon et al., 2022). Similarmente, para determinar el impacto de DevOps en la entrega del software, en los equipos y en la satisfacción del cliente, se requiere determinar el tiempo de espera, la frecuencia del despliegue, el tiempo promedio de restauración y el porcentaje de cambios fallidos (Offerman et al., 2022). Para Humble y Molesky (2011) DevOps se guía por cuatro conceptos rectores: la cultura, la automatización, la medición y el intercambio. En particular, la medición permite determinar que los objetivos de liberación y mejora se han alcanzado. El proceso de la medición incluye aspectos relacionados con el negocio, como rentabilidad, hasta aspectos a nivel de la implementación del software como número de transacciones por unidad de tiempo (Humble y Molesky, 2011). Este esfuerzo implica una selección cuidadosa de los indicadores de desempeño que fomenten la productividad del equipo de desarrollo y que eliminen cuellos de botella en el proceso de desarrollo (Humble y Molesky, 2011).

Wilkes et al. (2023) señalan que las mediciones del *DevOps Research and Assessment* (DORA) han sido ampliamente aceptadas por la industria del software y se utilizan para medir el desempeño de DevOps. El marco de medición DORA muestra cuatro métricas que se correlacionan con el desempeño de DevOps: frecuencia de despliegue, tiempo de espera para los cambios, tiempo promedio de recuperación, y tasa de fallos por modificaciones. Por su parte, Amaro et al. (2023) identifican y validan 24 mediciones que permiten determinar los resultados de adoptar iniciativas DevOps. Entre ellas, se encuentran mediciones relacionadas con la calidad del producto como tasa de incidentes y errores de producción, volumen de tiques del cliente, tiempo medio entre fallas y cobertura de pruebas automáticas de código. Sin embargo, se informa que las organizaciones enfrentan dificultades para rastrear las mediciones y se requiere mucho esfuerzo y experiencia para obtener estos datos (Amaro et al., 2023). Por lo



tanto, implementar un proceso de medición representa un reto para la mayoría de las organizaciones de software. Por otra parte, las organizaciones no sólo deben priorizar los indicadores de la calidad del producto, sino también las mejoras en la cultura y mejoras del proceso (Amaro et al. 2023).

Por su parte, Diamantopoulos et al. (2020) señalan que el monitoreo efectivo del proceso de desarrollo requiere analizar la calidad del software y que la evaluación se puede automatizar. Por ejemplo, al identificar las violaciones a las prácticas de codificación aceptadas. Además, Visser y Hulstijn (2023) sugieren una adaptación del modelo DMAIC (define, mide, analiza, mejora y controla) para establecer un proceso que permita el monitoreo del desempeño de equipos ágiles/DevOps para que el mejoramiento sea continuo. En particular, considera indicadores de la estabilidad del producto, como tasa de fallos por modificaciones, tiempo promedio de recuperación, y densidad de defectos. En conclusión, las iniciativas de medición de DevOps se han centrado en evaluar el impacto del proceso en diferentes indicadores de interés para la organización. Por su parte las fallas o defectos se utilizan en la perspectiva de procesos para medir la calidad (Kitchenham y Pfleeger, 1996; García-Mireles et al., 2015).

Otras iniciativas podrían responder a la necesidad de mejorar la calidad del producto de software. Algunas investigaciones abordan cómo integrar prácticas de seguridad en DevOps. Por ejemplo, DevSecOps es definido como la integración de las prácticas de seguridad en DevOps (Mohan y Sthmane, 2016). Prates et al. (2019) señalan que la implementación de DevOps podría comprometer la seguridad del software debido a ciclos de desarrollo más rápidos e incremento del número de despliegues. Por otra parte, para desarrollar tecnología de mejora de la seguridad, su revisión de literatura reporta nueve métricas que sirven para medir la efectividad de DevSecOps (Prates et al., 2019). Además, Mallouli et al., 2020 señalan que se requieren nuevas métricas de seguridad para monitorear el canal de DevSecOps para alertar y evitar posibles incidentes de seguridad. Sugieren un enfoque basado en aprendizaje de máquina para incluir controles de seguridad en las herramientas de soporte a las prácticas de DevSecOps (Mallouli et al., 2020).

Finalmente, Snyder y Curtis (2017) reportan la experiencia de implementar DevOps y de evaluar el impacto en la calidad estructural del software. La calidad estructural del software considera aspectos como robustez, seguridad, eficiencia del desempeño, facilidad de cambio, y transferibilidad que se agregaron en el Índice de Calidad Total. Se reporta que las mediciones de tamaño y calidad estructural del software contribuyeron a la productividad y a la reducción del tiempo de desarrollo. Sin embargo, no se indican las métricas específicas usadas para medir los aspectos de calidad señalados.

### **4.3. Medición de la mantenibilidad del software**

Como se mencionó previamente en la sección de metodología, hay escasez de publicaciones científicas que aborden la relación entre prácticas de DevOps y mantenibilidad del software. Además, hay poca información respecto de cómo medir la mantenibilidad en este contexto. Por lo tanto, se realizó una búsqueda exploratoria para identificar métricas de mantenibilidad que se hayan utilizado en otros contextos.

En particular, se encontraron métricas que se han evaluado empíricamente en aplicaciones web, aplicaciones móviles, en software orientado a objetos, entre otros. Las métricas

encontradas se clasificaron tomando como base las subcaracterísticas de calidad de la mantenibilidad descritas en el ISO/IEC 25010 (ISO, 2011). Algunas métricas pueden ser consideradas en varias subcaracterísticas de la mantenibilidad. La Tabla 2, a manera de ejemplo, muestra algunas métricas de mantenibilidad que se han usado para evaluar aplicaciones web desde el diseño del software y la codificación, según las propuestas de Ghosheh, Black & Qaddour (2008) y Chae et al., (2007). Como se observa, la selección de las métricas está fuertemente influenciada por las características del software, la etapa del proceso de desarrollo y las metas de información que se persigan. Por lo tanto, un análisis apropiado de estas métricas debería considerar factores contextuales particulares y las necesidades de información de la organización.

Por otra parte, un conjunto de métricas ampliamente usado es la propuesta de Chidamber y Kemerer (1994) que incluye métricas para código orientado a objetos en donde se pueden usar las métricas como NOR (número de clases raíz), DIT (profundidad del árbol de herencia), y WMC (métodos ponderados por clase para medir la complejidad de los métodos).

Subcaracterística	Métricas (Diseño) Ghosheh, Black & Qaddour (2008)	Métricas (Código) Chae et al. (2007)
Analizabilidad	Métricas de tamaño: número de páginas web, número de páginas del servidor, número de formularios Métricas de complejidad: número total de enlaces, número total de relaciones submit	Tamaño: LOC (Líneas de código) Acoplamiento: RFC (Medida de respuesta a una clase)
Comprobabilidad	Métricas de complejidad	Tamaño: LOC (Líneas de código) Acoplamiento: RFC (Medida de respuesta a una clase)
Modificabilidad	Métricas de tamaño Métricas de complejidad	Tamaño: LOC (Líneas de código)
Modularidad	Acoplamiento de control Acoplamiento de datos	Acoplamiento: DAC – Acoplamiento de abstracción de datos
Reusabilidad	Número de relaciones de inclusión sobre el número de páginas web.	---

Tabla 2 – Algunas métricas para evaluar la mantenibilidad de aplicaciones web (basado en Ghosheh, Black & Qaddour (2008) y Chae et al. (2007))

Para implementar un marco de medición, se deberían tomar en cuenta los requisitos de las mediciones, como: que sean relevantes para el negocio y para el equipo DevOps, que sean medibles objetivamente, confiables y comparables (Visser y Hulstijn, 2023). Por otra parte, el proceso de medición debería estar bien integrado con el proceso de desarrollo existente (Visser y Hulstijn, 2023). Además, los factores que pudieran impactar al desempeño del equipo deberían poder identificarse (Visser y Hulstijn, 2023), por ejemplo, los señalados por Gaorusi et al. (2019). Entre más temprano más temprano una organización inicie su proceso de medición, más pronto podrá establecer

una línea base para medir las mejoras relativas de iniciativas como DevOps (Forsgren y Kernsten, 2018).

Realizar la evaluación a través de mediciones es un reto cuando se procura recolectar medidas de todo el canal de entrega del software debido al uso de múltiples sistemas que se han introducido al proceso de desarrollo para implementar prácticas de DevOps y que recolectan datos a través de las distintas áreas de la organización (Forsgren y Kernsten, 2018). Las decisiones del negocio deberían estar basados en una recolección de datos completa, correcta y exhaustiva que utilice modelos de análisis apropiados (Forsgren y Kernsten, 2018). La decisión sobre el conjunto de métricas por calcular es una tarea difícil que involucra considerar una serie de factores como las características de cada proyecto, el perfil de los desarrolladores y el personal de operaciones, y otros factores contextuales (Diamantopoulos et al., 2020). Además, se requiere un análisis periódico de cada métrica para verificar su confiabilidad con los cambios que se realizan durante el proyecto (Diamantopoulos et al., 2020).

## 5. Discusión

Respecto de los hallazgos del mapeo sistemático de literatura, los estudios primarios no mencionan la utilización de algún modelo de calidad de producto, tampoco definen las características, ni se tiene información en todos los casos de cómo fueron medidas. Solo el estudio (Bezemer et al., 2019), que aborda el rendimiento, lo define y según la definición que utilizan se podría traducir a eficiencia de desempeño. Por otra parte, la mayoría de los estudios primarios (9 de 19) abordan la seguridad, mientras que las otras características de calidad del software están abordadas hasta en un máximo de tres estudios primarios. Por lo tanto, analizar el impacto de DevOps en las características de calidad del producto es un tema abierto a la investigación.

Por su parte, las recomendaciones que se encontraron en los estudios primarios señalan que las pruebas automatizadas, implementaciones continuas y automatizadas, la colaboración y comunicación entre los miembros del equipo son las prácticas más sugeridas y que, a opinión de los estudios empíricos seleccionados, tienen mayor impacto positivo en la calidad de un producto de software. Estas recomendaciones coinciden, además, con ser prácticas de DevOps de las más mencionadas como potenciales para alcanzar mejoras en la calidad de producto de software (Mishra & Otaiwi, 2020; Céspedes et al., 2020).

Respecto de las iniciativas de medición en entornos DevOps se identifican dos áreas de interés relevantes. Por una parte, la necesidad de demostrar que las iniciativas de DevOps son efectivas ha llevado a la implementación de marcos de medición que utilizan indicadores como el tiempo de espera, la frecuencia de despliegue, el tiempo promedio de restauración del servicio y el porcentaje de cambio fallidos (Offerman et al. 2022). Por otra parte, la atención a las amenazas a la seguridad ha llevado al desarrollo de DevSecOps (Mohan y Sthmane, 2016), en donde las iniciativas tratan de establecer controles de seguridad en el código y establecer nuevas métricas apropiadas para este entorno (Prates et al., 2019; Mallouli et al., 2020). Sin embargo, estas iniciativas están orientadas a la perspectiva de proceso para medir al software. Pocos esfuerzos se han realizado para medir la calidad del producto (Snyder y Curtis, 2017).

Respecto de la mantenibilidad del software en DevOps, la revisión exploratoria de literatura señala que hay muy pocas publicaciones que aborden este tema. Como se informa en el ISO/IEC 25010 (ISO, 2011) la mantenibilidad del software está estrechamente relacionada con varias prácticas de DevOps. Por lo tanto, es un área de oportunidad para investigar en qué medida DevOps contribuye a la mantenibilidad del software y establecer la relación entre reducir los tiempos de liberación del producto con su mantenibilidad. Por otra parte, las métricas de mantenibilidad reportadas en la Tabla 2 han sido validadas en contextos distintos al entorno DevOps. Como se señaló previamente, éstas necesitan evaluarse para ver si son apropiadas para los ambientes DevOps y validarse en empresas que hayan adoptado este tipo de enfoque de desarrollo.

Las limitaciones principales de este estudio se deben a que la revisión sólo utilizó la búsqueda automática en cuatro bases de datos que son comunes en la investigación en ingeniería de software. Por lo tanto, no se puede establecer que se realizó una búsqueda exhaustiva. Por otra parte, las búsquedas en Scopus de las iniciativas de medición en DevOps y la mantenibilidad son de naturaleza exploratoria. Por lo tanto, tampoco se puede establecer la completitud del proceso de búsqueda. Sin embargo, lo que si se muestra con estos resultados es un área de oportunidad para abordar la calidad del producto en entornos DevOps, dado el resultado de las revisiones sistemáticas analizadas que son consistentes con los resultados presentados en este trabajo.

## 6. Conclusiones

Respecto del mapeo sistemático, el análisis preliminar mostró pocos estudios empíricos, en entornos DevOps, que abordaran la calidad del producto del software en términos de las características descritas en el ISO/IEC 25010 (ISO, 2011). Pocos estudios definen explícitamente las características de calidad o citan a un estándar apropiado. Además, no se encontraron especificaciones operativas de las métricas usadas para determinar la relación entre las prácticas de DevOps y la calidad del software. Respecto de las iniciativas de medición en DevOps, éstas tienden a medir la calidad del producto desde la perspectiva del proceso. Hay pocos trabajos publicados que indiquen cómo medir la mantenibilidad en entornos DevOps.

Aunque la seguridad es la característica de calidad que se encontró con mayor frecuencia en los estudios primarios, aún falta más trabajo para definir métricas apropiadas y herramientas para usarse en entornos DevOps. Además, el resto de las características de calidad del ISO/IEC 25010, que están definidas por consenso, podrían fundamentar otras investigaciones sobre la calidad del producto de software en dichos entornos. Respecto de la medición de la calidad del producto, la revisión exploratoria de literatura muestra que hay oportunidades para desarrollar procesos de medición.

## Referencias

Alnafessah, A., Gias, A.U., Wang, R., Zhu, L., Casale, G., & Filieri, A. (2021). Quality aware DevOps research: Where do we stand? *IEEE Access*, 9, 44476-44489. <https://doi.org/10.1109/ACCESS.2021.3064867>

- Amaro, R., Pereira, R., & da Silva, M. M. (2023). Capabilities and metrics in DevOps: A design science study. *Information & Management*, 60(5), 103809.
- Bermon Angarita, L., Fernández Del Carpio, A. & Osorio Londoño, A. A. (2022). A Bibliometric Analysis of DevOps Metrics. *DESIDOC Journal of Library & Information Technology*, 42(6). pp. 387-396.
- Bertolino, A., Angelis, G. D., Guerriero, A., Miranda, B., Pietrantuono, R., & Russo, S. (2023). DevOpRET: Continuous reliability testing in DevOps. *Journal of Software: Evolution and Process*, 35(3), e2298.
- Bezemer, C. P., Eismann, S., Ferme, V., Grohmann, J., Heinrich, R., Jamshidi, P., Shang, W., van Hoorn, A., Villavicencia, M., Walter, J. & Willnecker, F. (2019). How is performance addressed in DevOps?. In *Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering* (pp. 45-50).
- Capizzi, A., Distefano, S. & Mazzara, M. (2019). From devops to devdataops: Data management in devops processes. In *International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, pp. 52-62. [https://doi.org/10.1007/978-3-030-39306-9\\_4](https://doi.org/10.1007/978-3-030-39306-9_4)
- Céspedes, D., Angeleri, P., Melendez, K. & Dávila, A. (2020). Software Product Quality in DevOps Contexts: A Systematic Literature Review, In: Mejia, J., Muñoz, M., Rocha, Á. A., Calvo-Manzano, J. (eds) *Trends and Applications in Software Engineering. CIMPS 2019. Advances in Intelligent Systems and Computing*, vol 1071. Springer, Cham. <https://doi.org/10.1007/978-3-030-33547-2>, 2020.
- Chae, H. S., Kim, T. Y., Jung, W. S., & Lee, J. S. (2007). Using metrics for estimating maintainability of web applications: An empirical study. In *6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)* (pp. 1053-1059). IEEE.
- Chidamber, S.R., Kemerer, C.F. (1994)- A metrics suite for object oriented design. *IEEE Trans. Softw. Eng.* 20(6), 476-493.
- Debois, P. (2008). Agile infrastructure and operations: how infra-gile are you?. In *Agile 2008 Conference*. pp. 202-207.
- Diamantopoulos, T., Papamichail, M. D., Karanikiotis, T., Chatzidimitriou, K. C., & Symeonidis, A. L. (2020). Employing contribution and quality metrics for quantifying the software development process. In *Proceedings of the 17th International Conference on Mining Software Repositories*. pp. 558-562.
- Domínguez-Acosta, M. F. & García-Mireles, G. A. (2021). Identifying Activities for Enhancing Software Quality in DevOps Settings. *10th International Conference On Software Process Improvement (CIMPS)*, Torreón, Coahuila, Mexico, 2021, pp. 84-89, <https://doi.org/10.1109/CIMPS54606.2021.9652761>.
- Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps, *IEEE Software*, 33(3), 94-100.

- Forsgren, N., & Kersten, M. (2018). DevOps metrics. *Communications of the ACM*, 61(4), 44-48. <https://doi.org/10.1145/3159169>
- García-Mireles, G. A., Moraga, M. Á., García, F., & Piattini, M. (2015). Approaches to promote product quality within software process improvement initiatives: a mapping study. *Journal of Systems and Software*, 103, 150-166.
- Garousi, V., Tarhan, A., Pfahl, D., Coşkunçay, A., & Demirörs, O. (2019). Correlation of critical success factors with success of software projects: an empirical investigation. *Software Quality Journal*, 27, 429-493.
- Ghosheh, E., Black, S., & Qaddour, J. (2008). Design metrics for web application maintainability measurement. In *2008 IEEE/ACS International Conference on Computer Systems and Applications* (pp. 778-784). IEEE.
- Gunawan, F., & Budiardjo, E. (2021). A quest of software process improvements in DevOps and Kanban: A case study in small software company. In *Proceedings of the 2021 4th International Conference on Software Engineering and Information Management* (pp. 39-45).
- Humble, J. & Molesky, J. (2011). Why enterprises must adopt devops to enable continuous delivery. *Cutter IT J.*, 24(8), 6-12.
- ISO (2011) ISO/IEC 25010: Systems and software engineering – system and software product quality requirements and evaluation (SQuaRE) – System and software quality models.
- Kato, D., Shimizu, A., & Ishikawa, H. (2022). Quality classification for testing work in DevOps. In *Proceedings of the 14th International Conference on Management of Digital EcoSystems* (pp. 156-162).
- Kitchenham, B. & Pfleeger, S. L. (1996). Software quality: the elusive target, *IEEE software*, 13(1), 12-21.
- Kitchenham, B. & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering (EBSE 2007-001). Keele University and Durham University Joint Report.
- Lazuardi, M., Raharjo, T., Hardian, B. & Simanungkalit, T. (2021). Perceived Benefits of DevOps Implementation in Organization: A Systematic Literature Review. In *2021 10th International Conference on Software and Information Engineering, ICSIE 2021* (pp. 10-16). Association for Computing Machinery. <https://doi.org/10.1145/3512716.3512718>.
- Mallouli, W., Cavalli, A. R., Bagnato, A., & De Oca, E. M. (2020). Metrics-driven DevSecOps. In *ICSOFT* (pp. 228-233).
- Mishra, A. & Otaiwi, Z. (2020). DevOps and software quality: A systematic mapping, *Computer Science Review*, 38, 100308.



- Mohan, V. & Othmane, L. B. (2016). SecDevOps: Is It a Marketing Buzzword? - Mapping Research on Security in DevOps. *2016 11th International Conference on Availability, Reliability and Security (ARES)*, Salzburg, Austria, pp. 542-547, <https://doi.org/10.1109/ARES.2016.92>.
- Mohan, V., ben Othmane, L., & Kres, A. (2018). BP: Security concerns and best practices for automation of software deployment processes: An industrial case study. *In 2018 IEEE Cybersecurity Development (SecDev)* (pp. 21-28). IEEE.
- Moyón, F., Soares, R., Pinto-Albuquerque, M., Mendez, D., & Beckers, K. (2020). Integration of security standards in devops pipelines: An industry case study. *In Product-Focused Software Process Improvement: 21st International Conference, PROFES 2020*, (pp. 434-452). Springer International Publishing.
- Offerman, T., Blinde, R., Stettina, C. J., & Visser, J. (2022). A Study of Adoption and Effects of DevOps Practices. *In 2022 IEEE 28th International Conference on Engineering, Technology and Innovation (ICE/ITMC) & 31st International Association For Management of Technology (IAMOT) Joint Conference* (pp. 1-9). IEEE.
- Peña Olivero, N., Ávila George, H. & García-Mireles, G., (2023). Impacto de las prácticas de DevOps en la calidad de producto de software: hallazgos preliminares de un mapeo sistemático. *12 International Conference on Software Process Improvement (CIMPS 2023)*.
- Perera, P., Bandara, M., & Perera, I. (2016). Evaluating the impact of DevOps practice in Sri Lankan software development organizations. *In 2016 sixteenth international conference on advances in ict for emerging regions (icter)* (pp. 281-287). IEEE.
- Prates, L., Faustino, J., Silva, M., & Pereira, R. (2019). Devsecops metrics. *In Information Systems: Research, Development, Applications, Education: 12th SIGSAND/PLAIS EuroSymposium 2019*, (pp. 77-90). Springer International Publishing.
- Rahman, A. A. U., & Williams, L. (2016). Software security in DevOps: synthesizing practitioners' perceptions and practices. *In Proceedings of the international workshop on continuous software evolution and delivery* (pp. 70-76).
- Snyder, B., & Curtis, B. (2017). Using analytics to guide improvement during an Agile-DevOps transformation. *IEEE Software*, 35(1), 78-83.
- Tamburri, D. A., Bersani, M. M., Mirandola, R., & Pea, G. (2018). Devops service observability by-design: Experimenting with model-view-controller. *In Service-Oriented and Cloud Computing: 7th IFIP WG 2.14 European Conference, ESOC 2018*, (pp. 49-64). Springer International Publishing.
- Tomas, N., Li, J., & Huang, H. (2019). An empirical study on culture, automation, measurement, and sharing of devsecops. *In 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)* (pp. 1-8). IEEE.

Visser, T., & Hulstijn, J. (2023). Measuring Agile/DevOps team performance. *In Agile-ISE@ CAiSE* (pp. 17-23).

Wilkes, B., Milani, A. M. P., & Storey, M. A. (2023). A Framework for Automating the Measurement of DevOps Research and Assessment (DORA) Metrics. *In 2023 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (pp. 62-72). IEEE.

Zeller, M. (2021). Towards continuous safety assessment in context of devops. *In Computer Safety, Reliability, and Security. SAFECOMP 2021 Workshops: DECSoS, MAPSOD, DepDevOps, USDAI, and WAISE* (pp. 145-157). Springer International Publishing.