FACULTAD DE POSTGRADO

TESIS DE POSTGRADO

ANÁLISIS DEL NIVEL DE IMPLEMENTACIÓN DE PRÁCTICAS DEVOPS Y LA CALIDAD DEL SOFTWARE DE UNA SOCIEDAD FINANCIERA

SUSTENTADO POR:

ARIEL FERNANDO CASTRO MEJÍA

AKIN ARTURO RAMÍREZ GARCÍA

PREVIA INVESTIDURA AL TÍTULO DE MÁSTER EN GESTIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN

TEGUCIGALPA, F.M., HONDURAS, C.A.

ABRIL, 2026

UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

UNITEC

FACULTAD DE POSTGRADO

AUTORIDADES UNIVERSITARIAS

RECTORA
ROSALPINA RODRÍGUEZ GUEVARA

VICERRECTOR ACADÉMICO NACIONAL
JAVIER ABRAHAM SALGADO LEZAMA

SECRETARIO GENERAL
ROGER MARTÍNEZ MIRALDA

DECANA DE POSTGRADO
ANA DEL CARMEN RETALLY VARGAS

ANÁLISIS DEL NIVEL DE IMPLEMENTACIÓN DE PRÁCTICAS DEVOPS Y LA CALIDAD DEL SOFTWARE DE UNA SOCIEDAD FINANCIERA

TRABAJO PRESENTADO EN CUMPLIMIENTO DE LOS REQUISITOS EXIGIDOS PARA OPTAR AL TÍTULO DE MÁSTER EN

GESTIÓN DE TECNÓLOGIAS DE LA INFORMACIÓN

ASESOR METODOLÓGICO

Dr. JUAN JACOBO PAREDES HELLER

ASESOR TEMÁTICO

Dr. JUAN JACOBO PAREDES HELLER

MIEMBROS DE LA TERNA:

Alba Gabriela Garay Romero
Carlos Roberto Amador
Marvin Roberto Mendoza Valencia

DERECHOS DE AUTOR

ARIEL FERNANDO CASTRO MEJÍA

AKIN ARTURO RAMÍREZ GARCÍA

FACULTAD DE POSTGRADO

ANÁLISIS DEL NIVEL DE IMPLEMENTACIÓN DE PRÁCTICAS DEVOPS Y LA CALIDAD DEL SOFTWARE DE UNA SOCIEDAD FINANCIERA

NOMBRE DE LOS MAESTRANDO:

ARIEL FERNANDO CASTRO MEJÍA

AKIN ARTURO RAMÍREZ GARCÍA

RESUMEN

Este estudio analiza si hay relación entre adoptar prácticas DevOps y obtener mejor calidad en el software que produce una entidad del sector financiero. Se usa ISO/IEC 25010 como referencia normativa. El método aplicado es cuantitativo, las variables se observan sin modificarlas y la información se recoge en un solo período, lo que permite buscar correlaciones entre los factores que se estudian. Para cuantificar las prácticas de DevOps se observan tres elementos: hasta dónde ha madurado la automatización de integración y despliegue continuo, qué tanto se gestiona la infraestructura mediante código, y en qué medida utilizan contenedores. La calidad se mide usando tres atributos del modelo ISO/IEC 25010 relevantes para bancos: si el código permite modificaciones con facilidad, si resguarda adecuadamente los datos, y si opera sin desperdiciar recursos computacionales. Lo que motiva este trabajo es contribuir con datos concretos acerca de cómo DevOps repercute en la calidad del software financiero, especialmente porque la mayoría de las investigaciones previas han puesto atención en qué tan rápido se entrega el software, pero han descuidado examinar su calidad intrínseca.

Palabras clave: DevOps, calidad del software, ISO/IEC 25010, integración y despliegue continuo, infraestructura como código, sociedad financiera.

GRADUATE SCHOOL

ANALYSIS OF THE LEVEL OF IMPLEMENTATION OF DEVOPS PRACTICES AND SOFTWARE QUALITY AT A FINANCIAL SOCIETY

NAME OF THE STUDENT:

ARIEL FERNANDO CASTRO MEJÍA

AKIN ARTURO RAMÍREZ GARCÍA

ABSTRACT

This study analyzes whether there is a relationship between adopting DevOps practices and obtaining better quality in the software produced by an entity in the financial sector. ISO/IEC 25010 is used as a regulatory reference. The method applied is quantitative, the variables are observed without modification, and the information is collected in a single period, which allows for the search for correlations between the factors under study. To quantify DevOps practices, three elements are observed: the extent to which continuous integration and deployment automation has matured, the extent to which infrastructure is managed through code, and the extent to which containers are used. Quality is measured using three attributes of the ISO/IEC 25010 model relevant to banks: whether the code allows for easy modifications, whether it adequately protects data, and whether it operates without wasting computational resources. The motivation for this work is to contribute concrete data on how DevOps impacts the quality of financial software, especially since most previous research has focused on how quickly software is delivered but has neglected to examine its intrinsic quality.

Keywords: DevOps, software quality, ISO/IEC 25010, continuous integration and deployment, infrastructure as code, financial society.

DEDICATORIA

A Dios, fuente inagotable de sabiduría y fortaleza, quien me ha concedido la inteligencia para comprender, la energía para perseverar y las oportunidades que han marcado este camino. Cada página de este trabajo es testimonio de su gracia infinita y de la confianza inquebrantable de que, con Él, ninguna meta es inalcanzable.

A mi madre, el sostén más firme de mi existencia. Su amor incondicional, sus sacrificios y su respaldo permanente han impulsado cada paso de mi formación académica. Ella me ha mostrado con su vida que la fortaleza y la entrega son las bases sobre las cuales se edifican los sueños. Gracias a su ejemplo, he aprendido en la vida, que con esfuerzo, fe y perseverancia todo es alcanzable. Madre querida, este triunfo es tanto tuyo como mío porque sin ti nunca hubiera llegado hasta aquí.

A cada persona que aportó algo valioso para convertir este anhelo en realidad. Que este documento demuestre que cuando se combinan la determinación humana y el respaldo divino, los objetivos se materializan y las aspiraciones se concretan.

Akin Arturo Ramírez García

Dedico este trabajo de investigación, en primer lugar, a mi esposa, por creer siempre en mí y por brindarme su apoyo incondicional durante todo este proceso. Gracias por cada palabra de ánimo en las noches de desvelo, por tu comprensión y por el apoyo que me diste en nuestro hogar, lo cual me permitió mantenerme enfocado y culminar con éxito este logro.

A mi madre, por estar siempre a mi lado apoyándome en cada etapa de mi vida académica, por su amor incondicional y por nunca dejar de creer en mí, aun en los momentos más difíciles, gracias a su esfuerzo y dedicación pude llegar hasta este punto.

Finalmente, dedico este trabajo a mis familiares, amigos y colegas de trabajo, quienes de una u otra forma me brindaron su apoyo, ánimo y colaboración durante el desarrollo de esta investigación. Su comprensión y disposición para ayudarme fueron fundamentales para mantener el enfoque y la constancia necesarios para culminar este proceso académico.

Ariel Fernando Castro Mejía

AGRADECIMIENTOS

A Dios, por su infinita misericordia y por darme fuerzas y entusiasmo en cada momento de este proceso académico. Él me dio la claridad mental en los días difíciles, la perseverancia cuando el cansancio amenazaba vencerme y la paz necesaria para continuar. Sin su presencia constante, este logro no habría sido posible.

A mi madre, mi compañera incondicional en esta travesía. Gracias por cada palabra de aliento, por creer en mí incluso cuando yo dudaba, y por los sacrificios silenciosos que hiciste para que yo pudiera dedicarme plenamente a este proyecto. Tu apoyo ha sido el cimiento sobre el cual construí este sueño.

A cada familiar, amigo y persona que de alguna forma estuvo presente durante estos años, brindándome su apoyo, comprensión y ánimo. Sus gestos, por pequeños que parecieran, fueron combustible para seguir adelante.

Akin Arturo Ramírez García

Agradezco primeramente a Dios por darme la fortaleza, la fe y la perseverancia necesarias para continuar aun en los momentos más complejos de este proceso, permitiéndome llegar hasta la etapa final de este trabajo de investigación.

De manera especial, agradezco a mi esposa por su apoyo emocional, comprensión y acompañamiento constante durante mis estudios, así como por su dedicación y paciencia, que fueron fundamentales para alcanzar este objetivo académico.

Agradezco también a mi madre, por su amor, confianza y apoyo permanente a lo largo de mi vida, por creer en mí desde siempre y por sentirse orgullosa de cada logro alcanzado. Su respaldo ha sido fundamental en cada etapa de mi formación académica y personal, brindándome siempre ánimo para seguir adelante aun cuando las circunstancias fueron difíciles.

Ariel Fernando Castro Mejía

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

CAPÍTULO I. PLANTEAMIENTO DE LA INVESTIGACIÓN

El presente capítulo establece las bases de los conceptos necesarios para comprender el estudio de esta investigación. Aquí se expone el panorama general de toda el área de interés, así como los fundamentos teóricos que justifican la relevancia del tema y la importancia que tiene por el hecho de estar ligado a una entidad financiera. Asimismo, se revisan los antecedentes teóricos y empíricos que permiten identificar las brechas de conocimiento y delimitar el alcance que tendrá la investigación propuesta.

1.1. INTRODUCCIÓN

En el contexto actual, el desarrollo de software se ha consolidado como un componente estratégico para la transformación digital y la competitividad de las organizaciones en todos los sectores productivos. La acelerada digitalización de los servicios ha convertido al software en un elemento esencial de sostenibilidad e innovación empresarial, no solo como recurso tecnológico, sino como un eje fundamental para garantizar eficiencia, continuidad operativa y valor al cliente.

De acuerdo con un estudio del sector tecnológico, la adopción de DevOps ha crecido de manera sostenida en los últimos años, alcanzando una presencia significativa en el ámbito financiero, donde la estabilidad operativa y la agilidad son factores críticos para la competitividad (Redgate, 2020). Este crecimiento refleja un cambio estructural hacia modelos de desarrollo más colaborativos, automatizados y centrados en la entrega continua de software (Forsgren, Humble, & Kim, 2018; Humble & Molesky, 2011). Según estos autores, la adopción de prácticas DevOps y metodologías de entrega continua permite a las organizaciones reducir los ciclos de despliegue, mejorar la calidad del software y aumentar la capacidad de respuesta ante incidentes, consolidando un enfoque más ágil y eficiente en la gestión de proyectos tecnológicos.

En el caso de las instituciones financieras, la adopción de DevOps se asocia con beneficios tangibles, como una reducción de los tiempos de despliegue, la automatización de pruebas y un incremento en la capacidad de respuesta ante incidentes o fallos críticos (Klemetti et al., 2022). Sin embargo, a pesar de los beneficios evidentes, la evaluación de la calidad del software en entornos DevOps continúa siendo una de las áreas menos abordadas. Atributos como mantenibilidad, fiabilidad, eficiencia y seguridad suelen quedar fuera de las evaluaciones sistemáticas, lo que impide conocer con precisión el

impacto de las prácticas DevOps sobre la calidad del producto final (Mishra & Otaiwi, 2020; García-Mireles, Peña Olivero, & Ávila-George, 2024).

Para abordar esta limitación, la norma ISO/IEC 25010 propone un marco estandarizado de evaluación de la calidad del software, que incluye atributos clave como funcionalidad, fiabilidad, eficiencia, mantenibilidad, portabilidad y seguridad (ISO/IEC, 2011). No obstante, estudios recientes destacan que la integración efectiva de estas métricas de producto en los flujos DevOps es aún incipiente, especialmente en sectores donde la trazabilidad, la estabilidad y la conformidad regulatoria son esenciales (López et al., 2022). Esta desconexión entre la velocidad de entrega y la calidad del producto constituye un vacío de conocimiento que requiere mayor atención empírica y metodológica.

Particularmente en el sector financiero, los retos son más complejos debido a la coexistencia de infraestructuras legadas, la necesidad de cumplir normativas estrictas y la criticidad de los sistemas transaccionales. En este entorno, la falta de mediciones de calidad objetivas puede derivar en incrementos de la deuda técnica, fallos operativos y riesgos de seguridad (Şenkal et al., 2023). Por ello, la gobernanza tecnológica basada en métricas se vuelve un componente estratégico para garantizar la resiliencia, la eficiencia y la sostenibilidad del software.

En este contexto, la presente investigación tiene como propósito evaluar cuantitativamente el impacto de las prácticas DevOps sobre la calidad del producto de software en una entidad financiera, tomando como referencia los atributos definidos en ISO/IEC 25010. Se analizarán tres dimensiones clave, mantenibilidad, eficiencia y seguridad, estableciendo su relación con la madurez en la adopción de prácticas DevOps tales como integración continua, despliegue continuo, automatización de pruebas y uso de infraestructura como código. Este estudio pretende generar evidencia empírica que contribuya a comprender cómo las prácticas DevOps inciden en la calidad técnica del software y a diseñar modelos de medición más integrados y sostenibles.

En síntesis, esta investigación busca contribuir tanto al conocimiento científico como a la práctica profesional, ofreciendo un marco de análisis que fortalezca la trazabilidad y la gobernanza tecnológica. Al integrar métricas de producto y proceso en un enfoque automatizado, se aspira a reducir la brecha entre la agilidad operativa y la calidad del

software, promoviendo la confiabilidad y la resiliencia de los sistemas financieros modernos.

1.2. ANTECEDENTES DEL PROBLEMA

En los últimos años, el desarrollo de software ha experimentado transformaciones profundas impulsadas por la adopción de metodologías ágiles y prácticas DevOps, con el objetivo de reducir los tiempos de entrega, mejorar la eficiencia de los procesos de desarrollo y operación, y responder de manera ágil a los requerimientos del negocio (Ebert, Gallardo, Hernantes, & Serrano, 2016). DevOps se ha consolidado como un enfoque integral que promueve la colaboración entre los equipos de desarrollo, pruebas y operaciones, facilitando la automatización de tareas repetitivas, la supervisión continua de procesos y la retroalimentación constante a lo largo del ciclo de vida del software (Ebert, Gallardo, Hernantes, & Serrano, 2016). Estas prácticas han permitido a las organizaciones acelerar los procesos de despliegue, mejorar la entrega de valor al usuario final y establecer entornos de desarrollo más resilientes y adaptativos, capaces de enfrentar cambios en los requisitos y demandas del mercado de manera eficiente (Ebert, Gallardo, Hernantes, & Serrano, 2016).

A pesar de los beneficios evidentes, la evaluación de la calidad del software en entornos DevOps continúa presentando limitaciones importantes. La mayoría de las métricas tradicionalmente implementadas se centran en indicadores de proceso, como la frecuencia de despliegues, la velocidad de entrega o el tiempo medio de recuperación ante fallos (Forsgren & Kersten, 2018). Si bien estos indicadores son valiosos para medir la eficiencia de los procesos, la medición de las propiedades internas del software que determinan su calidad real recibe menor atención (Mishra & Otaiwi, 2020). Características como la mantenibilidad, eficiencia, fiabilidad, portabilidad y seguridad, que impactan directamente la sostenibilidad del software a largo plazo, no siempre se reflejan en los indicadores de proceso y, por tanto, pueden quedar desatendidas en la gestión de proyectos DevOps (Mishra & Otaiwi, 2020; García-Mireles, Peña Olivero, & Ávila-George, 2024). Esta situación evidencia una brecha significativa en la evaluación de la calidad del software, que limita la comprensión integral del efecto de las prácticas DevOps sobre los productos finales y genera incertidumbre sobre la robustez de las soluciones entregadas.

Investigaciones recientes señalan que la integración de métricas de producto dentro de los flujos DevOps es un paso esencial para abordar esta brecha (García-Mireles, Peña

Olivero, & Ávila-George, 2024). La norma ISO/IEC 25010 proporciona un marco estandarizado para la medición de la calidad del software, definiendo atributos críticos como funcionalidad, fiabilidad, eficiencia, mantenibilidad, portabilidad y seguridad. López et al. (2022) indican que, mientras que los entornos ágiles y DevOps aplican múltiples métricas de proceso, existen pocos estudios que desarrollen indicadores capaces de evaluar directamente la calidad del software, generando una desconexión entre la rapidez de entrega y la garantía de productos confiables. La falta de métricas de producto integradas en los flujos de desarrollo dificulta la gestión integral de la calidad y limita la capacidad de las organizaciones para mantener altos estándares en sus productos (López et al., 2022).

Esta problemática se vuelve particularmente crítica en sectores altamente regulados, como el financiero, donde la precisión, consistencia y seguridad de los sistemas son esenciales. La implementación de DevOps en instituciones financieras ha permitido mejorar los tiempos de liberación y automatizar pruebas de software, pero también ha introducido riesgos adicionales relacionados con la mantenibilidad y la integridad del software. Despliegues frecuentes o automatizados sin un control riguroso de calidad pueden comprometer la consistencia de los sistemas transaccionales, generar errores en la gestión de datos y afectar la integridad de la información crítica para la operación del negocio (Prates, Faustino, Silva, & Pereira, 2019). Asimismo, la normativa vigente en el sector financiero exige que las organizaciones cuenten con evidencias verificables de control y calidad, lo que subraya la necesidad de implementar procesos de medición que estén alineados con estándares internacionales y permitan la trazabilidad de la calidad de los productos de software (Orozco Garcés, Pardo Calvache, & Suescún Monsalve, 2022).

La ausencia de métricas de producto integradas en los flujos DevOps representa una limitación para la gobernanza tecnológica y la gestión de riesgos. La implementación de un sistema automatizado que combine métricas de proceso y métricas de producto permitiría monitorear continuamente la calidad del software, garantizar la continuidad operativa y fortalecer la resiliencia organizacional. La combinación de ambos enfoques facilita una evaluación más completa de la efectividad de los procesos de desarrollo y de la calidad intrínseca del software, permitiendo a las organizaciones tomar decisiones informadas que optimicen sus productos y reduzcan riesgos operativos (Şenkal et al., 2023).

En este sentido, aunque DevOps ha transformado los procesos de desarrollo de software y ha mejorado la eficiencia y la agilidad, la ausencia de métricas de producto integradas limita la capacidad de las organizaciones para garantizar software confiable y sostenible, la adopción de un sistema automatizado que integre métricas de proceso y métricas de producto representa no solo una solución técnica, sino también una estrategia organizacional. Permite a las instituciones financieras asegurar la continuidad operativa, fortalecer la resiliencia frente a fallos, optimizar la eficiencia de sus equipos de desarrollo y mejorar la experiencia del usuario final. Asimismo, contribuye a cerrar la brecha existente entre la velocidad de entrega, característica central de DevOps, y la necesidad de garantizar la calidad y confiabilidad del software.

1.3. DEFINICIÓN DEL PROBLEMA

En este apartado se realiza el enunciado y el planteamiento el problema y se formulan las preguntas de investigación.

1.3.1 ENUNCIADO DEL PROBLEMA

Actualmente las entidades financieras enfrenta un reto importante en la era de la transformación digital, en donde el asegurar la calidad del software en un ámbito de entrega continua, se ha vuelto esencial para ser competitivo, la adopción de prácticas de DevOps ha venido a responder de forma estratégica, prometiendo reducir tiempos de despliegue de aplicativos, aumentar la frecuencia de despliegues y facilitar la colaboración entre equipos de desarrollo y operaciones (Forsgren & Kersten, 2018), sin embargo, esta aceleración del ciclo de vida del software plantea interrogantes fundamentales sobre su efecto en la calidad del producto final, especialmente en un sector donde los fallos pueden tener consecuencias económicas, reputacionales y regulatorias significativas.

La evidencia científica indica una brecha preocupante, a pesar de que DevOps se ha establecido como metodología principal, en donde su evaluación se ha centrado en métricas de proceso, como la frecuencia de despliegue, tiempo de restauración, tasa de fallos, mientras que la calidad del software, entendida a través de estándares como ISO/IEC 25010, sigue sin ser suficientemente analizada (Peña et al. , 2024), estos autores realizaron un análisis sistemático de la literatura donde se identificó que en 19 investigaciones empíricas revisadas, pocos estudios definen de manera clara las características de calidad o establecen métricas operativas para medir la mantenibilidad, seguridad o eficacia del producto, esta falta de enfoque es especialmente preocupante,

dado que la norma ISO/IEC 25010 establece ocho atributos esenciales de calidad, funcionalidad, desempeño, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad, de los cuales solo la seguridad ha recibido atención considerable en los entornos DevOps (ISO, 2011; Peña et al. , 2024).

Las organizaciones del sector están implementando gradualmente prácticas de DevOps, como la automatización de integración y despliegue continuo, infraestructura como código y contenerización, con el fin de permanecer competitivos y cumplir expectativas de clientes digitalizados, sin embargo, estas mismas instituciones operan bajo normativas estrictas que requieren trazabilidad, auditoría, robustez en seguridad y alta disponibilidad. La importancia de esta investigación para la Maestría en Gestión de Tecnologías de Información con orientación en Ciberseguridad es clara y directa, porque se ocupa de la necesidad de crear procesos de medición estructurados que permitan evaluar el efecto de las prácticas DevOps en aspectos esenciales de calidad, especialmente en seguridad, donde la identificación temprana de vulnerabilidades y el tiempo de resolución son métricas clave, en mantenibilidad, se garantiza la viabilidad del código a largo plazo y en eficiencia está relacionada con el uso de recursos y el tiempo de respuesta del sistema. Segundo, responde al llamado de la comunidad científica para desarrollar especificaciones operativas de métricas que establezcan la relación entre prácticas DevOps y calidad del software (Peña et al., 2024). Tercero, proporciona evidencia empírica en un contexto industrial poco representado en la literatura.

1.3.2 FORMULACIÓN DEL PROBLEMA.

La revisión bibliográfica indica que las investigaciones sobre DevOps se han centrado en métricas de proceso, mientras que la calidad del producto de software según ISO/IEC 25010 permanece insuficientemente estudiada (Peña et al., 2024) y no se ha encontrado especificaciones operativas que relacionen prácticas DevOps con características como mantenibilidad, seguridad y eficiencia en entidades financieras, ante esta brecha, surge la interrogante:

¿Cuál es la relación entre el grado de implementación de prácticas DevOps y las métricas de calidad del producto de software (mantenibilidad, seguridad y eficiencia) en una entidad financiera?

1.3.3 PREGUNTAS DE INVESTIGACIÓN.

En este apartado se formulan las preguntas de investigación que surgen como resultado del planteamiento del problema de la investigación, además, guían el presente proyecto final.

1. ¿Cuál es el nivel de implementación de prácticas DevOps (automatización CI/CD, infraestructura como código y contenerización) en la entidad financiera?
2. ¿Qué relación existe entre la implementación de automatización de integración continua y las métricas de mantenibilidad del software en la entidad financiera?
3. ¿Qué relación existe entre el uso de infraestructura como código y contenerización con las métricas de eficiencia del software en la entidad financiera?
4. ¿Qué relación tienen los procesos de despliegue automatizado en las métricas de seguridad del producto de software de la entidad financiera?

1.4. OBJETIVOS (GENERAL Y ESPECÍFICOS)

En el siguiente apartado se muestran los objetivos que han de servir como base para la investigación.

1.4.1 OBJETIVO GENERAL

Esta investigación propone:

Determinar la relación entre el grado de implementación de prácticas DevOps y las métricas de calidad del producto de software (mantenibilidad, seguridad y eficiencia) según ISO/IEC 25010 en una entidad financiera.

1.4.2 OBJETIVOS ESPECÍFICOS

El presente estudio pretende:

1. Cuantificar el nivel de implementación de prácticas DevOps mediante indicadores de automatización CI/CD, infraestructura como código y contenerización en la entidad financiera.
2. Analizar la correlación entre la implementación de automatización de integración continua y las métricas de mantenibilidad del software en la entidad financiera.
3. Evaluar la relación entre infraestructura como código y contenerización con las métricas de eficiencia de desempeño del software en la entidad financiera.

4. Examinar relación entre los procesos de despliegue automatizado en las métricas de seguridad del producto de software de la entidad financiera.

1.5. JUSTIFICACIÓN

El presente trabajo pretende determinar cuál es la relación que existe entre el grado de implementación de las prácticas DevOps y las métricas de calidad del producto de software en el contexto de una entidad financiera. El enfoque en estas variables se justifica debido a que a través del análisis sistemático de literatura que fue realizado por Peña et al. (2024), fue posible identificar una brecha que resulta significativa, en donde de los 19 estudios empíricos que fueron revisados sobre DevOps, resulta que solo unos pocos realmente definen con claridad las características de calidad según lo que establece la norma ISO/IEC 25010, y lo que es más importante, ninguno logra establecer especificaciones que sean operativas de las métricas para relacionar DevOps con cosas como mantenibilidad, seguridad o eficiencia del producto de software. Las investigaciones que existen actualmente más bien se centran en lo que son métricas de proceso, por ejemplo, la frecuencia de despliegue o cuánto tiempo toma la restauración, estas son las llamadas métricas DORA que mencionan Forsgren y Kersten (2018), sin embargo, las características que son propias e intrínsecas del producto de software siguen sin recibir la atención suficiente en términos de investigación.

Decidimos estudiar el sector financiero específicamente porque resulta que estas organizaciones se encuentran enfrentando una situación bastante compleja, por un lado necesitan acelerar todo lo que son sus ciclos de desarrollo usando prácticas DevOps si quieren mantenerse competitivos en el mercado actual, pero por otro lado también tienen que operar cumpliendo con marcos regulatorios que son bastante estrictos y que les exigen tener trazabilidad completa, realizar auditorías, mantener seguridad robusta y garantizar alta disponibilidad de sus sistemas. Esta situación dual nos lleva a preguntarnos si acaso toda esta velocidad que promete DevOps no estará de alguna forma comprometiendo lo que es la calidad final del producto, y esto cobra mayor relevancia cuando pensamos que cualquier fallo que ocurra en un sistema financiero puede traer consecuencias que son económicas, reputacionales y hasta regulatorias bastante serias para las instituciones.

Vale la pena mencionar que esta investigación tiene relevancia clara para los estudiantes de la Maestría en Gestión de Tecnologías de Información con orientación en Ciberseguridad, porque aborda de forma específica cómo DevOps impacta las métricas relacionadas con seguridad, donde cosas como detectar vulnerabilidades, el tiempo que

lleva remediarlas y cumplir con los estándares son aspectos críticos según lo que plantean Rahman y Williams (2016). También está el hecho de que la investigación ayuda a desarrollar capacidades para evaluar cuantitativamente los sistemas de información, algo que se alinea con lo que debe saber hacer un gestor de tecnologías cuando tiene que tomar decisiones basándose en evidencia real.

Esta investigación busca entregar información que sea útil y práctica para quienes gestionan tecnología y están implementando o pensando implementar DevOps, mostrando evidencia sobre qué prácticas específicas como automatización CI/CD, infraestructura como código o contenerización realmente mejoran la calidad, lo cual ayuda a priorizar inversiones y diseñar estrategias que logren balancear velocidad con calidad. Como hay pocas investigaciones en contextos latinoamericanos según Peña et al. (2024), este trabajo aporta conocimiento regional específico. La viabilidad está garantizada porque tenemos acceso a una entidad financiera que está iniciando a usar implementaciones DevOps, las métricas se pueden extraer de herramientas existentes.

CAPÍTULO II. MARCO TEÓRICO

Este capítulo reúne los fundamentos teóricos que sustentan la investigación sobre DevOps y calidad del software. A través de cuatro componentes, se presentan los antecedentes técnicos, las definiciones clave, los enfoques teóricos aplicables y el método seleccionado para el estudio. Su propósito es ofrecer un marco claro que oriente la formulación del problema, respalde las decisiones metodológicas y oriente el diseño del estudio en función de la literatura científica existente.

2.1 ANÁLISIS DE LA SITUACIÓN ACTUAL

Diversos estudios documentan beneficios concretos derivados de la implementación de DevOps. Forsgren, Humble y Kim (2018) reportan que su aplicación puede incrementar la eficiencia operativa, disminuir la incidencia de errores en fases posteriores del desarrollo y mejorar la resiliencia de los sistemas ante fallos. Además, investigaciones recientes muestran que la automatización de pruebas, la utilización de infraestructura como código y la adopción de flujos de integración y despliegue continuo fortalecen la estabilidad y la predictibilidad de los sistemas de software (Erich, Amrit, & Daneva, 2017; Rodríguez et al., 2016). Estos hallazgos evidencian que la incorporación de DevOps no solo tiene impacto en los procesos de desarrollo, sino también en la calidad final de los productos entregados.

Por estas razones, el análisis del efecto de DevOps sobre métricas de calidad del producto constituye un área de investigación relevante, que puede aportar modelos más integrales y rigurosos para evaluar la efectividad de estas prácticas en entornos organizacionales con altos requerimientos de confiabilidad, seguridad y mantenibilidad (López et al., 2022). Además, considerando la creciente complejidad del software y la necesidad de cumplir con normativas estrictas en sectores críticos como el financiero y de telecomunicaciones, resulta necesario contextualizar la adopción de DevOps dentro de marcos normativos y estándares internacionales. Esta contextualización permite identificar no solo los beneficios operativos y de calidad, sino también las limitaciones actuales en la medición objetiva de resultados y la implementación de prácticas consistentes. Así, la revisión de la literatura sirve como base para diseñar estudios que conecten los niveles de madurez DevOps con indicadores medibles de calidad, seguridad y eficiencia, fortaleciendo la pertinencia y aplicabilidad de la investigación en entornos reales.

2.1.1 ANÁLISIS DEL MACROENTORNO

DevOps se puede conceptualizar como un enfoque socio-técnico que combina prácticas de automatización, colaboración continua y supervisión integral del ciclo de vida del software, con el fin de optimizar los procesos de entrega y permitir que las modificaciones en los productos se realicen de manera ágil y confiable. Según García-Mireles, Peña Olivero y Ávila-George (2024), este modelo no solo transforma la interacción entre los equipos de desarrollo y operaciones, sino que también contribuye a la estabilidad del software al facilitar actividades que afectan directamente atributos de calidad, como la mantenibilidad, la eficiencia y la confiabilidad, de acuerdo con los criterios de la norma ISO/IEC 25010 (ISO/IEC, 2023). La adopción de DevOps se ha visto impulsada por principios ágiles y Lean, que promueven la mejora continua, la entrega frecuente de valor y la capacidad de respuesta ante cambios en los requisitos del negocio (Humble & Molesky, 2011).

En la última década, la investigación sobre DevOps ha mostrado un crecimiento sostenido, motivado por la necesidad de acelerar los ciclos de desarrollo y mejorar la calidad de los productos de software en entornos altamente competitivos (Mishra & Otaiwi, 2020). Las organizaciones buscan implementar prácticas que permitan responder rápidamente a cambios en los requisitos, reducir errores y optimizar la eficiencia operativa, especialmente en sectores críticos como el financiero.

DevOps se ha consolidado como un enfoque alineado con tendencias globales en ingeniería de software, apoyado en principios ágiles y Lean, así como en estándares internacionales orientados a la calidad del producto. En este contexto, la norma ISO/IEC 25010 proporciona un marco de referencia para evaluar atributos como mantenibilidad, eficiencia y confiabilidad, los cuales se ven influenciados por la adopción de prácticas DevOps (ISO/IEC, 2023).

Adicionalmente, la implementación de prácticas de DevOps se ve influenciada por tendencias macroeconómicas y tecnológicas, como la digitalización acelerada, la globalización de los servicios de TI y la presión competitiva por innovar de manera rápida y segura. Estas tendencias no solo limitan la estrategia tecnológica de las organizaciones, sino que también incrementan la necesidad de contar con marcos de referencia claros que permitan medir el impacto de la automatización, la integración y la entrega continuas en la calidad de los productos de software. La evidencia global muestra que los países y empresas que integran estas prácticas con estándares internacionales tienden a alcanzar

mayores niveles de confiabilidad y eficiencia, lo que refuerza la pertinencia de estudiar DevOps en un contexto macroestructural.

2.1.2 ANÁLISIS DEL MICROENTORNO

Mishra y Otaiwi (2020) realizaron un mapa sistemático de la literatura sobre DevOps y calidad de software, identificando tres dimensiones fundamentales: cultura colaborativa, automatización y monitoreo continuo. La cultura colaborativa fomenta la comunicación constante entre los equipos de desarrollo y operaciones, promoviendo la responsabilidad compartida sobre la calidad del producto. La automatización incluye la integración y despliegue continuos, la ejecución de pruebas automatizadas y la gestión programática de la infraestructura, lo que disminuye errores humanos y acelera los ciclos de entrega. Por último, el monitoreo continuo permite supervisar el desempeño del software en tiempo real, detectar fallos tempranamente y generar retroalimentación inmediata. Estas dimensiones no solo promueven la eficiencia, sino que también tienen el potencial de impactar directamente en las métricas de calidad del producto, como mantenibilidad, eficiencia y confiabilidad, definidas en la norma ISO/IEC 25010 (ISO/IEC, 2023).

De manera complementaria, diversas investigaciones han identificado prácticas DevOps concretas que resultan fundamentales para mejorar la calidad del software. Entre estas prácticas destacan la integración continua, el despliegue continuo, la automatización de pruebas y la gestión de la infraestructura (Erich et al., 2017). La adopción sistemática de estas prácticas favorece la calidad intrínseca del producto, no obstante, también existe una carencia de métricas objetivas y estandarizadas que permitan cuantificar con precisión el impacto real de estas prácticas sobre el software (López et al., 2022).

Los estudios empíricos también ofrecen hallazgos relevantes. Offermann et al. (2022) encuestaron a desarrolladores y gerentes de TI a nivel internacional, encontrando una correlación positiva entre la madurez en la adopción de DevOps y la percepción de mejoras en eficiencia, estabilidad y velocidad de entrega. No obstante, los autores destacan que la ausencia de mecanismos claros de gobernanza y métricas definidas puede conducir a prácticas inconsistentes y resultados no replicables. Esta situación refuerza la necesidad de contar con métricas claras y estandarizadas para evaluar cómo el grado de implementación de DevOps contribuye a atributos de calidad definidos por estándares como ISO/IEC 25010 (ISO/IEC, 2023).

Cordova Urbina, Díaz Sifuentes y Mendoza de los Santos (2025), a través de una revisión sistemática de 17 estudios sobre DevOps en la gestión de servicios de TI, confirmaron que la adopción de estas prácticas contribuye a optimizar procesos, reducir defectos y mejorar la productividad. Sin embargo, enfatizan que la falta de métricas objetivas de producto limita la capacidad de evaluar de manera precisa el impacto de DevOps sobre la calidad del software. Este hallazgo resalta la relevancia de integrar métricas de calidad definidas en estándares internacionales, como ISO/IEC 25010, con el grado de implementación de DevOps (ISO/IEC, 2023). De esta forma, es posible evaluar atributos específicos de la variable dependiente, tales como funcionalidad, eficiencia, mantenibilidad, confiabilidad, portabilidad y seguridad.

En síntesis, la revisión de estudios relevantes confirma que DevOps ofrece beneficios tangibles en términos de eficiencia operativa, reducción de defectos y mejora en la estabilidad de los sistemas. No obstante, persiste una brecha crítica en la medición objetiva de la calidad del producto (García-Mireles et al., 2024). Esta limitación restringe la capacidad de las organizaciones para cuantificar y optimizar los resultados de sus prácticas DevOps, especialmente en contextos regulados y de alta criticidad. La evidencia revisada subraya la necesidad de investigaciones orientadas a establecer relaciones claras entre el grado de adopción de DevOps y métricas específicas de calidad del software, proporcionando insumos empíricos que puedan guiar la toma de decisiones estratégicas, fortalecer la gobernanza tecnológica y mejorar la confiabilidad de los sistemas en entornos corporativos complejos (Şenkal et al., 2023).

2.1.3 ANÁLISIS INTERNO

La revisión previa muestra que, si bien existe consenso en que la adopción de DevOps contribuye a mejorar la calidad del software, aún persisten vacíos importantes respecto a su aplicación en sectores altamente regulados como el financiero (Orozco Garcés, Pardo Calvache, & Suescún Monsalve, 2022). Este tipo de organizaciones operan bajo estrictos requerimientos de seguridad, confiabilidad y trazabilidad, lo que demanda mecanismos más rigurosos para evaluar el impacto real de DevOps sobre la calidad del producto. En este sentido, diversos autores han señalado que la literatura técnica todavía no ofrece lineamientos suficientemente claros sobre cómo medir de manera objetiva los atributos de calidad dentro de entornos DevOps, especialmente cuando se requiere cumplir simultáneamente con normativas internas, auditorías externas y modelos de

control de riesgos tecnológicos (Orozco Garcés, Pardo Calvache, & Suescún Monsalve, 2022).

García-Mireles et al. (2024) enfatizan que, aunque las investigaciones reconocen el potencial de DevOps para fortalecer la mantenibilidad, la eficiencia operativa y la estabilidad de los sistemas, los estudios empíricos rara vez explican cómo deben evaluarse estos atributos mediante métricas estandarizadas o comparables entre organizaciones. Esta falta de claridad metodológica genera brechas en la implementación práctica, ya que los equipos suelen depender de indicadores heterogéneos o definidos localmente, lo que dificulta la validación rigurosa de mejoras en la calidad del software. En consecuencia, se vuelve necesario avanzar hacia marcos de medición más robustos que integren prácticas DevOps con modelos reconocidos de aseguramiento de la calidad, como ISO/IEC 25010, permitiendo establecer relaciones verificables entre los niveles de madurez DevOps y los resultados reales en la calidad del producto (ISO/IEC, 2023).

Adicionalmente, la literatura especializada destaca que el éxito de DevOps no depende únicamente de la adopción técnica de herramientas o prácticas automatizadas, sino también del compromiso activo de la alta gerencia. La transformación hacia un modelo DevOps implica cambios profundos en la cultura organizacional, la gestión del riesgo, la asignación de responsabilidades y los flujos de trabajo tradicionales entre equipos. Sin un respaldo explícito de los niveles directivos reflejado en políticas claras, recursos suficientes, capacitación continua y un marco de gobernanza coherente los esfuerzos de implementación suelen fragmentarse, perder continuidad o enfrentar resistencia interna. Esta falta de apoyo estratégico puede limitar la consolidación de una cultura colaborativa y, en consecuencia, comprometer la capacidad de la organización para obtener mejoras sostenibles en la calidad del software. Por ello, el liderazgo ejecutivo se reconoce como un habilitador crítico que permite alinear los objetivos de negocio con las prácticas DevOps, garantizando su adopción efectiva y su impacto medible en los resultados tecnológicos y operativos (Offermann et al., 2022).

2.2 CONCEPTUALIZACIÓN

En esta parte del trabajo se explican los conceptos básicos que son necesarios para entender la investigación. DevOps es una forma de trabajar que adoptan las organizaciones para que los equipos de desarrollo y operaciones trabajen juntos en lugar de por separado, con el objetivo de lanzar actualizaciones de software de manera continua (Ebert et al., 2016). Esta manera de trabajar se inspira en los principios ágiles, donde la

idea central es que todos colaboren estrechamente, automatizando tareas donde sea posible y compartiendo la responsabilidad de los resultados (Humble & Molesky, 2011, como se cita en Peña Olivero et al., 2024).

El estudio aborda también la calidad del software usando como base la norma ISO/IEC 25010, que lista varias características importantes como qué tan confiable es el sistema, qué tan eficiente resulta, qué tan fácil es usarlo y qué tan seguro es, para poder determinar si cumple con lo que se espera (ISO/IEC, 2023). Cada concepto que viene a continuación tiene relación directa con lo que se va a medir en este estudio.

2.2.1 DEVOPS

Cuando hablamos de DevOps nos referimos a juntar dos áreas que tradicionalmente trabajaban separadas: desarrollo y operaciones. La idea es que colaboren todo el tiempo para poder sacar nuevas versiones de las aplicaciones (Ebert et al., 2016). Lo que se pretende lograr con esto es que todo lo relacionado con crear y actualizar software sea más rápido, desde revisar el código hasta probarlo y vigilar cómo funciona, para así poder entregar cosas nuevas o mejoras más seguido (Gómez-Aguirre et al., 2023). Hay varios componentes importantes en DevOps: usar código para configurar la infraestructura, actualizar el software de forma continua, estar monitoreando constantemente cómo va todo, y mantener una comunicación constante entre los equipos.

2.2.2 INTEGRACIÓN CONTINUA Y ENTREGA CONTINUA (CI/CD)

Estas dos formas de trabajar son muy importantes en DevOps cuando queremos asegurar que el software tenga buena calidad. La integración continua funciona así: cuando alguien del equipo hace cambios en el código, esos cambios se juntan automáticamente con el código principal y de inmediato se corren pruebas para ver que todo siga funcionando bien (Aguirre Flórez, 2025). La entrega continua va todavía más allá porque automatiza también el proceso de llevar ese código hasta el ambiente donde realmente se usa. Según explican Humble y Farley (2010), estos procesos que funcionan solos van revisando constantemente que cada versión nueva cumpla con lo que debe cumplir antes de liberarla. Así se evitan muchos errores que se cometerían haciéndolo manualmente, y se pueden detectar fallas más temprano, lo que al final hace que el producto salga mejor.

2.2.3 INFRAESTRUCTURA COMO CÓDIGO (IAC)

La Infraestructura como Código representa una forma distinta de gestionar la infraestructura tecnológica. Morris (2016) explica que IaC consiste en aplicar prácticas de desarrollo de software para automatizar la infraestructura, utilizando código para definir y administrar recursos en lugar de hacerlo manualmente. Este enfoque trae beneficios importantes como la repetibilidad, que permite recrear ambientes idénticos cuando se necesite; el versionamiento, que facilita controlar y rastrear los cambios; y la capacidad de probar las configuraciones antes de aplicarlas. Todo esto contribuye a que el software sea más consistente y opere de forma más eficiente (Wittig & Wittig, 2018).

2.2.4 CONTENERIZACIÓN

La contenerización es una tecnología que empaqueta aplicaciones junto con todo lo que necesitan para funcionar, creando unidades que se pueden mover fácilmente entre diferentes entornos. A diferencia de las máquinas virtuales tradicionales, los contenedores son más livianos porque comparten el sistema operativo de la máquina donde se ejecutan. Esta tecnología se ha vuelto importante para DevOps porque ofrece varias ventajas. Los contenedores aseguran que la aplicación funcione igual en desarrollo, pruebas y producción; mantienen las dependencias aisladas para evitar conflictos; y permiten hacer despliegues más frecuentes con menor riesgo (Burns et al., 2016). Docker se ha convertido en la herramienta más utilizada para crear contenedores, y frecuentemente se combina con Kubernetes para gestionar muchos contenedores al mismo tiempo (Matthias & Kane, 2018).

2.2.5 CALIDAD DEL SOFTWARE SEGÚN ISO/IEC 25010

El estándar ISO/IEC 25010 proporciona un marco para evaluar qué tan bien un sistema de software cumple con lo que esperan sus usuarios en condiciones reales de uso. Esta evaluación no se limita únicamente a verificar si el software funciona correctamente, sino que abarca aspectos más amplios como su estructura interna, su comportamiento durante la ejecución y la experiencia que ofrece a quienes lo utilizan.

Varios autores coinciden en que para evaluar adecuadamente la calidad del software se deben considerar atributos diversos como el rendimiento, la seguridad, qué tan fácil resulta mantenerlo y su fiabilidad (López et al., 2022). No obstante, en contextos donde se implementan prácticas DevOps, se ha notado que la tendencia ha sido enfocarse más en medir aspectos del proceso de desarrollo, dejando de lado la evaluación directa del producto final.

Este estándar organiza la calidad en ocho características principales, entre las cuales esta investigación se enfoca en tres que resultan particularmente relevantes para el sector financiero: mantenibilidad, seguridad y eficiencia de desempeño.

2.2.6 MANTENIBILIDAD

Cuando hablamos de mantenibilidad nos referimos a qué tan sencillo resulta modificar un software para corregir problemas, mejorarlo o adaptarlo cuando cambian las necesidades o el entorno donde opera. Estudios recientes muestran que muchos equipos que trabajan con DevOps no cuentan con formas claras de medir este aspecto, lo que dificulta identificar cuando el código se vuelve cada vez más complejo y difícil de manejar (Amaro et al., 2023).

El estándar ISO/IEC explica la mantenibilidad como la facilidad con que se pueden hacer cambios efectivos al software (ISO/IEC, 2011). Esto incluye desde arreglar errores hasta ajustar el sistema para cumplir con nuevas regulaciones. Para instituciones financieras, donde las reglas cambian con frecuencia y los sistemas deben actualizarse sin detener las operaciones, este atributo se vuelve fundamental.

2.2.7 SEGURIDAD DEL SOFTWARE

La seguridad tiene que ver con la capacidad del software para proteger la información que maneja, asegurando que solo las personas autorizadas puedan acceder a ella y que los datos no sean alterados de manera indebida (ISO/IEC, 2023). En ambientes DevOps, donde los cambios al software ocurren con mucha frecuencia, mantener la seguridad se vuelve más desafiante pero también más necesario. Los especialistas recomiendan que la seguridad se considere desde el inicio del desarrollo, no como algo que se agrega al final. Esto se logra incorporando análisis automáticos que revisen el código, pruebas continuas de seguridad y revisión de las librerías y componentes externos que se utilizan (Prates et al., 2019).

El enfoque DevSecOps surge precisamente para integrar estos controles de seguridad directamente en los procesos automatizados de construcción y despliegue del software. Herramientas que analizan el código en busca de vulnerabilidades y que escanean la infraestructura ayudan a detectar y corregir problemas de seguridad mucho más rápido (Şenkal et al., 2023).

2.2.8 EFICIENCIA DEL DESEMPEÑO

La eficiencia del desempeño se relaciona con qué tan rápido responde el sistema, cuántos recursos consume y cómo se comporta cuando muchos usuarios lo utilizan al mismo tiempo. En ambientes con despliegues frecuentes, es importante asegurarse de que los cambios no afecten negativamente el rendimiento del sistema. El estándar ISO/IEC describe este aspecto como la relación entre el desempeño del sistema y los recursos que necesita para funcionar (ISO/IEC, 2011). Esto incluye los tiempos de respuesta, el uso de memoria y procesador, y la capacidad máxima del sistema para manejar cargas de trabajo. En el sector financiero, donde se procesan muchas transacciones diariamente, el sistema debe mantener buenos tiempos de respuesta incluso en momentos de alta demanda. Por eso resulta importante incluir pruebas de rendimiento en los procesos automatizados, para detectar problemas antes de que los cambios lleguen a producción.

2.3 TEORÍAS DE SUSTENTO

En esta parte del trabajo se explican las bases teóricas que ayudan a entender por qué tendría sentido que las prácticas DevOps se relacionen con la calidad del software. Se van a presentar teorías que se estudiaron durante la maestría en Gestión de Tecnologías de la Información, y también se va a revisar cómo otros investigadores han trabajado estos temas y qué herramientas han usado.

2.3.1 BASES TEÓRICAS

Cuando uno revisa los estudios sobre DevOps, encuentra dos formas diferentes de verlo respecto a la calidad del software:

- Métricas DORA: enfoque en velocidad.

La primera forma, más común en la industria, ve a DevOps como prácticas técnicas para automatizar y hacer todo más rápido. Forsgren y Kersten (2018) desarrollaron las métricas DORA, que miden cuatro aspectos: cada cuánto se publican versiones, cuánto tarda un cambio en llegar a producción, qué tan rápido se recupera el sistema cuando falla, y cuántos cambios causan problemas. El equipo DORA, que después Google compró, lleva más de diez años investigando esto. Han encuestado a muchísimos profesionales y encontraron que los equipos buenos en estas métricas tienen el doble de probabilidades de lograr lo que la organización se propone (Forsgren et al., 2018).

- Teoría de sistemas complejos adaptativos: enfoque en calidad del producto.

La segunda perspectiva es más reciente y menos estudiada. Propone que DevOps no debería juzgarse solo por velocidad, sino también por cómo afecta las características propias del software. Esta teoría explica que los equipos van modificando sus formas de trabajar según la retroalimentación que reciben del entorno (Krebs, 2008). Cuando una organización tiene más madurez DevOps, tiene mejor capacidad para adaptarse, y eso se nota en que el software evoluciona más fácilmente, responde mejor a amenazas nuevas y usa los recursos de forma más inteligente.

Mishra y Otaiwi (2020) revisaron muchos estudios y encontraron que la mayoría se concentran en automatización, cultura y entrega continua. Según ellos, hace falta más investigación sobre métricas específicas de calidad del producto: usabilidad, eficiencia, mantenibilidad, portabilidad. Esta perspectiva se apoya en el modelo ISO/IEC 25010, que define características de calidad medibles independientemente de cómo trabaje cada empresa.

- Modelo de madurez de capacidades de Integración (CMMI).

Este modelo propone que las organizaciones evolucionan a través de niveles de madurez caracterizados por institucionalización creciente de procesos (SEI, 2010). Leppänen et al. (2015) proponen un modelo de madurez DevOps que adapta conceptos CMMI, identificando etapas desde adopción inicial hasta integración holística con mejora continua basada en métricas. La teoría CMMI sugiere que mayor madurez en procesos correlaciona con mayor calidad del producto.

- Articulación teórica aplicada al fenómeno investigado.

Cuando una organización decide adoptar ciertas prácticas de DevOps, estamos hablando de automatización CI/CD, de infraestructura como código, de contenerización, lo que pasa es que estas prácticas terminan funcionando como mecanismos que pueden llegar a influir en distintos atributos de calidad que tiene el producto: su mantenibilidad, su seguridad, su eficiencia.

Tomemos como ejemplo la automatización CI/CD. Según lo que reportan Şenkal et al. (2023), esto hace que se puedan detectar defectos más temprano en el proceso y además facilita que se haga pruebas de manera continua. Al final, esto impacta de forma directa en qué tan confiable es el código y qué tan fácil resulta mantenerlo después. Si miramos lo que pasa con la infraestructura como código, ahí se incorporan principios de

versionamiento, de revisión, de reproducibilidad, y todo eso va mejorando la consistencia del software y también su eficiencia operacional una vez que ya está desplegado. Y bueno, con la contenerización, usando tecnologías como Docker o como Kubernetes, lo que se logra es estandarizar los ambientes donde va a correr el software, y eso ayuda bastante con dos cosas: la portabilidad y la compatibilidad del sistema. Estas dos son características que ISO/IEC 25010 menciona de manera explícita (López et al., 2022).

2.3.2 METODOLOGÍAS DESARROLLADAS POR OTROS INVESTIGADORES

Cuando se revisa la literatura científica sobre qué metodologías se han usado para investigar DevOps y calidad de software, aparecen tres aproximaciones metodológicas que son las predominantes: los estudios de caso, las encuestas dirigidas a profesionales, y los análisis empíricos que se hacen sobre repositorios de código.

- **Estudios de caso múltiples.**

Los estudios de caso fueron la metodología que más se utilizó en las primeras investigaciones sobre DevOps. Por ejemplo, Lwakatare et al. (2019) llevaron a cabo un estudio de casos múltiples donde trabajaron con cinco empresas de tamaño mediano y pequeño que se dedican a desarrollar aplicaciones web. Para recolectar la información realizaron 26 entrevistas con personas que trabajan en estas empresas, además de hacer observaciones directamente en el lugar. Esta aproximación metodológica, que se sustenta en las directrices que propuso Yin (2003) para hacer investigación de casos en ingeniería de software, les permitió a estos investigadores obtener descripciones bastante detalladas sobre cómo se implementa DevOps en contextos industriales reales.

El análisis de los datos lo hicieron mediante codificación temática de cada caso de manera individual, y después realizaron una síntesis cruzada que les sirvió para identificar qué prácticas eran comunes entre los casos. De forma similar, Senapathi y Buchan (2018) realizaron un estudio de caso en profundidad en Nueva Zelanda donde entrevistaron a seis ingenieros de software que fueron monitoreando cómo se implementaban gradualmente los principios DevOps. Lo que documentaron fue notable: un incremento en la frecuencia de despliegue que pasó de 30 lanzamientos al mes a 120 lanzamientos mensuales.

- **Encuestas a profesionales.**

La segunda aproximación metodológica que aparece con frecuencia, sobre todo cuando se trata de estudios de gran escala, son las encuestas a profesionales. Un ejemplo importante de esto es el equipo DORA, que desde 2013 ha venido realizando encuestas cada año. El más reciente es el State of DevOps Report 2024, donde se encuestó a más de 39,000 profesionales de todo el mundo con el objetivo de identificar qué capacidades distinguen a los equipos de alto desempeño (Forsgren et al., 2024).

Lo bueno de esta metodología que se basa en encuestas es que permite recolectar datos de una muestra amplia y diversa, lo cual facilita hacer análisis estadísticos robustos y poder hacer generalizaciones a nivel de toda la industria. Mohammad (2017) también usó un diseño de encuesta para investigar de qué manera DevOps mejora la calidad del software, y para eso distribuyó cuestionarios a más de 300 profesionales que trabajan en empresas donde se práctica DevOps. En su estudio utilizó análisis de correlación de Pearson junto con regresión lineal múltiple para poner a prueba hipótesis sobre cómo se relacionan ciertos factores DevOps (automatización, cultura, medición y compartición) con la calidad del software. Lo que encontró fueron correlaciones fuertes que resultaron estadísticamente significativas al 99% de confianza.

- **Análisis empíricos de repositorios.**

La tercera aproximación metodológica es más reciente y se está volviendo cada vez más común: el análisis empírico de repositorios de código y sistemas de control de versiones. Rzig et al. (2022) realizaron un estudio empírico bastante grande sobre 4,031 proyectos de aprendizaje automático (ML) y 4,076 proyectos que no eran de ML, todos alojados en GitHub. El objetivo era analizar cómo se adoptan las herramientas DevOps, qué esfuerzos de mantenimiento se requieren y qué beneficios se obtienen.

Esta metodología tiene una ventaja importante: permite recolectar métricas objetivas de forma automatizada directamente desde los sistemas de versionamiento, lo que ayuda a minimizar los sesgos que suelen estar asociados con datos que las personas reportan sobre sí mismas. El análisis que hicieron incluyó revisar configuraciones de archivos DevOps y analizar los cambios en los commits para poder caracterizar los comportamientos de desarrollo.

2.3.3 INSTRUMENTOS UTILIZADOS

Si hablamos de las técnicas analíticas que se usan, la literatura muestra que predominan los métodos estadísticos correlacionales y comparativos. Los estudios cuantitativos recurren de manera consistente al análisis de correlación de Pearson o de Spearman cuando quieren examinar asociaciones entre variables continuas, y esto se complementa con regresión lineal múltiple en aquellos casos donde se busca establecer cuánto contribuye cada uno de múltiples predictores.

En cuanto a los instrumentos de medición, la forma de extraer métricas de calidad del software cambia dependiendo de qué característica se esté evaluando: cuando se trata de mantenibilidad, se usan métricas estáticas de código como la complejidad ciclomática, la profundidad de herencia y el acoplamiento; para seguridad, lo que se hace es analizar vulnerabilidades que se identifican mediante herramientas de análisis estático (SAST) y dinámico (DAST); y cuando se evalúa eficiencia, se miden cosas como los tiempos de respuesta, cuántos recursos consume el sistema y su rendimiento.

Por otro lado, las prácticas DevOps se cuantifican usando indicadores que pueden observarse, tales como el porcentaje de pruebas que están automatizadas, con qué frecuencia se ejecutan los pipelines de CI/CD, qué porcentaje de la infraestructura está codificada, y el grado de contenerización que tienen los servicios. Forsgren et al. (2018) desarrollan y validan un instrumento para medir capacidades DevOps que ha demostrado consistencia interna (α de Cronbach > 0.70) y validez convergente con medidas objetivas de rendimiento organizacional.

Wagner et al. (2012) proponen operacionalizaciones de características ISO/IEC 25010 mediante cuestionarios dirigidos a desarrolladores. Para mantenibilidad, incluyen ítems sobre facilidad de comprensión del código, tiempo requerido para implementar cambios, y riesgo de introducir defectos. Para seguridad, consultan sobre procesos de análisis de vulnerabilidades y cumplimiento de estándares. Para eficiencia, indagan sobre tiempos de respuesta y utilización de recursos bajo carga.

2.4 MARCO LEGAL

El desarrollo de software en entidades financieras de Honduras está sujeto a regulaciones que establecen requisitos sobre gestión tecnológica, seguridad de la información y calidad de sistemas. Aquí se identifican las principales normas aplicables.

2.4.1 COMISIÓN NACIONAL DE BANCOS Y SEGUROS (CNBS)

La CNBS es quien regula y supervisa el sistema financiero hondureño. A través de resoluciones y circulares establece requisitos sobre riesgos tecnológicos y operacionales.

Circular CNBS No. 025/2022 sobre Normas para la Gestión de las Tecnologías de la Información establece directrices para la administración de tecnologías de información y comunicaciones en instituciones del sistema financiero. Esta normativa regula los servicios financieros y operaciones realizadas por medio de redes electrónicas, tanto de uso externo como interno. La Junta Directiva de las instituciones financieras debe prestar especial atención a la administración del riesgo derivado de los sistemas de información, considerando que su continuidad, desarrollo y funcionamiento constituyen elementos centrales para la operatividad institucional.

Esta circular es directamente relevante para DevOps porque establece requisitos sobre gestión de cambios en sistemas, control de configuraciones, pruebas adecuadas previo a implementación de modificaciones, y segregación apropiada de funciones entre equipos de desarrollo y operaciones. La automatización CI/CD puede facilitar controles de cambio documentados y reproducibles, mientras que la colaboración DevOps debe reconciliarse con los requisitos de segregación de funciones que establece la normativa.

Resolución SB No. 1320/02-08-2011 sobre Normas de Gestión Integral de Riesgos requiere que las instituciones financieras identifiquen, midan, controlen y monitoreen riesgos asociados a procesos, tecnología, personas y eventos externos. Específicamente, requiere planes de continuidad del negocio y recuperación ante desastres. Prácticas DevOps como infraestructura como código y contenedores pueden contribuir a capacidades de recuperación más rápidas al permitir recreación automatizada de ambientes.

2.4.2 MARCO CONSTITUCIONAL Y LEGAL DE PROTECCIÓN DE DATOS

La Constitución de la República de Honduras, en su artículo 76, establece que "se garantiza la protección al honor, a la intimidad personal, familiar y a la propia imagen". El artículo 100 manifiesta que "toda persona tiene derecho a la inviolabilidad y al secreto de las comunicaciones". El artículo 182, numeral 2, reconoce la Garantía Constitucional de Hábeas Data que garantiza el derecho de acceso a la información sobre sí misma.

La Ley de Transparencia y Acceso a la Información Pública (Decreto No. 170-2006), en su artículo 2 numeral 6, establece la obligación de garantizar la protección, clasificación y seguridad de la información pública, incluyendo datos personales confidenciales. El artículo 24 establece que "los datos personales serán protegidos siempre" y regula el acceso y sistematización de archivos personales.

Aunque Honduras cuenta con un anteproyecto de Ley de Protección de Datos Personales que se encuentra en proceso legislativo desde 2018, el marco constitucional y la Ley de Transparencia ya establecen obligaciones sobre protección de datos que aplican al sector financiero. En el contexto de esta investigación, estas normas refuerzan la importancia de la seguridad como atributo de calidad del software. Prácticas DevSecOps que integran análisis de seguridad en pipelines CI/CD pueden facilitar el cumplimiento al detectar tempranamente vulnerabilidades que pondrían en riesgo datos personales.

2.4.3 ESTÁNDARES INTERNACIONALES ADOPTADOS

Aunque no constituyen ley formal, varios estándares internacionales funcionan de facto como requisitos para entidades financieras hondureñas que participan en sistemas de pago internacional o buscan certificaciones de calidad. ISO/IEC 27001:2013 sobre Sistemas de Gestión de Seguridad de la Información es ampliamente adoptado por instituciones financieras hondureñas. Su Anexo A incluye controles sobre gestión de cambios (A.12.1.2), segregación de ambientes de desarrollo y producción (A.12.1.4), y protección de código fuente (A.14.2.4). Estos controles se relacionan directamente con prácticas DevOps y establecen requisitos que deben considerarse al implementar automatización CI/CD y gestión de infraestructura.

PCI-DSS (Payment Card Industry Data Security Standard) es mandatorio para entidades que procesan, almacenan o transmiten datos de tarjetas de pago. Requisitos como el 6.3.2 sobre revisión de código personalizado antes de liberación a producción, y el 6.4 sobre procesos de control de cambios, son directamente relevantes a DevOps. La automatización de análisis de seguridad en pipelines CI/CD puede facilitar evidencia de cumplimiento con estos estándares.

2.4.4 IMPLICACIONES PARA LA INVESTIGACIÓN

El marco legal y regulatorio establece un contexto donde la calidad del software no es solo un objetivo técnico, sino un requisito regulatorio con implicaciones de cumplimiento. La seguridad del software, una de las variables de este estudio, está

mandatada por múltiples normas constitucionales y sectoriales. La mantenibilidad y eficiencia, aunque menos directamente reguladas, sustentan capacidades requeridas como gestión de cambios, continuidad operacional y recuperación ante desastres.

DevOps como conjunto de prácticas no está prescrito ni prohibido por la normativa hondureña. Sin embargo, su implementación debe reconciliarse con requisitos regulatorios existentes. Investigaciones previas en otros contextos regulados documentan que esta reconciliación es posible y que prácticas DevOps bien implementadas pueden incluso facilitar el cumplimiento al proporcionar automatización, trazabilidad y controles consistentes (Lwakatare et al., 2019).

Esta investigación, al examinar la relación entre nivel de DevOps y calidad del producto en una entidad financiera hondureña, contribuye evidencia sobre si adoptar DevOps en contextos regulados se asocia con resultados positivos de calidad que faciliten el cumplimiento normativo. Los hallazgos pueden orientar a otras instituciones financieras del país en la adopción de estas prácticas de manera que cumplan simultáneamente con los objetivos de agilidad, calidad y cumplimiento regulatorio.

CAPÍTULO III. METODOLOGÍA

En este capítulo se explica cómo se hizo la investigación para medir la relación entre prácticas DevOps y calidad del software en la financiera. Se usó un enfoque cuantitativo con diseño no experimental transversal correlacional. Se describe la población, la muestra seleccionada, y los instrumentos que se aplicaron para recoger datos. Para DevOps se midieron tres cosas: automatización de CI/CD, infraestructura como código y uso de contenedores. La calidad del software se evaluó con ISO/IEC 25010 viendo mantenibilidad, seguridad y eficiencia. También se habla de los aspectos éticos y las limitaciones que tuvo el estudio.

3.1 CONGRUENCIA METODOLÓGICA

En esta sección presentamos cómo organizamos la metodología de nuestra investigación. Incluimos la matriz metodológica, las variables que estudiamos y cómo las definimos para poder trabajar con ellas.

3.1.1 MATRIZ METODOLÓGICA

La matriz metodológica nos permite ver de forma clara los objetivos de la investigación, las preguntas que surgen de estos y las variables seleccionadas para el estudio.

Tabla 1. Matriz metodológica

Problema	Objetivo general	Objetivos específicos	Preguntas de investigación	Variable Independiente	Variable dependiente
¿Cuál es la relación entre el grado de implementación de prácticas DevOps y las métricas de calidad del producto de software (mantenibilidad, seguridad y eficiencia) según ISO/IEC 25010 en una entidad financiera?	Determinar la relación entre el grado de implementación de prácticas DevOps y las métricas de calidad del producto de software (mantenibilidad, seguridad y eficiencia) según ISO/IEC 25010 en una entidad financiera.	Cuantificar el nivel de implementación de prácticas DevOps mediante indicadores de automatización CI/CD, infraestructura como código y contenerización en la entidad financiera.	¿Cuál es el nivel de implementación de prácticas DevOps (automatización CI/CD, infraestructura como código y contenerización) en la entidad financiera?	Nivel de implementación de prácticas DevOps	Calidad del producto de software según ISO/IEC 25010
		Analizar la correlación entre la implementación de automatización de integración continua y las métricas de mantenibilidad del software en la entidad financiera.	¿Qué relación existe entre la implementación de automatización de integración continua y las métricas de mantenibilidad del software en la entidad financiera?		
		Evaluar la relación entre infraestructura como código y contenerización con las métricas de eficiencia de desempeño del software en la entidad financiera.	¿Qué relación existe entre el uso de infraestructura como código y contenerización con las métricas de eficiencia del software en la entidad financiera?		
		Examinar la relación entre los procesos de despliegue automatizado y las métricas de seguridad del producto de software de la entidad financiera.	¿Qué relación existe entre los procesos de despliegue automatizado y las métricas de seguridad del producto de software de la entidad financiera?		

Fuente: Elaboración propia (2026).

La variable independiente en este estudio tiene que ver con el nivel de implementación de prácticas DevOps que hay en la entidad financiera que se analiza. Desde una perspectiva conceptual, esta variable se puede definir como el grado de adopción y madurez que tienen las prácticas técnicas y culturales que son características del enfoque DevOps. Nos referimos específicamente a aquellas prácticas relacionadas con automatizar procesos, con la integración y el despliegue continuo, y con cómo se gestiona la infraestructura. Ebert et al. (2016) explican que DevOps representa un conjunto de prácticas que lo que buscan es integrar el desarrollo y las operaciones. Esto se hace mediante automatización, colaboración que no se detiene, y mejora iterativa, todo con el propósito de acelerar la entrega de software, pero sin comprometer su estabilidad.

Para los efectos operativos de esta investigación, el nivel de implementación de prácticas DevOps se divide en tres dimensiones que son fundamentales. La primera es la

automatización de integración y despliegue continuo (CI/CD). Esto incluye qué tan seguido se ejecutan los pipelines automatizados y qué porcentaje de cobertura tienen las pruebas automatizadas. La segunda dimensión es la infraestructura como código (IaC), que abarca el grado en que la infraestructura está codificada y también el uso que se da a herramientas de gestión de configuración. Y la tercera es la contenerización, que mide el porcentaje de servicios que usan la estrategia de contenedores para ser desplegados.

La variable dependiente en este caso corresponde a la calidad del producto de software. Según el modelo ISO/IEC 25010:2011, esto se puede conceptualizar como el conjunto de características que determinan la capacidad que tiene el software para satisfacer necesidades que están declaradas y también las implícitas, todo bajo condiciones que están especificadas. Para esta investigación en particular, se eligieron tres características de calidad del modelo ISO/IEC 25010 que resultan especialmente relevantes cuando hablamos del contexto de entidades financieras: la mantenibilidad, la seguridad y la eficiencia de desempeño.

La mantenibilidad se define como la capacidad que tiene el producto de software para ser modificado de forma efectiva y eficiente. Esto incluye correcciones, mejoras o adaptaciones que se le hacen al software cuando hay cambios en el entorno (ISO/IEC, 2011). La seguridad tiene que ver con el grado en que el producto protege la información y los datos para que personas o sistemas no autorizados no puedan leerlos o modificarlos. Y la eficiencia de desempeño representa el rendimiento que tiene en relación a la cantidad de recursos que se utilizan bajo condiciones que están establecidas.

3.1.2 ESQUEMA DE VARIABLES DE ESTUDIO

El diagrama muestra cómo se conectan las variables de esta investigación. La variable independiente mide básicamente qué tan implementado está DevOps en la financiera donde se realizó el levantamiento de datos. Para ello, el estudio se concentró en evaluar tres dimensiones distintas que son importantes. La automatización CI/CD es la primera, donde se revisaron aspectos como cuántas veces corren los pipelines y qué porcentaje del código tiene pruebas automáticas. Luego está la infraestructura como código, que tiene que ver con si están usando herramientas para gestionar servidores con scripts en vez de hacerlo todo manual. Y finalmente se analiza la contenerización, que consiste en verificar si se usa Docker u otras tecnologías similares para empaquetar aplicaciones.

La segunda variable, que sería la dependiente, corresponde a calidad del software. Para medirla, la investigación toma ISO/IEC 25010 como marco de referencia, priorizando mantenibilidad, seguridad y eficiencia porque son características que tienen bastante peso en organizaciones financieras. Lo que el modelo sugiere es que debería existir algún tipo de asociación entre estas dos variables.

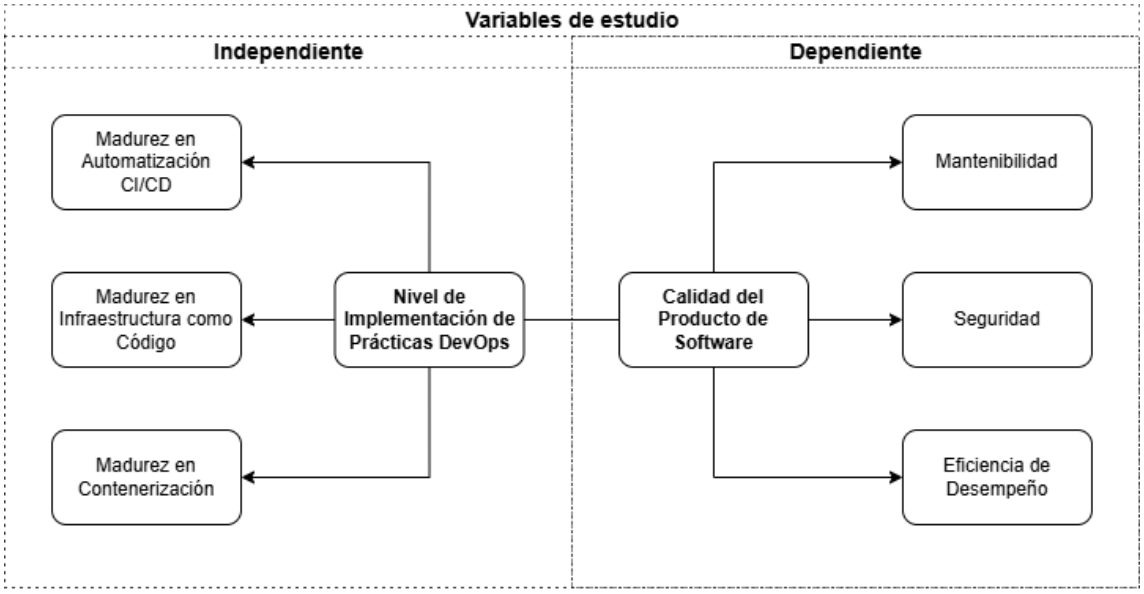

Figura 1 Esquema de variables de estudio
Fuente: Elaboración propia (2026).

3.1.3 OPERACIONALIZACIÓN DE LAS VARIABLES

Para la variable independiente (nivel de implementación de prácticas DevOps), se adoptaron las tres dimensiones operacionales definidas en la matriz metodológica del presente estudio: automatización de integración y despliegue continuo (CI/CD), infraestructura como código (IaC) y contenerización. Estas dimensiones son consistentes con las prácticas técnicas clave reportadas en estudios empíricos sobre DevOps en entornos organizacionales (Ebert et al., 2016; Mishra & Otaiwi, 2020). En cuanto a la variable dependiente (calidad del producto de software), se consideraron los tres atributos del estándar ISO/IEC 25010 que resultan más relevantes para el contexto de una sociedad financiera: mantenibilidad, seguridad y eficiencia de desempeño.

Cada indicador fue operacionalizado mediante ítems diseñados para un cuestionario estructurado con escala Likert de cinco puntos, orientado a profesionales de TI que participan en el ciclo de vida del software. Los instrumentos utilizados serán cuestionarios estructurados, sometidos a prueba piloto para asegurar su validez y claridad. La técnica de recolección será la encuesta online, lo que garantiza accesibilidad y

confidencialidad en el proceso de respuesta. Las escalas de tipo Likert permiten medir las percepciones de manera ordinal, facilitando su análisis estadístico posterior en relación con las hipótesis planteadas.

Tabla 2. Operacionalización de variables – Automatización CI/CD

Variable Independiente	Definición		Dimensión	Indicador	Preguntas	Respuestas	Escala	Técnica
	Conceptual	Operacional						
Nivel de implementación de prácticas DevOps	Enfoque socio-técnico basado en automatización, colaboración y entrega continua que integra desarrollo y operaciones.	Se mide mediante el puntaje total obtenido en los ítems asociados a las tres dimensiones DevOps: CI/CD, IaC y Contenerización.	Automatización CI/CD	Integración continua	P1. Los cambios que los desarrolladores hacen en el código se integran automáticamente en un repositorio central compartido por todo el equipo.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Pipelines	P2. Cada vez que se hace un cambio en el código, se ejecutan automáticamente procesos que compilan, prueban y validan el software sin intervención manual.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Pruebas	P3. Las pruebas automatizadas validan de forma completa los cambios más importantes y críticos del sistema antes de llegar a producción.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Detección temprana de errores	P4. Los errores en el código se detectan automáticamente en las primeras etapas del proceso, antes de que el software llegue a los usuarios finales.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Visibilidad de resultados CI/CD	P5. Los resultados de las pruebas y validaciones automáticas son visibles y accesibles para todos los miembros del equipo técnico.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	

Fuente: Elaboración propia (2026).

Tabla 3. Operacionalización de variables – Infraestructura como Código

Variable Independiente	Definición		Dimensión	Indicador	Preguntas	Respuestas	Escala	Técnica
	Conceptual	Operacional						
Nivel de implementación de prácticas DevOps	Enfoque socio-técnico basado en automatización, colaboración y entrega continua que integra desarrollo y operaciones.	Se mide mediante el puntaje total obtenido en los ítems asociados a las tres dimensiones DevOps: CI/CD, IaC y Contenerización.	Infraestructura como Código (IaC)	Scripts	P6. La infraestructura tecnológica (servidores, redes, bases de datos) se define y despliega usando archivos de código o plantillas automatizadas en lugar de configuraciones manuales.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Replicacion	P7. Los ambientes de desarrollo, calidad y producción se crean de forma idéntica utilizando las mismas definiciones de código, garantizando que funcionen igual.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Versionamiento	P8. Los cambios en la infraestructura se registran y versionan en un sistema de control (como Git), de la misma manera que se hace con el código del software.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Documentación sincronizada	P9. La documentación técnica de la infraestructura se mantiene actualizada y está sincronizada con las definiciones de código que realmente se utilizan.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Revisión técnica de cambios	P10. Los cambios propuestos en la infraestructura pasan por un proceso de revisión técnica por otros miembros del equipo antes de ser aplicados.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	

Fuente: Elaboración propia (2026).

Tabla 4. Operacionalización de variables – Contenerización

Variable Independiente	Definición		Dimensión	Indicador	Preguntas	Respuestas	Escala	Técnica
	Conceptual	Operacional						
Nivel de implementación de prácticas DevOps	Enfoque socio-técnico basado en automatización, colaboración y entrega continua que integra desarrollo y operaciones.	Se mide mediante el puntaje total obtenido en los ítems asociados a las tres dimensiones DevOps: CI/CD, IaC y Contenerización.	Contenerización	Despliegue	P11. Las aplicaciones y servicios se empaquetan y despliegan de forma estandarizada en contenedores (tecnologías como Docker o similares)	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Consistencia operativa	P12. El uso de contenedores ha logrado que las aplicaciones se comporten de manera consistente sin importar si están en desarrollo, calidad o producción.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Rapidez del despliegue	P13. Las nuevas versiones del software se despliegan de forma más rápida y confiable gracias al uso de contenedores.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Orquestacion	P14. Existe una gestión efectiva de los contenedores que permite controlar automáticamente aspectos como el escalamiento, la disponibilidad y el balanceo de carga.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Monitoreo continuo	P15. El monitoreo continuo de los contenedores permite al equipo detectar y diagnosticar problemas operativos antes de que afecten a los usuarios.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	

Fuente: Elaboración propia (2026).

Tabla 5. Operacionalización de variables – Mantenibilidad

Variable Dependiente	Definición		Dimensión	Indicador	Preguntas	Respuestas	Escala	Técnica
	Conceptual	Operacional						
Calidad del producto de software	Grado en que el software satisface los atributos definidos por ISO/IEC 25010.	Se mide mediante los puntos obtenidos en ítems de mantenibilidad, eficiencia y seguridad.	Mantenibilidad	Facilidad de cambio	P16. Cuando se necesita hacer cambios en el software, estos se pueden implementar sin causar problemas o errores inesperados en otras partes del sistema.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Organización del código	P17. La forma en que está organizado el código permite localizar rápidamente las partes que necesitan modificarse y entender cómo funcionan.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Corrección eficiente de errores	P18. Cuando se encuentran errores en el sistema, estos se pueden identificar, diagnosticar y corregir de forma eficiente sin comprometer la estabilidad general.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Deuda técnica	P19. El equipo gestiona de forma continua la deuda técnica del sistema mediante actividades planificadas de mejora y refactorización del código.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Documentación técnica	P20. El código fuente incluye documentación técnica que ayudan a comprender su funcionamiento - y facilitan el trabajo de mantenimiento futuro.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	

Fuente: Elaboración propia (2026).

Tabla 6. Operacionalización de variables – Eficiencia del desempeño

Variable Dependiente	Definición		Dimensión	Indicador	Preguntas	Respuestas	Escala	Técnica
	Conceptual	Operacional						
Calidad del producto de software	Grado en que el software satisface los atributos definidos por ISO/IEC 25010.	Se mide mediante los puntos obtenidos en items de mantenibilidad, eficiencia y seguridad.	Eficiencia del desempeño	Tiempo de respuesta	P21. El sistema mantiene tiempos de respuesta adecuados para el negocio, incluso cuando enfrenta diferentes niveles de carga de trabajo.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Uso eficiente de recursos	P22. El software utiliza de manera eficiente los recursos tecnológicos disponibles, sin desperdiciarlos.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Estabilidad del desempeño	P23. El desempeño del sistema se mantiene estable y consistente durante los periodos de mayor demanda operativa del negocio.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Escalabilidad	P24. Cuando se necesita mayor capacidad de procesamiento, el sistema puede crecer de forma proporcional agregando más recursos tecnológicos.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Cumplimiento de niveles de servicio	P25. Los indicadores de desempeño del sistema cumplen de manera consistente con los niveles de servicio que se han establecido para el negocio.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	

Fuente: Elaboración propia (2026).

Tabla 7. Operacionalización de variables – Seguridad

Variable Dependiente	Definición		Dimensión	Indicador	Preguntas	Respuestas	Escala	Técnica
	Conceptual	Operacional						
Calidad del producto de software	Grado en que el software satisface los atributos definidos por ISO/IEC 25010.	Se mide mediante los puntos obtenidos en items de mantenibilidad, eficiencia y seguridad.	Seguridad	Detección de vulnerabilidades	P26. Las vulnerabilidades de seguridad en el software se identifican y corrigen de forma oportuna antes de que el sistema llegue a producción.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Control de accesos	P27. Los mecanismos de autenticación y control de acceso aseguran que solo los usuarios autorizados puedan acceder a los recursos según el nivel de privilegios que les corresponde.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Efectividad DevSecOps	P28. Las prácticas de seguridad (análisis de código, revisión de dependencias, pruebas de seguridad) están integradas en todas las etapas del proceso de desarrollo.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Protección de datos	P29. Las configuraciones de seguridad del sistema protegen de manera efectiva la confidencialidad, integridad y disponibilidad de la información.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	
				Auditorias	P30. Se realizan auditorías y evaluaciones de seguridad de forma regular para verificar que se cumplen las políticas y controles establecidos.	Nunca	1	Encuesta
						Casi nunca	2	
						Algunas veces	3	
						Casi siempre	4	
						Siempre	5	

Fuente: Elaboración propia (2026).

3.1.4 HIPÓTESIS

Tomando en cuenta que esta investigación tiene un alcance correlacional, se han planteado las hipótesis que van a guiar el análisis de los datos que se recolecten.

- **Hipótesis de investigación**

H_1: El nivel de implementación de prácticas DevOps en la sociedad financiera, el cual se mide considerando tres dimensiones (automatización CI/CD, infraestructura como código y contenerización), está relacionado de manera positiva con la calidad del

producto de software, evaluada a través de los atributos de mantenibilidad, seguridad y eficiencia de desempeño definidos en la norma ISO/IEC 25010.

- **Hipótesis nula**

H_0: El nivel de implementación de prácticas DevOps en la sociedad financiera no presenta relación estadísticamente significativa con la calidad del producto de software, cuando esta calidad se evalúa usando los atributos de mantenibilidad, seguridad y eficiencia de desempeño que define ISO/IEC 25010.

3.2 ENFOQUE Y MÉTODOS

La presente investigación se desarrolla bajo un enfoque cuantitativo, dado que busca analizar el fenómeno de estudio a partir de datos medibles y susceptibles de tratamiento estadístico. Este enfoque resulta factible para examinar la relación existente entre el grado de implementación de prácticas DevOps, considerada como la variable independiente, y las métricas de calidad del producto de software, definidas como variables dependientes. En particular, el estudio se centra en los atributos de mantenibilidad, eficiencia y seguridad, establecidos en la norma ISO/IEC 25010, lo que permite estructurar el análisis sobre un marco conceptual reconocido y estandarizado.

El diseño metodológico adoptado es no experimental, ya que no se realiza una manipulación intencional de las variables de estudio. En cambio, estas son observadas tal como se manifiestan en su contexto real, dentro de una entidad financiera que ha incorporado prácticas DevOps en sus procesos de desarrollo y operación de software. Este tipo de diseño es adecuado cuando el investigador busca describir y analizar relaciones existentes sin interferir en las condiciones organizacionales, lo cual es especialmente relevante en entornos regulados donde la experimentación directa resulta limitada.

En cuanto al alcance de la investigación, este es de tipo correlacional, ya que el objetivo principal consiste en identificar y analizar la relación estadística entre la madurez en la adopción de prácticas DevOps y los niveles percibidos de calidad del producto de software. Este enfoque permite evaluar si existe relación entre las variables, sin establecer relaciones de causalidad directa, y es consistente con estudios anteriores en entornos de naturaleza organizacional (Hernández, Fernández & Baptista, 2014). Asimismo, el estudio adopta un diseño transeccional o transversal, puesto que la recolección de datos se realiza en un único período del tiempo. Esta estrategia proporciona una visión puntual

del estado de implementación de DevOps y de la calidad del software en la organización, facilitando la comparación entre diferentes perfiles y áreas funcionales.

Como técnica de recolección de datos se empleará una encuesta estructurada, aplicada a colaboradores de las áreas de desarrollo, pruebas y operaciones. El instrumento utiliza una escala tipo Likert, diseñada en coherencia con las variables conceptuales del estudio, y ha sido sometido a un proceso de validación mediante revisión de expertos y prueba piloto, con el fin de garantizar su claridad, pertinencia y confiabilidad.

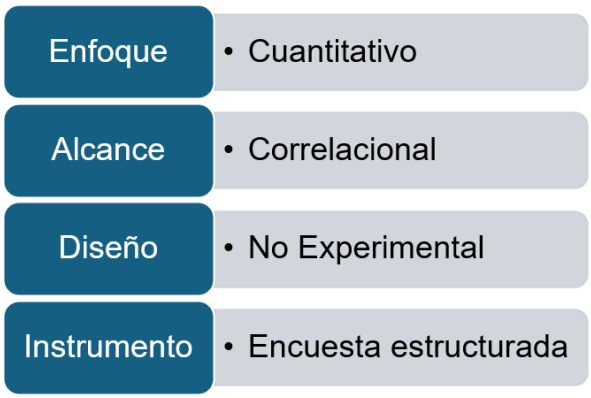

Figura 2. Enfoque y métodos utilizados
Fuente: Elaboración propia (2026).

3.3 DISEÑO DE LA INVESTIGACIÓN

3.3.1 POBLACIÓN

La población objeto de este estudio está conformada por colaboradores que desempeñan funciones en las áreas de desarrollo de software, pruebas y operaciones tecnológicas de una entidad financiera. Estos profesionales participan de manera directa o indirecta en diversas fases del ciclo de vida del software, tales como el análisis de requerimientos, la construcción de soluciones, la ejecución de pruebas, el despliegue entre ambientes, así como las actividades de mantenimiento y monitoreo de aplicaciones que soportan procesos críticos del negocio. En conjunto, la población está integrada por 76 personas, entre las que se incluyen ingenieros de software, personal de infraestructura, especialistas vinculados a prácticas DevOps, líderes técnicos y responsables de equipos de trabajo.

La selección de esta población resulta adecuada para los objetivos de la investigación, dado que sus integrantes interactúan con procesos influenciados por la adopción de prácticas DevOps dentro de la organización. Su participación en flujos de

trabajo que incorporan automatización, integración y despliegue continuo, así como mecanismos de monitoreo y gestión de incidentes, les proporciona una perspectiva operativa relevante.

Asimismo, la experiencia de los participantes en sus respectivos roles facilita la identificación de percepciones relacionadas con la calidad del producto de software, incluyendo aspectos como estabilidad, mantenibilidad, eficiencia y confiabilidad. Desde esta perspectiva, la población seleccionada no solo aporta información sobre la adopción técnica de DevOps, sino también sobre su impacto percibido en los resultados del desarrollo y operación del software, lo cual es especialmente relevante en un contexto financiero caracterizado por altos requerimientos de control, seguridad y continuidad operativa. La inclusión de perfiles técnicos y jerárquicos diversos permite contrastar puntos de vista entre áreas y niveles de responsabilidad, contribuyendo a fortalecer la validez del estudio, al permitir examinar la relación entre las prácticas DevOps y la calidad del producto de software desde múltiples perspectivas organizacionales.

3.3.2 MUESTRA

El tamaño de la muestra fue determinado mediante la aplicación de la fórmula estadística para poblaciones finitas con estimación de proporciones, la cual es aplicada cuando se busca obtener resultados representativos en estudios de carácter descriptivo y correlacional. Para este cálculo se estableció un nivel de confianza del 95% y un margen de error del 5%, parámetros ampliamente aceptados en investigaciones sociales y organizacionales orientadas a analizar percepciones y prácticas profesionales (Hernández, Fernández & Baptista, 2014). La elección de estos valores permite equilibrar la precisión de los resultados con la viabilidad operativa del estudio.

La fórmula empleada se expresa de la siguiente manera:

$$n = \frac{z^2 * p * q * N}{e^2 * (N - 1) + z^2 * p * q}$$

Donde N representa el tamaño de la población, Z corresponde al valor asociado al nivel de confianza seleccionado, P y Q representan la probabilidad de ocurrencia y no ocurrencia del fenómeno analizado, respectivamente, y e indica el margen de error permitido. En este estudio se consideró un valor poblacional de 76 colaboradores, un valor Z de 1.96 para el nivel de confianza del 95%, una proporción P = 0.5 y Q = 0.5 con el fin de asumir máxima variabilidad, y un margen de error de 0.05.

$$n = \frac{1.96^2 \cdot 0.5 \cdot 0.5 \cdot 76}{(1.96^2 \cdot 0.5 \cdot 0.5) + (0.05^2 \cdot (76-1))} = 64$$

Al sustituir estos valores en la fórmula, se obtuvo un tamaño de muestra aproximado de 64 participantes. Este número se considera suficiente para garantizar la validez estadística de los resultados, permitiendo inferencias confiables sobre la población estudiada dentro de los límites de error establecidos. La selección de la muestra se realizará considerando la disponibilidad de los colaboradores y su participación en actividades relacionadas con el desarrollo, las pruebas y las operaciones de software. Este criterio busca asegurar la inclusión de distintos perfiles funcionales, favoreciendo la diversidad de perspectivas y manteniendo la coherencia metodológica con los objetivos de la investigación.

3.3.3 TÉCNICAS DE MUESTREO

Para el desarrollo de la presente investigación se empleará una técnica de muestreo probabilístico al azar simple, seleccionada en función de las características del contexto organizacional y de las condiciones operativas propias de la entidad financiera objeto de estudio. De acuerdo con Hernández, Fernández y Baptista (2014), el muestreo probabilístico permite que todos los elementos de la población tengan la misma probabilidad de ser seleccionados, esto ayuda a reducir sesgos. En este caso, la población objetivo está compuesta por 76 colaboradores que desempeñan funciones en áreas vinculadas al ciclo de vida del software, lo que garantiza la pertinencia del grupo seleccionado respecto a las variables del estudio.

El proceso de selección de la muestra se realizará invitando a participar a la mayor cantidad posible de colaboradores que cumplan con los criterios establecidos, priorizando aquellos perfiles directamente involucrados en actividades de desarrollo, pruebas, operaciones y gestión tecnológica. Adicionalmente, se procurará mantener una diversidad

funcional dentro de la muestra, incorporando distintos roles técnicos y niveles jerárquicos, con el fin de obtener una visión más integral del fenómeno estudiado. Esta estrategia fortalece la validez interna del estudio al permitir contrastar percepciones y experiencias desde múltiples perspectivas organizacionales.

3.3.4 UNIDAD DE ANÁLISIS Y UNIDAD DE RESPUESTA

En la presente investigación, la unidad de análisis y la unidad de respuesta coinciden y están conformadas por los colaboradores del área de Tecnologías de Información de la entidad financiera. Se consideran dentro de este grupo aquellos profesionales que participan directamente en actividades relacionadas con el ciclo de vida del software, tales como desarrollo, pruebas, operaciones, gestión de infraestructura y adopción de prácticas DevOps. La elección del individuo como unidad central del estudio responde a que el análisis estadístico se realiza sobre las respuestas individuales proporcionadas por los mismos sujetos que intervienen en los procesos técnicos evaluados. En este sentido, cada colaborador constituye simultáneamente una unidad de observación y una fuente primaria de información, ya que forma parte del fenómeno investigado y, al mismo tiempo, responde el instrumento de recolección de datos.

La población total del área está integrada por 76 colaboradores, de los cuales 64 completaron válidamente el cuestionario estructurado aplicado para la recolección de información. Estas respuestas constituyen la base empírica del estudio y permiten desarrollar el tratamiento estadístico previsto en el diseño metodológico. Al tratarse de una proporción significativa del universo poblacional, se mantiene un nivel de confianza del 95%, coherente con el enfoque cuantitativo y el alcance correlacional adoptado. Cada cuestionario diligenciado recoge la percepción individual del participante respecto al grado de implementación de prácticas DevOps y a los atributos de calidad del producto de software evaluados, específicamente mantenibilidad, eficiencia y seguridad según la norma ISO/IEC 25010. La información fue recopilada mediante una escala tipo Likert de cinco puntos, bajo condiciones de anonimato y participación voluntaria, garantizando confidencialidad y uso exclusivo con fines académicos.

3.4 TÉCNICAS, INSTRUMENTOS Y PROCEDIMIENTOS APLICADOS

La técnica de recolección de datos será la encuesta por cuestionario. Este instrumento incluirá preguntas basadas en una escala Likert de cinco puntos, siempre alineadas a los indicadores definidos. En el caso del nivel de DevOps, se formularán preguntas sobre la frecuencia de las prácticas DevOps, así como preguntas orientadas a

medir la calidad del software. Se realizará primero un análisis de normalidad para determinar que las preguntas propuestas permitan identificar asociaciones apropiadas entre las variables.

El cuestionario se construirá a partir de literatura especializada, utilizando estudios previos que brindan una base sólida para redactar los ítems de manera adecuada. Por ejemplo, Muhammad et al. (2023) analizan cómo la adopción de prácticas DevOps contribuye a mejorar la confiabilidad y la calidad en sistemas basados en la nube, proporcionando lineamientos claros sobre procesos, automatización y métricas relevantes. Asimismo, Erich, Lübke y van der Werf (2019) presentan una perspectiva arquitectónica de DevOps que describe prácticas, principios y consideraciones técnicas que pueden ser traducidas en indicadores medibles dentro de un cuestionario estructurado. Por otra parte, se incluirán atributos de la norma ISO/IEC 25010 para las preguntas relacionadas con la calidad del software.

La recolección se realizará de manera electrónica, solicitando el apoyo de los colaboradores de la entidad financiera y garantizando que cada uno reciba el instrumento de encuesta. Se informará a cada participante el propósito de la investigación, enfatizando que forma parte de un estudio de maestría y destacando su importancia para el área de TI. El instrumento será aplicado de manera anónima, asegurando no incluir preguntas que permitan identificar a un participante. El uso de este cuestionario estructurado se justifica plenamente dentro del enfoque cuantitativo del estudio, ya que permite recoger datos estandarizados, medibles y comparables sobre la percepción de los participantes respecto a la adopción de prácticas DevOps y la calidad del software.

3.4.1 FUENTES OBJETIVAS COMPLEMENTARIAS DE MEDICIÓN

Con el propósito de situar el diseño metodológico adoptado en su contexto más amplio y de orientar futuras réplicas del estudio hacia diseños de mayor robustez empírica, se identifican a continuación las principales fuentes de datos objetivos que pueden complementar y triangular las percepciones recopiladas mediante el cuestionario estructurado. La literatura metodológica especializada señala que la combinación de datos subjetivos con indicadores objetivos del sistema incrementa la confiabilidad y la capacidad explicativa de los resultados en estudios sobre adopción de prácticas DevOps (Amaro et al., 2023; Rzig et al., 2022).

En el contexto de la dimensión de automatización CI/CD, los registros de ejecución de pipelines disponibles en herramientas como Jenkins, GitLab CI o Azure DevOps permiten extraer métricas directas y verificables: la frecuencia de ejecución del pipeline por período, la tasa de éxito de compilación expresada como porcentaje de builds exitosos sobre el total, el porcentaje de cobertura de pruebas automatizadas reportado por frameworks de testing, y el tiempo promedio desde el commit hasta el despliegue en producción. Estas métricas corresponden en buena medida a los indicadores DORA documentados por Forsgren y Kersten (2018).

Para la dimensión de Infraestructura como Código, los repositorios de Terraform, Ansible o Puppet constituyen fuentes objetivas que permiten determinar el porcentaje de infraestructura codificada versus gestionada manualmente, la frecuencia de cambios en scripts de configuración medida por commits, y el nivel de cobertura de revisión colaborativa expresado en porcentaje de pull requests aprobadas. Para la dimensión de contenerización, los registros de Kubernetes o Docker Swarm ofrecen indicadores como el porcentaje de servicios desplegados en contenedores sobre el total en producción, la tasa de disponibilidad de los pods, y la frecuencia de reinicios no planificados como indicador de estabilidad operativa.

Respecto a los atributos de calidad del software, la herramienta SonarQube permite obtener métricas objetivas de mantenibilidad a través de la complejidad ciclomática, la densidad de code smells y el índice de deuda técnica, operacionalizando de manera directa los sub atributos de analizabilidad y modificabilidad contemplados en la norma ISO/IEC 25010. La seguridad puede evaluarse objetivamente mediante reportes generados por herramientas SAST como Checkmarx o Semgrep, y herramientas DAST como OWASP ZAP, que producen indicadores verificables como el número de vulnerabilidades clasificadas por criticidad y el tiempo promedio de remediación. La eficiencia del desempeño, por su parte, puede medirse a través de plataformas de monitoreo como Dynatrace, Prometheus o Grafana, que registran el tiempo promedio de respuesta transaccional, el percentil 95 bajo carga máxima y el porcentaje de utilización de recursos en producción.

La integración de estas fuentes objetivas en versiones posteriores del estudio permitiría reducir el sesgo de autoinforme documentado por Podsakoff et al. (2003) y elevar el nivel de evidencia desde la percepción profesional informada hacia la medición

directa del sistema, fortaleciendo la base empírica sobre la cual se fundamentan las relaciones estadísticas identificadas.

3.5 FUENTES DE INFORMACIÓN

La presente investigación se utilizará exclusivamente fuentes primarias para la recolección de datos. Esta fuente estará estructurada por la información obtenida directamente de los colaboradores de la entidad financiera a través de la aplicación del cuestionario estructurado. El cuestionario que se implementará en la financiera permitirá medir tanto el nivel de adopción de prácticas DevOps como los atributos de calidad del software desde la perspectiva del personal involucrado en el ciclo de desarrollo y operación.

Es importante señalar que no se utilizarán fuentes secundarias, ya que el diseño metodológico no contempla el uso de reportes institucionales, registros administrativos, ni bases de datos externas. Por otra parte, los documentos normativos como la ISO/IEC 25010 y los artículos científicos utilizados para fundamentar la investigación, no se consideran fuentes secundarias, sino fuentes bibliográficas que sustentan el marco teórico y la definición de las variables del estudio.

3.6 ÉTICA EN LA INVESTIGACIÓN

Esta investigación se llevará a cabo bajo principios éticos estrictos. La idea es garantizar que el trabajo tenga integridad científica y que haya respeto hacia quienes participan, así como hacia la organización donde se hace el estudio.

Consentimiento informado: Lo primero que se hará es obtener una autorización formal de la dirección de TI de la entidad financiera. Esto es necesario para poder acceder a datos técnicos y para aplicar los cuestionarios. Los líderes técnicos van a participar de manera voluntaria, pero antes recibirán información clara sobre cuáles son los objetivos, qué procedimientos se van a seguir y cómo se va a usar la información que proporcionen.

Confidencialidad: Todos los datos que se recolecten van a ser anonimizados. Esto significa que se eliminará cualquier información que pueda permitir identificar proyectos específicos, personas o la entidad financiera en sí. Cuando se presenten los resultados, se hará en forma agregada. El acceso a datos que sean sensibles va a estar restringido únicamente al equipo investigador, y los miembros de este equipo firmarán acuerdos de confidencialidad.

Honestidad y objetividad: Se van a reportar todos los datos que se obtengan, sin manipulación y sin que se omita nada de forma selectiva. Las limitaciones que tenga el estudio serán reconocidas de manera explícita. No se van a establecer conclusiones que vayan más allá de lo que los datos realmente permiten afirmar.

Integridad académica: Se respetarán de manera estricta las normas APA 7ª edición cuando se trate de citas y referencias, evitando así cualquier forma de plagio. En este caso todos los instrumentos que se usen y los procedimientos que se sigan serán documentados, de manera que esto permita que la investigación sea replicable.

Beneficencia y no maleficencia: El propósito de esta investigación es producir conocimiento útil que ayude a mejorar cómo funcionan las cosas en las organizaciones, pero haciendo esto sin perjudicar a las personas que participan y sin interferir con las operaciones que son críticas para la entidad. Los resultados que se lleguen a obtener se compartirán con la organización, de modo que puedan sacar provecho de ellos si así lo consideran conveniente.

3.7 LIMITACIONES DEL ESTUDIO

Al tratarse de un estudio de tipo no experimental, transversal y correlacional, la primera limitación es que no se puede establecer causalidad entre las variables analizadas, sino únicamente identificar asociaciones estadísticas (Hernández, Fernández & Baptista, 2014). Otra limitación relevante radica en el uso de percepciones individuales como fuente de datos, ya que la medición de variables como la calidad del software está basada en autoinforme, lo que puede introducir sesgos de subjetividad o deseabilidad social en las respuestas (Podsakoff et al., 2003). Aunque se utilizarán ítems estructurados y escalas validadas, existe el riesgo de que algunos participantes sobrestimen o subestimen su nivel de madurez DevOps o la percepción de calidad del producto.

También se debe considerar la delimitación del área de estudio. La investigación se realizará únicamente en una entidad financiera específica, lo cual limita la generalización de los hallazgos a otros sectores, regiones o instituciones con diferentes niveles de madurez tecnológica y estructuras organizativas (Bryman, 2012). Asimismo, al aplicar un muestreo no probabilístico por conveniencia, no se garantiza representatividad estadística, lo que podría restringir la extrapolación de los resultados a toda la población objetivo.

Por último, pueden surgir limitaciones operativas o logísticas como baja tasa de respuesta, resistencia institucional a compartir información o dificultades para aplicar el instrumento en todos los equipos previstos. Estos factores pueden afectar la completitud o balance de la muestra, y se tomarán en cuenta al interpretar los resultados finales.

CAPÍTULO IV. RESULTADOS Y ANÁLISIS

Este capítulo presenta los resultados observados para las variables nivel de implementación de prácticas DevOps y calidad del producto de software, así como el análisis realizado a partir de los datos recolectados por medio del instrumento y técnica detallados en el capítulo anterior. Del mismo modo, la interpretación del análisis se realiza para cada variable con sus dimensiones e indicadores respectivamente: para DevOps se evalúan integración continua y despliegue continuo, infraestructura como código y gestión de contenedores; para calidad del software se analizan los atributos de mantenibilidad, eficiencia y seguridad según la norma ISO/IEC 25010. Finalmente, se procede a la comprobación de la hipótesis de investigación planteada para este estudio mediante análisis correlacional.

4.1 INFORME DEL PROCESO DE RECOLECCIÓN DE DATOS

La recolección de datos tuvo lugar durante dos semanas en la entidad financiera que autorizó su participación. Para conformar la muestra se recurrió a un procedimiento de muestreo probabilístico al azar simple, buscando que todos los subgrupos tuvieran las mismas condiciones de participación. La población técnica de la institución sumaba 76 colaboradores repartidos en cuatro áreas: 44 en la Vicepresidencia de Innovación, 13 en la Vicepresidencia de Tecnología, 12 en Banca Digital y 7 en Arquitectura IT. Solo se consideró al personal con funciones directas en desarrollo de software o actividades vinculadas a DevOps, quedando fuera quienes desempeñaban roles administrativos u operativos sin relación con esos procesos.

De las 76 personas convocadas, 64 completaron el instrumento de forma satisfactoria, lo que representa una tasa de participación del 84.2%. El cuestionario constó de 30 preguntas organizadas en seis dimensiones, con cinco ítems cada una, usando una escala de cinco niveles que iba desde "Nunca" hasta "Siempre". Las preguntas buscaban capturar tanto el nivel de adopción de prácticas DevOps como la percepción sobre los atributos de calidad del software definidos en el estándar ISO/IEC 25010.

Tabla 8. Mapa de roles de los participantes en la encuesta

Grupo	Roles incluidos	Dimensiones DevOps evaluadas	Atributos de Calidad Evaluados
Desarrollo	Desarrolladores de software Arquitectos de software	CI/CD	Mantenibilidad
Operaciones	Especialistas de Infraestructura SRE Administradores de sistemas	IaC, Contenerización	Eficiencia del desempeño
Transversales	Líder Técnico Desarrollo Líder Técnico DevOps Especialistas en ciberseguridad	CI/CD, IaC, Contenerización	Seguridad

Fuente: Elaboración propia a partir de la estructura organizacional de la sociedad financiera (2026).

Como se observa en la tabla anterior, los 64 participantes se distribuyen en tres categorías funcionales que reflejan los tres pilares del enfoque DevOps: desarrollo, operaciones y roles transversales. La inclusión equilibrada de los tres grupos garantiza que los resultados no reflejen la perspectiva de una sola disciplina, sino una visión organizacional completa del fenómeno estudiado. Esta diversidad fortalece la validez de contenido del instrumento.

Tabla 9. Nivel de confiabilidad

Estadísticas de fiabilidad

Alfa de Cronbach	N de elementos
,957	30

Fuente: Elaboración propia (análisis en SPSS, 2026).

Antes de aplicar el cuestionario, se revisó qué tan consistentes eran sus ítems entre sí, para lo cual se calculó el coeficiente Alfa de Cronbach. El valor obtenido fue de 0.957, cifra que supera con holgura el umbral de 0.70 que la literatura metodológica considera aceptable, y que además rebasa el 0.90 asociado a niveles de excelencia en fiabilidad. Este resultado respalda que el instrumento medía de forma confiable las variables del estudio. Vale señalar que, dado que el trabajo se realizó en una sola institución financiera, los hallazgos reflejan la realidad de ese contexto particular y no corresponde trasladarlos directamente a otras organizaciones del sector.

4.2 PRESENTACIÓN DE RESULTADOS

Los resultados que se presentan a continuación describen, de manera organizada, lo que los 64 profesionales de TI de la entidad financiera reportaron sobre las prácticas DevOps y los atributos de calidad del software. La información sigue el orden de las seis

dimensiones evaluadas y cierra con análisis comparativos que sientan la base para las pruebas estadísticas que se desarrollan más adelante.

4.2.1 ANÁLISIS DETALLADO DE PRÁCTICAS DEVOPS

Nivel de madurez de CI/CD

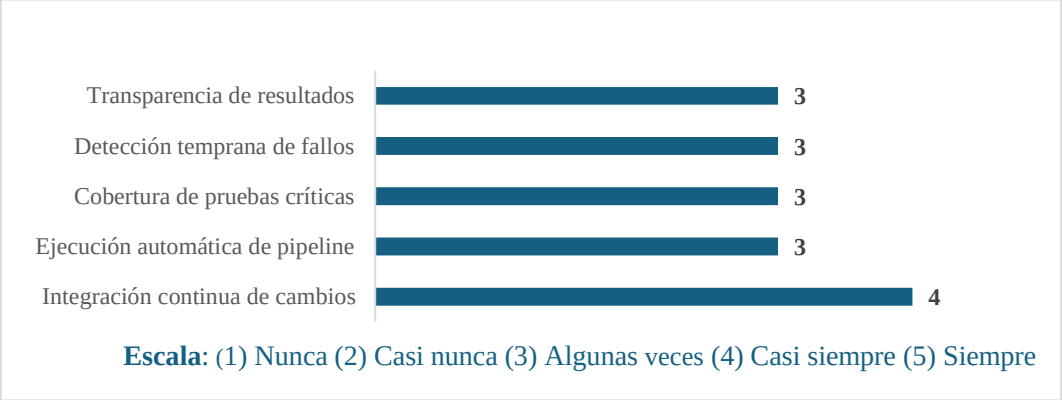

Figura 3. Mediana Dimensión CI/CD
Fuente: Elaboración propia (2026).

La mediana de cada indicador dentro de la dimensión de integración continua y despliegue continuo reveló que la mayor parte de las prácticas evaluadas se ubican en un nivel de ocurrencia intermedio. La transparencia de resultados, la detección temprana de fallos, la cobertura de pruebas críticas y la ejecución automática de pipeline obtuvieron todas mediana de 3, valor que corresponde a "Algunas veces" en la escala utilizada. La excepción fue la integración continua de cambios, que alcanzó mediana de 4 ubicándose en "Casi siempre", lo que la distingue como la práctica con mayor grado de consolidación dentro de esa dimensión.

Ese comportamiento refleja que la entidad financiera ha avanzado en la incorporación de prácticas de automatización, aunque de manera parcial. La integración continua de cambios opera como un punto de entrada ya asentado, mientras que capacidades que la complementan, como la cobertura de pruebas críticas y la ejecución automática de pipeline, todavía no se ejecutan con la regularidad necesaria. Fortalecer precisamente esas prácticas es lo que permitiría detectar problemas de forma sistemática antes de que lleguen a ambientes productivos, algo que en el sector financiero cobra especial relevancia, dado que los errores en producción pueden derivar en consecuencias tanto operativas como regulatorias para la institución.

Infraestructura como Código (IaC)

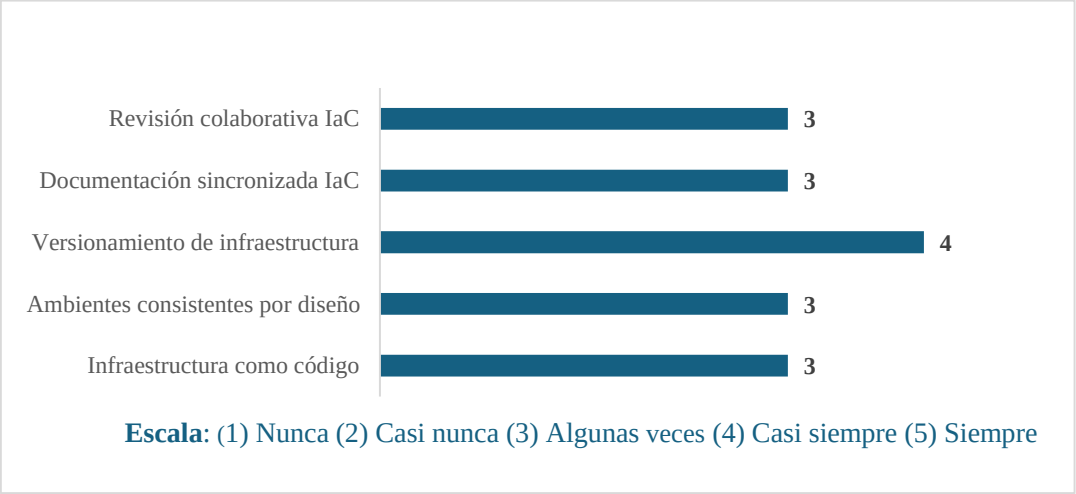

Figura 4. Mediana Dimensión IaC
Fuente: Elaboración propia (2026).

La dimensión de infraestructura como código arrojó un comportamiento parecido al que se observó en CI/CD. La revisión colaborativa de IaC, la documentación sincronizada, la creación de ambientes consistentes por diseño y el uso de infraestructura como código en sí mismo se ubicaron todos en mediana de 3, es decir, en "Algunas veces". Solo el versionado de infraestructura rompió ese patrón, con mediana de 4 que lo coloca en "Casi siempre".

Ese resultado no extraño, ya que el control de versiones es una práctica que los equipos técnicos incorporan temprano en su trayectoria profesional, mucho antes de adentrarse en DevOps. Lo que sí merece atención es que la revisión colaborativa y la documentación sincronizada, dos prácticas que exigen trabajo coordinado entre distintos equipos, sigan siendo ocasionales. Esa situación limita el alcance real de IaC, porque versionar los scripts de configuración es solo una parte del proceso: el valor completo de esta práctica aparece cuando la definición, la revisión y la documentación de la infraestructura operan juntas de forma habitual, condición que la organización todavía está en proceso de alcanzar.

Gestión de Contenedores

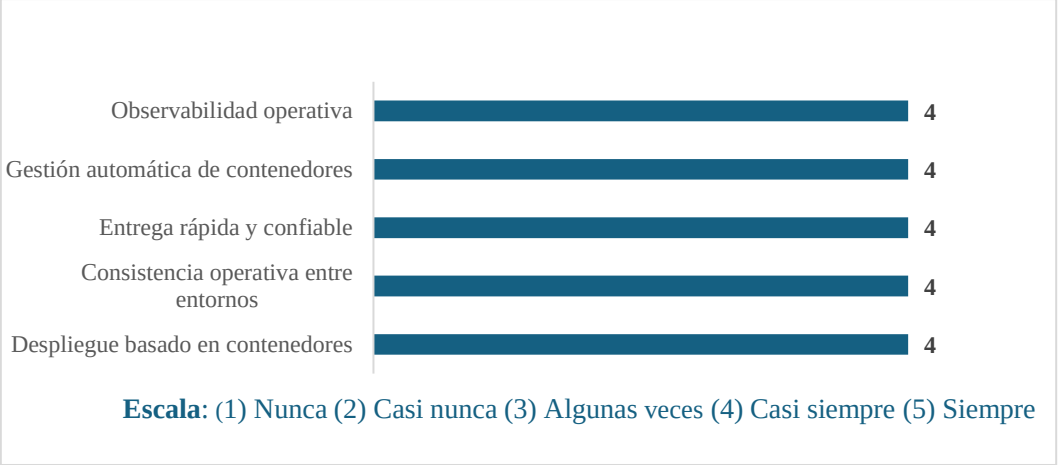

Figura 5. Mediana Dimensión Contenedores
Fuente: Elaboración propia (2026).

La gestión de contenedores resultó ser la dimensión DevOps con el perfil de madurez más uniforme y elevado dentro de la organización. La observabilidad operativa, la gestión automática de contenedores, la entrega rápida y confiable, la consistencia operativa entre entornos y el despliegue basado en contenedores obtuvieron todos mediana de 4, equivalente a "Casi siempre".

Que todos los indicadores de esa dimensión compartan el mismo valor habla de una práctica que la entidad financiera ha logrado extender de manera transversal, con presencia regular tanto en la construcción como en la operación del software. Ese panorama contrasta con lo visto en CI/CD e IaC, donde varios indicadores todavía se mueven en frecuencias intermedias. El nivel alcanzado en contenedores no es un detalle menor: esta dimensión funciona como base tecnológica que facilita la estandarización de ambientes y la entrega consistente de software, dos condiciones que la organización puede aprovechar para impulsar la madurez en las otras dimensiones donde aún quedan brechas por cerrar.

4.2.2 ANÁLISIS DETALLADO DE ATRIBUTOS DE CALIDAD

Mantenibilidad del Software

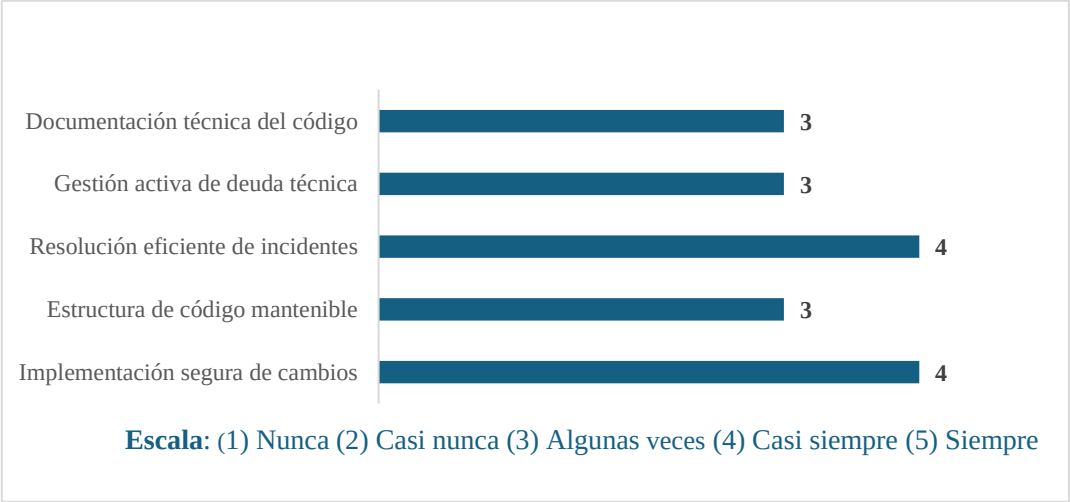

Figura 6. Mediana Dimensión Calidad
Fuente: Elaboración propia (2026).

Al revisar los indicadores de mantenibilidad mediante la mediana, lo primero que salta a la vista es una división entre lo preventivo y lo reactivo. La documentación técnica del código, la gestión de deuda técnica y la estructura de código mantenible se ubicaron todas en mediana de 3, es decir, "Algunas veces", señal de que las prácticas que cuidan la salud estructural del software no tienen todavía un lugar fijo en la rutina de los equipos. Distinto fue el caso de la resolución eficiente de incidentes y la implementación segura de cambios, que llegaron a mediana de 4 o "Casi siempre", mostrando más solidez en lo que ocurre después de que algo falla.

Ese desequilibrio cobra peso particular en una entidad financiera. Las regulaciones cambian, el negocio evoluciona y el software tiene que seguirles el paso sin detenerse. Cuando la documentación y la gestión de deuda técnica son ocasionales, el código va acumulando capas de complejidad que nadie registra ni controla, y cada nuevo cambio se vuelve un poco más riesgoso que el anterior. La institución ha desarrollado reflejos para responder cuando los problemas aparecen, pero construir las condiciones para que aparezcan menos sigue siendo una tarea pendiente.

Eficiencia del Sistema

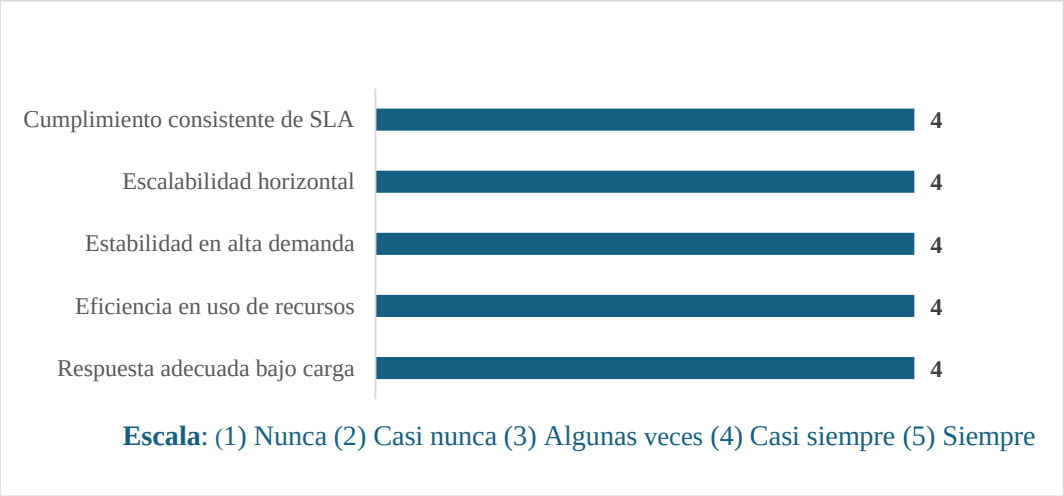

Figura 7. Mediana Dimensión Rendimiento
Fuente: Elaboración propia (2026).

La eficiencia del desempeño fue la dimensión que mostró el comportamiento más uniforme y favorable de todos los atributos de calidad analizados. El cumplimiento consistente de SLA, la escalabilidad horizontal, la estabilidad en alta demanda, la eficiencia en uso de recursos y la respuesta adecuada bajo carga obtuvieron todos mediana de 4, equivalente a "Casi siempre".

Que todos los indicadores de esa dimensión compartan el mismo valor no es un resultado menor. Los sistemas de la entidad funcionan bien desde la perspectiva del rendimiento, tanto en condiciones normales como cuando la carga aumenta, algo que en el sector bancario tiene un peso especial dado el volumen de transacciones que se procesan de manera simultánea y las exigencias de disponibilidad que el negocio impone. La estabilidad observada en aspectos como la escalabilidad horizontal y la respuesta bajo carga sugiere que las decisiones arquitectónicas tomadas por la organización han dado frutos concretos en este atributo, y que el nivel de madurez alcanzado en la gestión de contenedores podría estar ayudando a sostener ese desempeño de manera sostenida.

Seguridad del Software

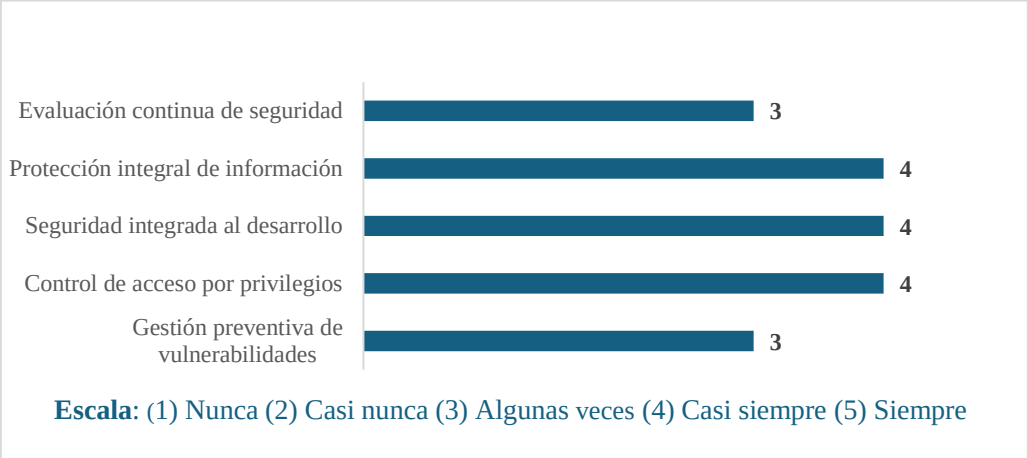

Figura 8. Mediana Dimensión Seguridad
Fuente: Elaboración propia (2026).

Los indicadores de seguridad no se comportaron de manera uniforme. La protección integral de información, la seguridad integrada al desarrollo y el control de acceso por privilegios llegaron todos a mediana de 4 o "Casi siempre", lo que muestra que los controles para cuidar los datos y regular quién accede a qué se aplican con regularidad. Distinto fue el caso de la evaluación continua de seguridad y la gestión preventiva de vulnerabilidades, que se quedaron en mediana de 3, apareciendo solo "Algunas veces" en la práctica diaria.

Lo que ese patrón revela es que la institución tiene consolidado lo que las regulaciones le exigen de manera explícita, como proteger datos y controlar accesos, pero las prácticas que van más allá del cumplimiento mínimo todavía no tienen el mismo arraigo. En un entorno donde el software cambia con frecuencia, esa diferencia importa: cada despliegue es una oportunidad para que entren nuevos riesgos, y detectarlos depende de que la evaluación de vulnerabilidades ocurra de forma continua y no solo cuando algo ya salió mal. Esa es precisamente la lógica detrás del enfoque DevSecOps, que plantea meter la seguridad dentro del proceso de desarrollo desde el principio, no agregarla como un paso al final cuando el costo de corregir ya es mayor.

4.2.3 ANÁLISIS EXPLORATORIO DE RELACIONES DEVOPS-CALIDAD

Tabla 10. Análisis Exploratorio de Relación Devops / Calidad

Relación DevOps-Calidad: Distribución por Niveles

		CALIDAD_Cat					
		Bajo (1.00-2.33)		Medio (2.34-3.66)		Alto (3.67-5.00)	
		Recuento	% del N de fila	Recuento	% del N de fila	Recuento	% del N de fila
DEVOPS_Cat	Bajo (1.00-2.33)	2	28,6%	5	71,4%	0	0,0%
	Medio (2.34-3.66)	0	0,0%	27	87,1%	4	12,9%
	Alto (3.67-5.00)	0	0,0%	2	8,7%	21	91,3%

Fuente: Elaboración propia (análisis en SPSS, 2026)

Como cierre de la etapa descriptiva, se cruzaron los índices generales de ambas variables buscando detectar si la distribución de los datos ya anticipaba algún patrón antes de llegar a las pruebas estadísticas formales. Los resultados de esa tabulación resultaron bastante reveladores. El 91.3% de los casos que registraron un nivel alto de implementación DevOps se encontraban al mismo tiempo en un nivel alto de calidad del software. Del otro lado, ningún caso con DevOps bajo llegó a calidad alta, dato que habla por sí solo sobre una relación que no parece casual. En los valores intermedios ocurrió algo similar: el 87.1% de quienes tenían DevOps medio cayeron también en calidad media, lo que muestra que esa correspondencia entre variables no es exclusiva de los extremos, sino que se repite en toda la escala.

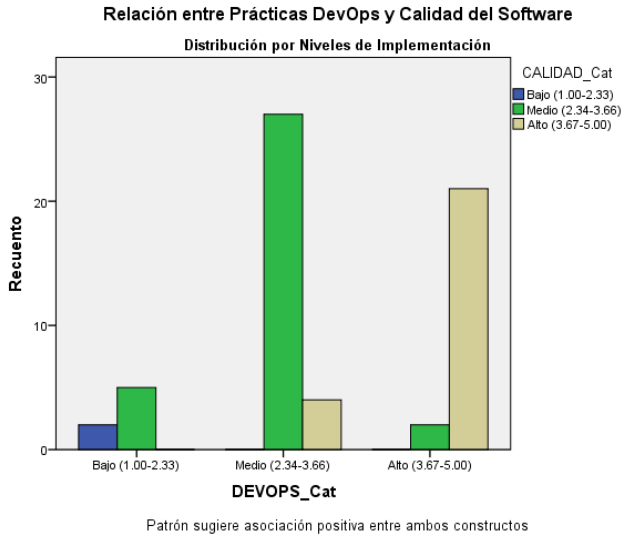

Figura 9. Relación entre niveles de implementación de prácticas DevOps y niveles de calidad del software

Fuente: Elaboración propia (análisis en SPSS, 2026)

La figura permite ver en imágenes lo que los números ya describían. A medida que sube el nivel de implementación DevOps, la proporción de casos con calidad alta crece de manera sostenida, hasta casi desaparecer en los niveles más bajos de adopción. Esa progresión no hace más que confirmar lo que la tabulación cruzada ya había anticipado. El hecho de que no aparezca ningún caso donde DevOps alto conviva con calidad baja, ni tampoco DevOps bajo con calidad alta, refuerza la idea de que la relación entre ambas variables sigue una dirección definida y no responde al azar. Ese hallazgo exploratorio deja abierto el camino hacia el análisis inferencial, donde esa misma relación se somete a herramientas estadísticas de mayor precisión.

4.2.4 SÍNTESIS DE HALLAZGOS DESCRIPTIVOS

Lo que los resultados descriptivos dejaron como saldo fue un panorama que habla bien de la viabilidad del estudio y al mismo tiempo revela patrones que merecen atención. Las distribuciones de todas las dimensiones se acercaron a la normalidad, lo que abre la puerta a los análisis inferenciales que vienen más adelante, y la variabilidad registrada resultó suficiente para que las correlaciones puedan detectarse con solidez estadística.

Los hallazgos identifican las áreas de excelencia (Contenedores, Eficiencia, Seguridad), oportunidades de consolidación (CI/CD) y prioridades de mejora (IaC, Mantenibilidad), proporcionando diagnóstico específico valioso para la gestión estratégica de tecnología en la entidad financiera. La evidencia exploratoria de relación DevOps-Calidad establece fundamento empírico que será sometido a análisis estadístico formal en la siguiente sección.

4.3 ANÁLISIS ESTADÍSTICO

El análisis estadístico inferencial tuvo como propósito evaluar formalmente la relación entre el nivel de implementación de prácticas DevOps y los atributos de calidad del software. Como las variables se midieron con una escala Likert de cinco puntos, cuya naturaleza es ordinal, el coeficiente de correlación Rho de Spearman fue la herramienta estadística que mejor se ajustaba a ese tipo de datos. Su ventaja principal es que permite establecer la dirección y magnitud de una asociación entre variables ordinales sin necesidad de asumir que los datos siguen una distribución normal, condición que no aplica cuando se trabaja con escalas de ese tipo. Los análisis tomaron como base los 64 casos que completaron el instrumento en su totalidad, con un nivel de significancia fijado en $\alpha = 0.05$ para todas las pruebas.

4.3.1 ESTADÍSTICAS INFERENCIALES Y PRUEBAS DE HIPÓTESIS

Evaluación de la hipótesis central

Para contrastar la hipótesis principal de investigación se calculó el coeficiente Rho de Spearman entre el índice general de prácticas DevOps y el índice general de calidad del software. Los resultados se presentan en la tabla siguiente.

Tabla 11. Prueba de Hipótesis Central

Correlaciones

			Nivel General de Prácticas DevOps	Calidad General del Software
Rho de Spearman	Nivel General de Prácticas DevOps	Coeficiente de correlación	1,000	,774**
		Sig. (bilateral)	.	,000
		N	64	64
	Calidad General del Software	Coeficiente de correlación	,774**	1,000
		Sig. (bilateral)	,000	.
		N	64	64

**. La correlación es significativa en el nivel 0,01 (2 colas).

Fuente: Elaboración propia (análisis en SPSS, 2026)

El coeficiente entre ambas variables resultó en Rho = 0.774, $p < 0.01$ y N = 64, valor que corresponde a una correlación positiva fuerte y estadísticamente significativa. Ese resultado fue suficiente para rechazar la hipótesis nula que planteaba la ausencia de relación entre el nivel de implementación de prácticas DevOps y la calidad del producto de software y así darle sustento empírico a la hipótesis de investigación. Lo que el coeficiente refleja en la práctica es que, dentro de la entidad financiera, los entornos con mayor madurez DevOps van acompañados de una percepción más favorable sobre la calidad del software en sus tres dimensiones evaluadas: mantenibilidad, eficiencia y seguridad.

La relación encontrada no responde al azar. Su magnitud cae dentro del rango que convencionalmente se interpreta como fuerte, lo que le da solidez a esa conclusión. El contexto donde ese hallazgo apareció le suma relevancia adicional: las instituciones financieras operan bajo regulaciones estrictas y con sistemas que sostienen procesos críticos, condiciones que convierten la calidad del software en una exigencia de funcionamiento y no simplemente en una meta aspiracional. Que el nivel de madurez DevOps muestre una asociación fuerte con ese atributo en ese entorno específico es, por tanto, un resultado con implicaciones concretas para la gestión tecnológica de la organización.

Dispersión DevOps – Calidad

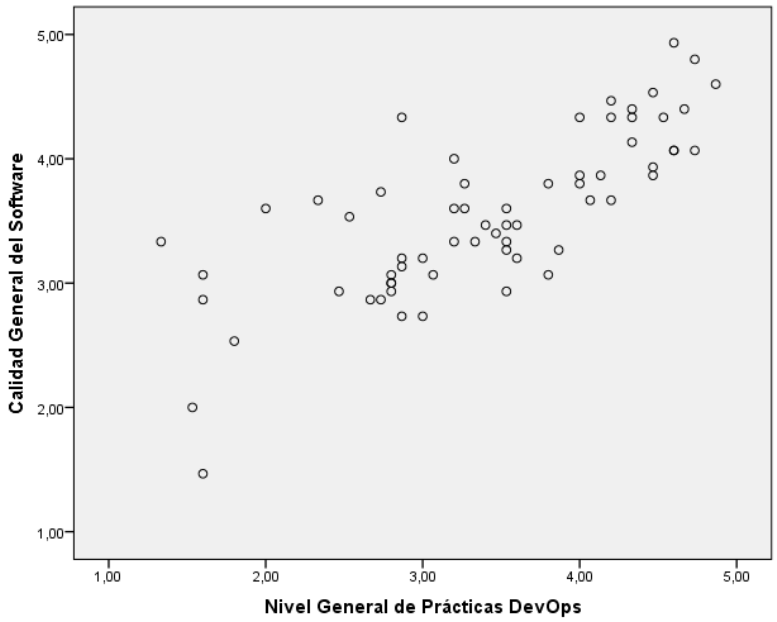

Figura 12. Gráfico de dispersión DevOps–Calidad
Fuente: Elaboración propia (análisis en SPSS, 2026)

El gráfico de dispersión le pone cara a lo que los números describieron. Cuando se observa cómo se distribuyen los puntos, la dirección que siguen es clara: los casos con niveles más altos de implementación DevOps tienden a ubicarse hacia los valores superiores del eje de calidad, mientras que los de menor adopción quedan hacia el extremo opuesto. Esa progresión refuerza lo que el coeficiente Rho de Spearman ya había dejado establecido: la relación entre ambas variables es positiva, sigue un orden consistente y no se rompe en ninguna parte del rango observado.

Análisis de correlaciones específicas entre dimensiones

Con el propósito de profundizar en la comprensión de las relaciones entre las variables, se calculó una matriz de correlaciones Rho de Spearman que incluye las tres dimensiones DevOps evaluadas, integración continua y despliegue continuo, infraestructura como código y gestión de contenedores, junto con los tres atributos de calidad del software analizados, mantenibilidad, eficiencia del sistema y seguridad del software. Los resultados se presentan en la tabla siguiente.

Tabla 12. Prueba de Hipótesis Por Dimensiones

Correlaciones

			Integración Continua y Despliegue Continuo	Infraestructura como Código	Gestión de Contenedores	Mantenibilidad del Software	Eficiencia del Sistema	Seguridad del Software
Rho de Spearman	Integración Continua y Despliegue Continuo	Coeficiente de correlación	1,000	,751**	,715**	,707**	,702**	,639**
		Sig. (bilateral)	.	,000	,000	,000	,000	,000
		N	64	64	64	64	64	64
	Infraestructura como Código	Coeficiente de correlación	,751**	1,000	,639**	,620**	,683**	,552**
		Sig. (bilateral)	,000	.	,000	,000	,000	,000
		N	64	64	64	64	64	64
	Gestión de Contenedores	Coeficiente de correlación	,715**	,639**	1,000	,639**	,633**	,450**
		Sig. (bilateral)	,000	,000	.	,000	,000	,000
		N	64	64	64	64	64	64
	Mantenibilidad del Software	Coeficiente de correlación	,707**	,620**	,639**	1,000	,770**	,643**
		Sig. (bilateral)	,000	,000	,000	.	,000	,000
		N	64	64	64	64	64	64
	Eficiencia del Sistema	Coeficiente de correlación	,702**	,683**	,633**	,770**	1,000	,655**
		Sig. (bilateral)	,000	,000	,000	,000	.	,000
		N	64	64	64	64	64	64
	Seguridad del Software	Coeficiente de correlación	,639**	,552**	,450**	,643**	,655**	1,000
		Sig. (bilateral)	,000	,000	,000	,000	,000	.
		N	64	64	64	64	64	64

**. La correlación es significativa en el nivel 0,01 (2 colas).

Fuente: Elaboración propia (análisis en SPSS, 2026)

La matriz de correlaciones no dejó lugar a dudas: todas las combinaciones entre dimensiones DevOps y atributos de calidad arrojaron valores positivos y estadísticamente significativos al nivel 0.01. Ese resultado confirma que la relación identificada en la hipótesis central no es un fenómeno que ocurre solo cuando se miran las variables de manera agregada, sino que se replica con consistencia en cada cruce específico de dimensiones.

De las tres prácticas DevOps analizadas, la integración continua y despliegue continuo fue la que mostró las asociaciones más altas con los tres atributos de calidad: Rho = 0.707 con mantenibilidad, Rho = 0.702 con eficiencia del sistema y Rho = 0.639 con seguridad del software. Esos valores posicionan a CI/CD como la práctica con mayor alcance transversal sobre la calidad, algo que tiene sentido considerando que su funcionamiento obliga a verificar cada cambio antes de que llegue a producción, favoreciendo tanto la detección temprana de defectos como la estandarización de los procesos de construcción y verificación del software.

La infraestructura como código también registró correlaciones significativas en los tres atributos: Rho = 0.620 con mantenibilidad, Rho = 0.683 con eficiencia del sistema y Rho = 0.552 con seguridad del software. Que su aporte sea más notorio en eficiencia no sorprende, dado que IaC reduce la variabilidad en los despliegues al favorecer la

reproducibilidad de ambientes, condición que incide directamente sobre la estabilidad y el rendimiento operativo de los sistemas.

La gestión de contenedores registró Rho = 0.639 con mantenibilidad, Rho = 0.633 con eficiencia del sistema y Rho = 0.450 con seguridad del software. Ese último dato es el más bajo de toda la matriz, y su explicación está en que tener contenedores bien implementados agiliza las entregas y da consistencia a los ambientes, pero ninguna de esas ventajas cubre automáticamente los frentes de seguridad. Para que esa dimensión impacte también en seguridad, hacen falta controles adicionales dentro del pipeline, como el escaneo de imágenes o la gestión de secretos, que son los que el enfoque DevSecOps propone integrar desde el inicio del proceso.

Dentro de los atributos de calidad, la relación más fuerte de toda la matriz no fue entre una práctica DevOps y un atributo, sino entre dos atributos: mantenibilidad y eficiencia del sistema alcanzaron Rho = 0.770. Esa cifra refleja algo que desde lo técnico tiene bastante lógica: cuando el código está bien organizado y es fácil de tocar, el sistema tiende a funcionar mejor, porque la deuda técnica que se va acumulando con el tiempo es la que termina pesando sobre el rendimiento. La seguridad, por su lado, mostró Rho = 0.643 con mantenibilidad y Rho = 0.655 con eficiencia, valores que indican que los tres atributos se mueven en la misma dirección dentro de la entidad y se sostienen mutuamente.

Interpretación integrada de hallazgos inferenciales

Los resultados del análisis inferencial ofrecen una imagen consistente sobre la relación entre DevOps y calidad del software en la entidad estudiada. La correlación general de Rho = 0.774 establece que esa asociación es positiva y fuerte, y la matriz de correlaciones específicas ayuda a identificar qué dimensiones y atributos son los que más pesan en esa relación.

CI/CD sobresale como la práctica con mayor alcance sobre la calidad, porque sus coeficientes con los tres atributos evaluados son los más altos y los más parejos de toda la matriz. Eso indica que invertir en fortalecer esa dimensión dentro de la organización no beneficiaría un solo atributo, sino que su efecto se distribuiría de manera amplia. IaC, en cambio, muestra un aporte que apunta principalmente hacia la eficiencia del sistema, mientras que contenedores, pese a ser la práctica más consolidada, registró el coeficiente

más bajo frente a seguridad, señal de que dominar esa herramienta no es suficiente si no va acompañada de controles que cubran ese frente específico.

Mirando los atributos entre sí, la correlación entre mantenibilidad y eficiencia del sistema deja ver que trabajar en la estructura del código y reducir la deuda técnica genera efectos que se extienden también sobre el rendimiento operativo. Ese tipo de sinergia, sumada al resto de los hallazgos inferenciales, le da sustento estadístico sólido a la hipótesis central del estudio y entrega a la organización una base empírica concreta para priorizar dónde poner sus esfuerzos de mejora.

4.4 ANÁLISIS COMPLEMENTARIO DE DATOS

El análisis estadístico confirmó una asociación positiva entre las prácticas DevOps y la calidad del software; sin embargo, la matriz de correlaciones por dimensiones reveló un comportamiento particular que merece atención. La gestión de contenedores, siendo la práctica más consolidada de la organización con todos sus indicadores en mediana de 4, registró frente a la seguridad del software el coeficiente más bajo de toda la matriz (Rho = 0.450), por debajo de CI/CD (Rho = 0.639) e IaC (Rho = 0.552) para ese mismo atributo. Esto se explica al observar que la evaluación continua de seguridad y la gestión preventiva de vulnerabilidades fueron los indicadores con las medianas más bajas del estudio, ambos en 3, precisamente los controles que deberían estar integrados dentro del pipeline de contenedores mediante escaneo de imágenes, gestión de secretos y políticas de admisión. Rahman y Williams (2016) señalaron que, en entornos de despliegue frecuente, la ausencia de controles de seguridad automatizados dentro del flujo de entrega expone al producto a vulnerabilidades difíciles de detectar mediante revisiones manuales, advertencia que resulta directamente aplicable a los hallazgos obtenidos.

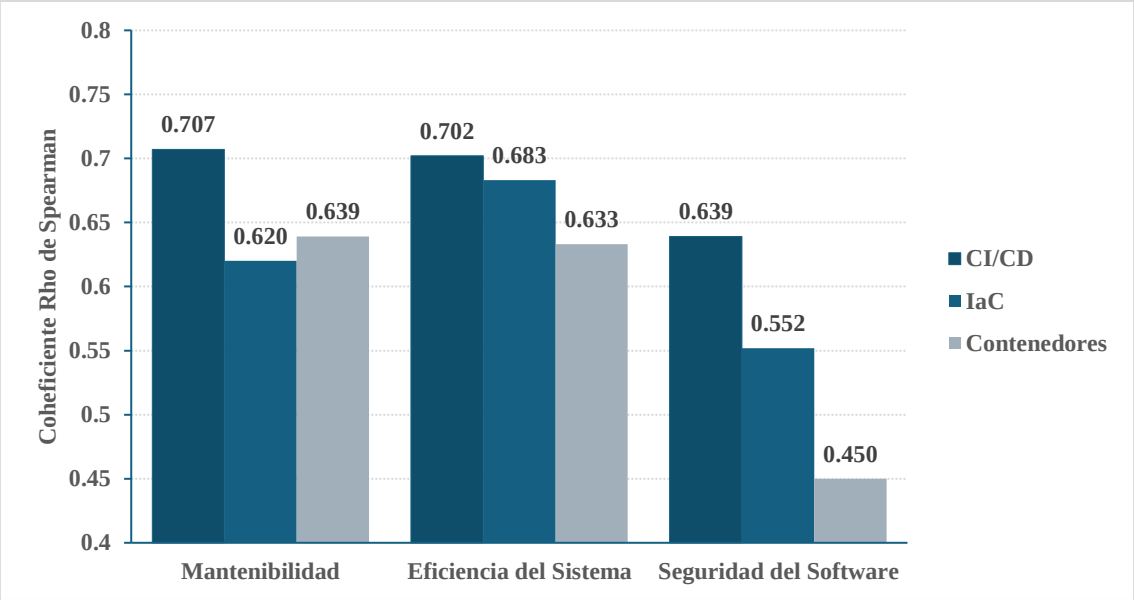

Figura 10. Gráfico de Coeficientes Rho de Spearman: prácticas DevOps y atributos de calidad
Fuente: Elaboración propia (análisis en SPSS, 2026)

Un segundo hallazgo relevante surge de las relaciones entre los propios atributos de calidad. La correlación entre mantenibilidad y eficiencia del sistema alcanzó el valor más alto de toda la matriz (Rho = 0.770), superando incluso la correlación general entre DevOps y calidad. Amaro, Pereira y da Silva (2023) documentaron que la mejora en la estructura interna del código y la reducción de la deuda técnica tienden a producir efectos simultáneos sobre el rendimiento operativo, dado que un software bien organizado consume menos recursos y responde con mayor estabilidad bajo carga. Esto indica que fortalecer la mantenibilidad mediante documentación técnica sistemática y refactorización planificada no opera de forma aislada, sino que genera un impacto adicional sobre la eficiencia del sistema.

Ambos patrones identificados apuntan en la misma dirección: el avance en la calidad del software no depende únicamente de adoptar más prácticas DevOps, sino de ampliar el alcance de las que ya están consolidadas. La infraestructura de contenedores existente representa una base técnica sobre la cual es viable incorporar los controles de seguridad faltantes sin rediseñar lo existente, convirtiendo esa intervención en una prioridad de alto retorno. De igual forma, atender los indicadores más débiles de mantenibilidad produciría, según la evidencia empírica de este estudio, mejoras que se extenderían también sobre la eficiencia operativa del sistema.

Desde una perspectiva metodológica, es pertinente precisar que la totalidad de los datos presentados en este capítulo corresponden a percepciones profesionales

estructuradas mediante escala Likert de cinco puntos, recopiladas de 64 colaboradores del área de Tecnologías de Información de la sociedad financiera. Este enfoque perceptual constituye una aproximación válida y ampliamente utilizada en investigación organizacional sobre DevOps (Forsgren et al., 2018; Mohammad, 2017; Offermann et al., 2022). No obstante, la triangulación de estos resultados con métricas objetivas extraídas directamente de las herramientas DevOps, como los registros de pipeline de Jenkins o Azure DevOps, los reportes de cobertura de SonarQube para mantenibilidad, los reportes SAST/DAST para seguridad, y los datos de plataformas de monitoreo APM para eficiencia del desempeño, representaría un paso metodológico adicional que elevaría el nivel de validez externa del estudio y reduciría el riesgo de sesgo de deseabilidad social inherente al autoinforme (Podsakoff et al., 2003). En ese sentido, los coeficientes Rho de Spearman obtenidos deben interpretarse como correlaciones entre percepciones informadas de profesionales técnicos con experiencia directa en los procesos evaluados, y no como correlaciones entre métricas objetivas del sistema, distinción que resulta relevante para situar con precisión académica el alcance y la generalización de los hallazgos.

4.5 DISCUSIÓN DE RESULTADOS

Los resultados obtenidos permiten establecer un diálogo con lo que investigadores previos han documentado sobre DevOps y calidad del software, abriendo además reflexiones particulares derivadas del contexto financiero en que se desarrolló el estudio.

4.5.1 COMPARACIÓN CON LA LITERATURA CIENTÍFICA

El coeficiente Rho de Spearman obtenido ($Rho = 0.774$, $p < 0.01$) guarda coherencia con los planteamientos de Forsgren et al. (2018), quienes documentaron que las organizaciones con mayor madurez DevOps tienden a alcanzar mejores resultados operativos. No obstante, existe una distinción metodológica relevante: mientras ese referente se concentró en métricas de proceso como frecuencia de despliegue y tiempo de recuperación, el presente estudio se orientó hacia atributos del producto definidos por ISO/IEC 25010, lo que amplía la perspectiva de análisis hacia dimensiones que históricamente han recibido menor atención empírica.

Esta orientación responde directamente a la brecha identificada por Mishra y Otaiwi (2020), quienes advirtieron que la mayor parte de los estudios sobre DevOps privilegian la automatización y la entrega continua, dejando en segundo plano atributos como mantenibilidad, eficiencia y seguridad del producto. Los hallazgos de esta

investigación contribuyen a cerrar ese vacío al aportar evidencia sobre esas dimensiones específicas en un sector poco representado en la literatura.

En esa misma línea, López et al. (2022) señalaron que la desconexión entre velocidad de entrega y evaluación de calidad del producto es uno de los desafíos más persistentes en la industria. Las correlaciones identificadas entre CI/CD y mantenibilidad (Rho = 0.707) y eficiencia (Rho = 0.702) sugieren que cuando la automatización alcanza niveles de madurez suficientes, ambos objetivos no son necesariamente contradictorios sino complementarios.

Desde el plano técnico, Şenkal et al. (2023) propusieron que los pipelines DevOps bien configurados pueden funcionar como mecanismos de aseguramiento continuo de calidad cuando se acompañan de métricas definidas. Los resultados obtenidos para CI/CD, que registró las correlaciones más altas con los tres atributos de calidad evaluados, ofrecen sustento empírico a esa propuesta. Finalmente, García-Mireles et al. (2024) señalaron la escasa representación de contextos latinoamericanos y sectores regulados en la investigación sobre DevOps, condición que este estudio aborda al desarrollarse en una entidad financiera hondureña.

4.5.2 ANÁLISIS REFLEXIVO Y CRÍTICO

Los resultados invitan a una lectura más específica sobre qué aspectos funcionan adecuadamente y cuáles demandan intervención dentro de la organización estudiada. Un primer punto de reflexión surge de la asimetría entre el nivel de implementación de contenedores y su correlación con seguridad (Rho = 0.450), la más baja de toda la matriz. Esto indica que una adopción tecnológica avanzada no se traduce automáticamente en mejoras equivalentes en todos los atributos de calidad. Prates et al. (2019) ya advertían que la aceleración en los ciclos de despliegue puede ampliar la superficie de exposición a riesgos si no se integran controles de seguridad específicos a lo largo del proceso de desarrollo, observación que cobra plena vigencia en este caso.

Un segundo elemento que merece atención es la situación de la mantenibilidad, donde los indicadores de documentación técnica y gestión de deuda técnica presentaron las medianas más bajas del estudio. En una entidad financiera sujeta a marcos regulatorios que exigen trazabilidad y capacidad de auditoría, un software difícil de modificar trasciende lo técnico y se convierte en un problema de gobernanza. Orozco-Garcés et al. (2022) señalaron precisamente que la transformación DevOps en sectores regulados

requiere mecanismos explícitos de medición de calidad del producto para que los cambios adoptados generen impacto real y sostenible en el tiempo.

Finalmente, el comportamiento de CI/CD como práctica articuladora merece destacarse. Sus correlaciones más altas y consistentes con los tres atributos de calidad sugieren que cuando esta dimensión opera con madurez, actúa como palanca que impulsa la calidad de forma transversal, lo que convierte su fortalecimiento en una prioridad estratégica antes de dispersar esfuerzos en otras iniciativas.

4.5.3 RELACIÓN CON OBJETIVOS E HIPÓTESIS

El primer objetivo específico buscaba cuantificar el nivel de implementación de prácticas DevOps. Los resultados mostraron que la gestión de contenedores representa la práctica más consolidada, seguida por CI/CD, mientras que IaC presenta las mayores oportunidades de mejora, con varios indicadores ubicados en la categoría de "Algunas veces".

El segundo objetivo planteaba analizar la correlación entre CI/CD y mantenibilidad. La asociación positiva fuerte obtenida ($Rho = 0.707$, $p < 0.01$) indica que a medida que maduran los procesos de integración y despliegue continuo, se percibe una mejora en la capacidad del software para ser modificado de forma efectiva, resultado coherente con la naturaleza de esta práctica como mecanismo de verificación sistemática del código.

El tercer objetivo apuntaba a evaluar la relación entre IaC y contenedores con la eficiencia del software. Ambas prácticas mostraron correlaciones significativas con este atributo ($Rho = 0.683$ e IaC y $Rho = 0.633$ para contenedores), confirmando que la gestión programática de la infraestructura y la estandarización de ambientes contribuyen a que los sistemas operen con mayor estabilidad bajo condiciones de uso habituales.

El cuarto objetivo buscaba examinar la relación entre el despliegue automatizado y las métricas de seguridad. CI/CD obtuvo la correlación más alta dentro de las prácticas DevOps evaluadas ($Rho = 0.639$, $p < 0.01$), seguida por IaC ($Rho = 0.552$) y contenedores ($Rho = 0.450$). La correlación más moderada de los contenedores con este atributo, pese a ser la práctica mejor implementada, señala una oportunidad concreta para integrar controles de seguridad dentro de los flujos de trabajo asociados a esta tecnología.

Los cuatro objetivos específicos fueron alcanzados y sus hallazgos convergen en una conclusión consistente con la hipótesis de investigación: el nivel de implementación de prácticas DevOps presenta una relación positiva y estadísticamente significativa con la calidad del producto de software ($Rho = 0.774$, $p < 0.01$), aportando evidencia empírica concreta para la toma de decisiones tecnológicas en contextos financieros de la región latinoamericana.

4.6 CONSIDERACIONES ÉTICAS

El desarrollo de esta investigación estuvo guiado por principios éticos que acompañaron cada etapa del proceso, desde la recolección de datos hasta la presentación de los resultados. Estos principios no se asumieron como requisitos formales sino como condiciones fundamentales para garantizar la validez y la responsabilidad del trabajo realizado.

En lo que respecta al consentimiento informado, se gestionó de forma anticipada la autorización formal de la dirección de Tecnologías de Información de la entidad financiera participante. Antes de la aplicación del instrumento, cada colaborador recibió información clara sobre los objetivos del estudio, el uso que se daría a los datos recopilados y el carácter estrictamente académico de la investigación, asegurando que la participación fuera completamente voluntaria en todos los casos.

Respecto a la confidencialidad, la información obtenida fue tratada de manera anónima desde su recolección. El instrumento fue diseñado sin campos que permitieran identificar a los participantes de forma individual, y los resultados se presentan de manera agregada a lo largo de todo el documento. La identidad de la institución financiera también fue resguardada, evitando incluir datos que pudieran comprometer su posición competitiva o reputacional. El acceso a la información recopilada estuvo restringido únicamente al equipo investigador.

En materia de honestidad y objetividad, los datos se reportaron en su totalidad sin omisiones selectivas ni ajustes que favorecieran determinadas conclusiones. Las limitaciones del estudio fueron reconocidas de manera explícita a lo largo del documento, y las conclusiones se mantuvieron dentro de los límites que los propios datos permiten sostener, sin extrapolar hallazgos más allá de lo que el diseño metodológico autoriza.

La integridad académica se sostuvo mediante el cumplimiento estricto de las normas de citación y referenciación según la séptima edición del manual APA,

garantizando el reconocimiento adecuado de todos los autores consultados y evitando cualquier forma de apropiación indebida de ideas o planteamientos ajenos.

Finalmente, en cuanto al principio de beneficencia y no maleficencia, el estudio fue concebido con el propósito de generar conocimiento útil para la organización y para el campo disciplinar, sin interferir en las operaciones críticas de la entidad ni generar perjuicio alguno a las personas que participaron. Los hallazgos obtenidos fueron puestos a disposición de la institución con la intención de que puedan orientar decisiones estratégicas en materia de adopción de prácticas DevOps y mejora de la calidad del software.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

Este capítulo presenta, de manera ordenada, lo que la investigación dejó como aprendizaje. Por un lado, las conclusiones recogen lo que los datos permitieron afirmar sobre el vínculo entre DevOps y la calidad del software dentro de una entidad financiera hondureña, dando respuesta puntual a cada uno de los objetivos trazados al inicio del estudio. Por otro lado, las recomendaciones convierten esos aprendizajes en orientaciones prácticas, tanto para la institución que participó en la investigación como para otras organizaciones del sector y para quienes en el futuro decidan profundizar en estas temáticas dentro del contexto latinoamericano.

5.1 CONCLUSIONES

Los hallazgos obtenidos a lo largo de esta investigación permiten formular conclusiones concretas sobre la relación entre el nivel de implementación de prácticas DevOps y la calidad del producto de software en el contexto financiero hondureño.

5.1.1 CONFIRMACIÓN DE LA HIPÓTESIS CENTRAL

Cuando se analizan los datos recogidos de 64 profesionales de tecnología, lo que queda claro es que adoptar DevOps de forma estructurada no es indiferente a la calidad del software que produce una organización financiera. La asociación encontrada entre ambas variables resultó positiva y significativa, lo que descartó la hipótesis nula y validó la hipótesis de investigación. Esta correlación fuerte refleja que, dentro de la sociedad financiera estudiada, un mayor nivel de implementación de prácticas DevOps, el cual fue medido a través de las dimensiones de automatización CI/CD, infraestructura como código y contenerización, se asocia con niveles favorables en los tres atributos de calidad del producto evaluados: mantenibilidad, seguridad y eficiencia de desempeño, según los define la norma ISO/IEC 25010. Más allá del dato estadístico, lo que esto representa para el sector financiero hondureño es contar con evidencia empírica propia sobre algo que muchos profesionales intuyen, pero pocos han podido medir con rigor dentro de la región.

5.1.2 NIVEL DE IMPLEMENTACIÓN DEVOPS ASIMÉTRICO

Dentro de la institución analizada, no todas las prácticas DevOps han avanzado al mismo ritmo. Los contenedores muestran un uso bastante extendido, pero la infraestructura como código sigue siendo la dimensión con mayor retraso, con la mayoría de sus indicadores aplicándose solo de forma ocasional. Lo que los resultados permiten concluir es que mientras esa brecha persista, la organización no podrá aprovechar

plenamente la relación que existe entre DevOps y calidad, porque precisamente IaC es la práctica que más se vincula con la estandarización de ambientes y la trazabilidad, dos aspectos que en el sector financiero no son opcionales sino obligaciones regulatorias concretas.

5.1.3 CI/CD COMO PRINCIPAL FACTOR DE MANTENIBILIDAD

Entre todas las combinaciones analizadas, la integración continua y despliegue continuo fue la práctica que mostró la asociación más fuerte con la capacidad del software para ser modificado de forma efectiva. Una práctica que obliga a verificar cada cambio antes de incorporarlo al producto va generando, casi sin que los equipos lo noten, condiciones que mejoran la estructura del código, adelantan la detección de problemas y frenan la acumulación de complejidad. Se concluye que fortalecer CI/CD, particularmente en cobertura de pruebas y ejecución automática de pipelines, es el camino más directo para mejorar la mantenibilidad del software en la entidad de forma sostenida.

5.1.4 IAC Y CONTENEDORES COMO SOPORTE DE LA EFICIENCIA OPERATIVA

IaC y la gestión de contenedores registraron correlaciones significativas con la eficiencia del desempeño del software, atributo que a lo largo del estudio presentó el comportamiento más uniforme y favorable de todos los evaluados. Ese resultado tiene una explicación técnica directa: cuando los ambientes se construyen de manera estandarizada y los despliegues son reproducibles, la variabilidad disminuye y el sistema mantiene estabilidad operativa incluso bajo condiciones de alta demanda. Los datos permiten concluir que las decisiones arquitectónicas adoptadas por la organización en torno a esas dos prácticas han tenido un impacto positivo y medible sobre ese atributo, y que consolidar su desarrollo mantiene el potencial de mejora en esa dimensión.

5.1.5 DESPLIEGUE AUTOMATIZADO Y CONTROLES DE SEGURIDAD

El análisis del impacto del despliegue automatizado sobre las métricas de seguridad mostró que CI/CD registró la correlación más alta entre las prácticas DevOps para ese atributo, seguido por IaC y contenedores. Sin embargo, el hallazgo más relevante es que la gestión de contenedores, siendo la dimensión más madura de la organización, presentó la correlación más baja con seguridad. Esto lleva a concluir que la automatización del despliegue favorece la seguridad del software, pero solo cuando va acompañada de controles integrados al pipeline como el escaneo de imágenes y la gestión

preventiva de vulnerabilidades. Sin esos controles, una alta madurez tecnológica no se traduce automáticamente en mejor seguridad del producto.

5.2 RECOMENDACIONES

A partir de las conclusiones anteriores, se presentan a continuación las recomendaciones que surgen de manera directa de los resultados del estudio.

5.2.1 INVERTIR EN LA MADURACIÓN DE IAC

Los hallazgos sobre el rezago de infraestructura como código en la entidad no solo señalan una brecha técnica, sino una exposición concreta frente a requisitos regulatorios que exigen trazabilidad y estandarización de ambientes. La recomendación que se desprende de esa conclusión es clara: la institución debería establecer un plan estructurado para elevar el nivel de madurez de IaC, aprovechando que los equipos ya dominan la gestión de contenedores. Esa base técnica existente reduce considerablemente el esfuerzo de partida, siempre que la transición contemple también los cambios organizacionales y culturales que suelen ser los que más frenan este tipo de procesos.

5.2.2 FORTALECER CI/CD COMO PALANCA DE MANTENIBILIDAD

Dado que la integración y despliegue continuo mostró la asociación más fuerte con la mantenibilidad del software, las acciones orientadas a ese atributo deberían concentrarse en los indicadores más débiles de esa dimensión: la cobertura de pruebas críticas y la ejecución automática de pipelines. Establecer estándares de documentación técnica, incorporar la gestión de deuda técnica como práctica habitual y refactorizar gradualmente los componentes más rígidos son cambios que toman tiempo, pero que resultan necesarios para que el software pueda evolucionar sin acumular riesgos que solo se vuelven visibles cuando el daño ya está hecho.

5.2.3 PROFUNDIZAR EN IAC Y CONTENEDORES PARA SOSTENER LA EFICIENCIA OPERATIVA

Los resultados evidenciaron que tanto IaC como la gestión de contenedores tienen una relación significativa con la eficiencia del desempeño del software, atributo que mostró el comportamiento más favorable de todo el estudio. La recomendación derivada de esa conclusión es que la institución mantenga y profundice las decisiones arquitectónicas que han sostenido ese nivel de rendimiento, evitando que la incorporación de nuevas prácticas o herramientas introduzcan variabilidad en los despliegues. Consolidar la reproducibilidad de ambientes mediante IaC y extender las buenas prácticas

de orquestación de contenedores son los caminos más directos para preservar ese resultado.

5.2.4 INTEGRAR CONTROLES DEVSECOPS DENTRO DEL PIPELINE

La conclusión sobre la brecha de seguridad en contenedores apunta hacia una intervención muy específica: incorporar dentro del pipeline existente los mecanismos de control que los datos mostraron como ocasionales, como el escaneo de imágenes, la gestión de secretos y la evaluación continua de vulnerabilidades. Dado que los equipos ya dominan la herramienta, significa que esa incorporación no exige rediseñar lo existente sino ampliar su alcance. En una entidad financiera donde cada despliegue representa una oportunidad de introducir vectores de riesgo, esa intervención no debería postergarse como una mejora futura sino atenderse como una necesidad operativa concreta.

5.2.5 ORIENTACIÓN PARA OTRAS INSTITUCIONES FINANCIERAS DE LA REGIÓN

Este estudio deja sobre la mesa evidencia que puede servirles a instituciones financieras de la región, tanto a las que todavía están evaluando si tiene sentido moverse hacia DevOps como a las que ya están en ese camino y quieren entender mejor qué esperar. Los datos no dejan duda de que la relación entre estas prácticas y la calidad del software existe y se puede medir, y que lo que se gana no es solo velocidad de entrega sino productos que internamente son más sólidos y sostenibles. Para las organizaciones que recién arrancan, los resultados sugieren un orden que aprovecha cómo estas prácticas se apoyan entre sí: primero contenedores, luego consolidar CI/CD y después madurar IaC, una secuencia que reduce el esfuerzo de aprendizaje y maximiza lo que cada etapa puede aportar.

5.2.6 DIRECCIONES FUTURAS PARA LA INVESTIGACIÓN

Este estudio deja abiertas varias preguntas que investigaciones posteriores deberían retomar. El diseño transversal utilizado tiene una limitación concreta: toma una fotografía del momento, sin poder seguir cómo va transformándose la relación entre DevOps y calidad a medida que los equipos ganan experiencia con estas prácticas. Estudios longitudinales que acompañen a las mismas organizaciones durante varios ciclos de desarrollo entregarían una comprensión más dinámica de ese proceso que aquí no fue posible obtener.

Quedaría pendiente también indagar si los distintos tipos de aplicaciones que coexisten en el sector financiero, se comportan de manera diferente frente a DevOps y

calidad, algo que difícilmente un estudio de alcance general logra capturar. Cruzar el enfoque perceptual de este trabajo con métricas objetivas tomadas de los propios sistemas de desarrollo enriquecería ese análisis de forma considerable. Llevar además esta línea de investigación a varias instituciones de la región abriría la posibilidad de identificar qué condiciones organizacionales y culturales empujan o frenan la adopción de DevOps en un contexto latinoamericano que todavía tiene mucho territorio por documentar con evidencia empírica sólida.

CAPÍTULO VI: APLICABILIDAD

El siguiente capítulo tiene como propósito presentar la propuesta que se implementará dentro de la entidad financiera. Las secciones de la propuesta han sido elaboradas con base en los resultados obtenidos a partir de la encuesta aplicada. Al contar con un conocimiento de la situación actual de la organización, la propuesta se encuentra diseñada para su ejecución inmediata, considerando que los indicadores de desempeño deberán cumplir con los estándares establecidos en la norma ISO/IEC 25010.

6.1 NOMBRE DE LA PROPUESTA

El nombre propuesto es "Modelo de implementación de prácticas DevOps para el fortalecimiento de la calidad del software". Este nombre resume el objetivo de la propuesta: aplicar un modelo práctico de DevOps que aumente la calidad del producto de software en la entidad financiera. La propuesta utiliza las variables y conclusiones de la tesis (DevOps y calidad del software) y las traduce en un plan específico de acción. También se pretenden abordar las debilidades detectadas como ser documentación, deuda técnica y métricas según los hallazgos en las encuestas.

6.2 JUSTIFICACIÓN DE LA PROPUESTA

La investigación evidencia que es muy importante integrar prácticas DevOps en el ciclo de desarrollo de software ya que ayuda a acelerar los despliegues y mejorar la calidad del producto, adicional ayuda a mejorar los puntos clave de investigación como ser mantenibilidad y la eficiencia del Software. Mishra y Otaiwi (2020) señalan que el enfoque DevOps está orientado a aumentar la velocidad, frecuencia y calidad de entrega. Adicionalmente, los colaboradores encuestados indicaron debilidades en la documentación y otras prácticas colaborativas que se tienen que mejorar. Autores como Kovi et al. destacan que la documentación es "el motor oculto del éxito DevOps": la falta de guías claras provoca errores de despliegue y tiempo de inactividad. La propuesta justifica que es necesario aplicar las recomendaciones del capítulo V (automatización de CI/CD, IaC, contenedores, métricas, capacitación y mejor documentación) para corregir esas deficiencias. Un estudio que está dentro del sector de investigación se ha demostrado que implementar DevOps da beneficios claros ya que puede incrementar ingresos de 15% a 20% (DevOps Institute, 2021). Esto demuestra que la inversión en DevOps generará retornos significativos en la institución financiera.

6.3 ALCANCE DE LA PROPUESTA

El alcance define los objetivos generales y específicos del plan de acción.

- **Objetivo general:** Fortalecer la calidad del software dentro de la entidad financiera mediante la implementación de prácticas DevOps e incorporación de métricas de calidad ISO/IEC 25010.
- **Objetivos específicos:**
    - Implementar pipelines que automaticen la compilación, las pruebas y los despliegues.
    - Desarrollar infraestructura como código de manera estandarizada siguiendo las mejores normas de despliegue.
    - Contenerizar aplicaciones críticas para estandarizar entornos de desarrollo y prueba.
    - Integrar métricas de calidad (mantenibilidad, eficiencia, seguridad) en los pipelines.
    - Mejorar la documentación de procesos.
    - Identificar la deuda técnica existente mediante revisiones de código en sistemas Legacy.
    - Capacitar a los 76 colaboradores de TI en DevOps (cursos de AWS, IaC, mejores prácticas de desarrollo).

Estos objetivos se centran en entregables concretos: pipelines configurados, scripts IaC, repositorios de contenedores, dashboards de métricas, documentación actualizada y personal capacitado.

6.4 DESCRIPCIÓN Y DESARROLLO DE LA PROPUESTA

Una propuesta que se adapte a las necesidades identificadas durante la investigación es esencial para que sea productiva la implementación, se describirán todas las actividades que se realizarán y se vincularán con las dimensiones que ayudará a cada uno de los puntos descritos dentro de la propuesta. Estos entregables estarán debidamente vinculados a las mejores prácticas DevOps y seguirán los lineamientos de la norma ISO/IEC 25010.

6.4.1 ¿QUÉ SE HARÁ Y CÓMO SE HARÁ?

Se plantea un modelo de implementación DevOps de seis meses, organizado en fases iterativas. Las acciones principales serán:

- **Diagnóstico inicial (1–2 semanas):** Evaluar el estado actual de procesos, herramientas y cultura. Identificar brechas en CI/CD, despliegues manuales, cobertura de documentación y nivel de automatización. Este análisis guiará el plan detallado.
- **Diseño de la solución DevOps (3–4 semanas):** Definir la arquitectura DevOps: selección de herramientas (por ejemplo: Azure Devops/GitHub/Jenkins/GitLab CI, Terraform/Ansible, Docker/Podman para creación de imágenes) y estrategia de integración y despliegue continuo. Se formalizan estándares de código e infraestructura. Se elaboran plantillas iniciales de documentación técnica.
- **Implementación técnica (5–16 semanas):**
    - CI/CD: Configuración de pipelines automáticos. Se automatizará la compilación, pruebas unitarias e integración continua en proyectos piloto. Esto eleva la velocidad de entrega y calidad (al detectar errores temprano).
    - Infraestructura como Código: Desarrollo de scripts Terraform/Ansible para aprovisionar los entornos (desarrollo, prueba, producción). La IaC permite despliegues rápidos y reproducibles, mejorando la consistencia del entorno.
    - Contenerización: Migración de aplicaciones críticas a contenedores Docker gestionados por Kubernetes. Los contenedores garantizan entornos idénticos en dev, test y prod, eliminando el error “funciona en mi máquina”.
    - Documentación y estandarización: Paralelamente, se crearán o actualizarán manuales y runbooks (por ejemplo, en Confluence o GitBook). Se documentarán procesos de despliegue y arquitecturas; según Kovi et al., esto es crucial para evitar errores en DevOps.
    - Monitoreo y métricas: Se integrarán herramientas de monitoreo (por ejemplo, Dynatrace, Prometheus, Grafana) para recolectar indicadores clave de calidad (desempeño, seguridad, mantenibilidad). Se configuran

dashboards visibles para los equipos, siguiendo buenas prácticas del estándar ISO 25010.

- **Capacitación (paralela, semanas 4–12):** Se impartirán cursos a los 76 ingenieros de desarrollo/operaciones en DevOps y calidad. Se eligieron cursos en Udemy sobre AWS, Terraform/Ansible, DevOps y mejores prácticas de desarrollo. Cada persona tomará al menos tres cursos especializados (por ejemplo, "AWS Certified DevOps Engineer", "Terraform para DevOps", "Desarrollo seguro"), seleccionados por su alta calificación. (En Udemy los cursos de certificación rondan US$20–$130, por lo que se presupuestará un costo promedio razonable.) La capacitación refuerza los conocimientos necesarios para adoptar las nuevas prácticas sin interrumpir la operación.
- **Soporte y ajustes finales (semanas 17–24):** Monitorear los primeros despliegues automatizados, recoger retroalimentación y corregir desviaciones. Se ajustarán los pipelines o scripts según lecciones aprendidas. Al finalizar, se validará el cumplimiento de objetivos de calidad establecidos.

6.4.2 DESARROLLO DE LOS ENTREGABLES

Los entregables previstos para la institución incluyen:

- **Pipelines CI/CD configurados:** Scripts de automatización (por ejemplo, en Jenkinsfile o GitLab CI) que compilan, prueban y despliegan el software sin intervención manual.
- **Scripts de Infraestructura como Código:** Repositorio de Terraform/Ansible versionado que crea los entornos TI bajo demanda. Esto proporciona "infraestructura como código" documentada, lo cual es clave en DevOps.
- **Contenedores:** Imágenes estandarizadas para las aplicaciones críticas y definiciones de despliegue en las plataformas, asegurando entornos idénticos entre pruebas y producción.
- **Documentación actualizada:** Manuales de despliegue, diagramas de arquitectura y runbooks desarrollados o mejorados. Estos documentos servirán como única fuente de información correcta en los procesos, evitando la pérdida de conocimiento importante para la entidad.

- **Dashboards de métricas de calidad:** Tableros que muestran en tiempo real los indicadores ISO/IEC 25010 (por ejemplo, rendimiento de aplicaciones, vulnerabilidades abiertas, número de cambios desplegados), integrados con la canalización DevOps. Esto permite tomar decisiones basadas en datos.
- **Plan de capacitación:** Memoria que documenta el contenido de los cursos, lista de asistentes (76 personas) y certificaciones obtenidas.

En conjunto, estos entregables permitirán a la entidad financiera aplicar inmediatamente las mejoras sugeridas y sostenerlas en el tiempo, cada elemento fue diseñado a la medida de las debilidades detectadas.

6.4.3 MATRIZ DE RIESGOS IDENTIFICADOS Y SU MITIGACIÓN MEDIANTE PRÁCTICAS DEVOPS

La implementación de prácticas DevOps en una sociedad financiera no está exenta de riesgos tecnológicos, operativos y de cumplimiento regulatorio. A su vez, las propias prácticas DevOps, cuando se implementan con el nivel adecuado de madurez, actúan como mecanismos de mitigación que reducen de manera significativa los riesgos inherentes a los procesos tradicionales de desarrollo y operación de software. Con el propósito de ofrecer una visión estructurada de esta dualidad y de complementar los entregables descritos en la sección anterior con una perspectiva de gestión de riesgos alineada con los requerimientos de la CNBS y los estándares PCI-DSS e ISO/IEC 27001, se presenta la siguiente matriz.

La matriz fue construida cruzando tres fuentes de información: las debilidades detectadas, la escasa documentación técnica, el rezago en IaC y la correlación más baja de la matriz entre contenedores y seguridad, el marco regulatorio aplicable y los aportes de la literatura especializada en gestión de riesgos tecnológicos en contextos DevOps (Prates, Faustino, Silva & Pereira, 2019; Rahman & Williams, 2016).

Tabla 13. Matriz de Riesgos Identificados

#	Riesgo identificado	Categoría	Probabilidad	Impacto	Práctica DevOps que lo reduce	Indicador de control
R01	Introducción de defectos en producción por cambios no verificados	Operativo	Alta	Alto	Automatización CI/CD, ejecución automática de pruebas en cada cambio con bloqueo ante fallos	Tasa de fallos post-despliegue < 5%
R02	Inconsistencia entre ambientes de desarrollo, pruebas y producción	Técnico	Alta	Alto	Contenerización, empaquetado de aplicaciones con Docker y orquestación con Kubernetes	% de servicios contenerizados > 80%
R03	Acumulación de deuda técnica que compromete la mantenibilidad del código	Técnico	Alta	Medio	CI/CD con análisis estático integrado, SonarQube como quality gate obligatorio en el pipeline	Índice de deuda técnica dentro del umbral definido
R04	Vulnerabilidades de seguridad no detectadas en despliegues frecuentes	Seguridad	Media	Crítico	DevSecOps, integración de SAST y DAST como etapas obligatorias del pipeline, bloqueo ante hallazgos críticos	Vulnerabilidades críticas abiertas = 0
R05	Variabilidad e inconsistencia en la configuración de infraestructura	Técnico	Media	Alto	Infraestructura como Código, definición de toda la infraestructura mediante Terraform o Ansible versionado	% de infraestructura codificada > 70%
R06	Incumplimiento de normativas CNBS por falta de trazabilidad en cambios	Regulatorio	Media	Crítico	CI/CD con registros auditables, log inmutable de cada despliegue	100% de despliegues con registro de auditoría
R07	Tiempo de recuperación prolongado ante fallos en producción	Operativo	Media	Alto	IaC y contenedores, recreación automatizada de ambientes y reinicio de contenedores por Kubernetes ante fallos detectados	MTTR (tiempo medio de recuperación) < 1 hora
R08	Pérdida de conocimiento organizacional por documentación técnica deficiente	Organizacional	Alta	Medio	Prácticas colaborativas IaC, documentación sincronizada con el código	% de procesos críticos documentados > 90%
R09	Exposición de datos sensibles de clientes por controles de acceso inadecuados	Seguridad	Baja	Crítico	Gestión de secretos en pipeline, HashiCorp Vault o equivalente, políticas RBAC en Kubernetes	0 exposiciones de secretos en logs de pipeline
R10	Degradación del desempeño bajo alta demanda transaccional	Desempeño	Media	Alto	Orquestación de contenedores, escalado horizontal automático (HPA) en Kubernetes	Disponibilidad del sistema > 99.5% según SLA

Fuente. Elaboración propia a partir de los hallazgos del estudio, la Circular CNBS No. 025/2022, Prates et al. (2019) y Rahman y Williams (2016).

Los riesgos más críticos que fueron identificados, son las vulnerabilidades de seguridad no detectadas (R04), incumplimiento regulatorio (R06) y exposición de datos sensibles (R09), coinciden con las brechas donde el análisis estadístico registró las correlaciones y medianas más bajas, lo que confirma que la matriz refleja la situación real de la organización. La automatización CI/CD se posiciona como la práctica con mayor alcance mitigador al reducir seis de los diez riesgos identificados.

6.5 MEDIDAS DE CONTROL

Para asegurar que la propuesta sea eficaz a lo largo de toda la implementación, se definirán indicadores de desempeño, conocidos como KPI, deben estar alineados con las metas de calidad que se deben cumplir una vez se finalice el plan:

- **Frecuencia de despliegues automáticos:** porcentaje de releases realizados vía pipeline CI/CD (meta mayor a 80% automatizados).
- **Cobertura de documentación:** porcentaje de procesos clave (despliegues, recuperaciones, configuraciones) documentados oficialmente (meta del 100%). Una documentación completa evita errores.

- **Tiempo medio de despliegue:** reducción en horas/días del proceso de despliegue. (DevOps busca reducir este tiempo a fracciones mediante automatización.)
- **Tasa de fallos en producción:** número de incidentes post-despliegue respecto al total. La meta es disminuir errores en producción gracias a los tests automáticos.
- **Niveles de eficiencia y mantenibilidad:** mediciones internas (por ejemplo, tiempo de respuesta de la aplicación, índice de deuda técnica) siguiendo ISO 25010.
- **Satisfacción del equipo TI:** encuesta interna post-capacitación y post-implementación (meta mayor a 90% satisfacción).

Estos indicadores se medirán periódicamente (mensual o trimestral). Por ejemplo, se revisará mensualmente la frecuencia de despliegues automatizados y la cobertura de documentación. Se emplearán herramientas de recolección para asegurar datos confiables. Según la literatura, medir estas métricas en cada etapa es esencial para evaluar el rendimiento DevOps. Se establecen límites aceptables (por ejemplo, mantener en lo mínimo los errores críticos mensuales) y se ajustará el plan si se exceden.

6.6 CRONOGRAMA DE IMPLEMENTACIÓN

A continuación, se presenta una posible tabla de cronograma de seis meses (24 semanas), organizada por fases, actividades y responsables:

Tabla 14. Cronograma de implementación

Fase/Actividad	Duración	Responsables
Fase 1: Diagnóstico y planificación	1 mes (Sem. 1-4)	Equipo Dev, Consultor DevOps
Evaluación de procesos y herramientas actuales	2 semanas	Equipo Dev, Consultor DevOps
Definición del alcance y recursos necesarios	2 semanas	Equipo de proyecto, TI
Fase 2: Diseño de la solución DevOps	1 mes (Sem. 5-8)	Arquitecto TI, Consultor DevOps
Selección de herramientas CI/CD y IaC	2 semanas	Arquitecto TI
Diseño de pipelines y repositorios	2 semanas	Equipo de Desarrollo
Fase 3: Implementación técnica	2 meses (Sem. 9-16)	Equipo DevOps, Equipo IaC
Desarrollo de pipelines CI/CD	4 semanas	Equipo DevOps
Implementación de IaC (Terraform)	3 semanas	Equipo IaC
Contenerización de aplicaciones	3 semanas	Equipo DevOps
Integración de documentación	2 semanas	Equipo Dev, Equipo DevOps
Gestión de deuda técnica	4 semanas	Equipo Dev, Equipo DevOps
Fase 4: Monitoreo y métricas	1 mes (Sem. 17-20)	Equipo DevOps, Calidad
Configuración de herramientas de monitoreo	2 semanas	Equipo DevOps
Creación de dashboards de métricas de calidad	2 semanas	Equipo de Calidad
Fase 5: Capacitación	Pararela (Sem. 5-18)	Consultores externos, Equipo Dev
Cursos Udemy (AWS, DevOps, IaC) para 76 personas	4 semanas	Instructores externos (En línea)
Fase 6: Evaluación y ajustes finales	1 mes (Sem. 21-24)	Equipo de Proyecto
Revisión de resultados, ajustes y documentación final	2 semanas	Equipo Dev, Consultor DevOps
Presentación de reportes de cierre	2 semana	Equipo de Proyecto

Fuente: Elaboración propia (2026).

Este cronograma es indicativo y puede ajustarse. Las fases pueden solaparse entre sí: por ejemplo, la capacitación se realiza de forma paralela a la implementación técnica para no retrasar el proyecto. Cada actividad tiene un responsable principal (interno o consultor externo) para asegurar seguimiento.

6.7 PRESUPUESTO E IMPACTO DEL PRESUPUESTO

El cuadro siguiente detalla los costos estimados del plan, considerando los precios de mercado en USD:

Tabla 15. Desglose del presupuesto

Concepto	Descripción	Costo Unitario (USD)	Cantidad	Costo Total (USD)
Capacitación DevOps (Udemy)	Cursos AWS, DevOps, IaC y buenas prácticas (3 cursos por persona)	50.00 USD	228	11,400.00 USD
Consultoría DevOps externa	Especialista en DevOps, IaC y nube (100 horas aprox.)	150.00 USD	100	15,000.00 USD
Herramientas de monitoreo	Implementación de Prometheus, Grafana o Dynatrace	200.00 USD	1	200.00 USD
Herramientas de documentación	Uso de Confluence	- USD	1	- USD
Gestión de deuda técnica	Herramientas (SonarQube) y refactorización	300.00 USD	1	300.00 USD
Infraestructura en la nube (AWS/Azure)	durante 6 meses	600.00 USD	6	3,600.00 USD
Infraestructura on-premise	Nodos físicos/virtuales para contenedores	2,500.00 USD	2	5,000.00 USD
Plataforma de contenedores (OpenShift)	Licenciamiento base y configuración para despliegue contenerizado	3,000.00 USD	1	3,000.00 USD
Gastos operativos	Soporte, conectividad, imprevistos	1,000.00 USD	1	1,000.00 USD
TOTAL				39,500.00 USD

Fuente: Elaboración propia (2026).

Impacto del presupuesto: En tecnología, la inversión inicial suele ser alta debido a todos los componentes que se requieren, adoptar prácticas DevOps en la entidad financiera de manera correcta gracias a las inversiones en capacitaciones y consultorías solicitadas, contribuye a optimizar los costos asociados a la gestión de tecnologías de la información, tener una alta disponibilidad de los sistemas y reducir errores, por ello el retorno de la inversión o ROI es significativo ya que puede reducir los costos de TI entre un 15% a 25% (DevOps Institute, 2021).

En este sentido, si la entidad logra una reducción en los gastos operativos como resultado de la automatización, la mejora en los procesos y la disminución de incidencias, se recuperará la inversión en un período de un año.

6.8 CONCORDANCIA DE SEGMENTOS DE LA TESIS CON LA PROPUESTA

En el siguiente cuadro se presentan las partes más importantes de cada capítulo, reflejando la concordancia que existe entre cada segmento, desde el título de la investigación hasta concluir con los objetivos de la propuesta.

Tabla 16. Concordancia de la propuesta con la tesis

Capitulo I			Capitulo II	Capitulo III			Capitulo V	Capitulo VI	
Título de la Investigación	**Objetivo General**	**Objetivos Específicos**	**Teorías / Metodologías de sustento**	**Variables**	**Población**	**Técnicas**	**Conclusiones**	**Nombre de la propuesta**	**Objetivos de la propuesta**
Nivel de implementación de prácticas DevOps y la calidad del software en una entidad financiera	Determinar la relación entre el grado de implementación de prácticas DevOps y la calidad del software según el modelo ISO/IEC 25010.	Cuantificar el nivel de implementación de prácticas DevOps mediante CI/CD, Infraestructura como Código y contenedores	Modelos DevOps (integración continua, entrega continua), Infraestructura como Código, Contenerización ; modelo de calidad ISO/IEC 25010.	Variable independiente: Implementación de prácticas DevOps	76 profesionales de TI de la entidad financiera	Encuesta estructurada (escala Likert)	Existe relación entre la adopción de prácticas DevOps y la mejora en la calidad del software	Modelo de implementación de prácticas DevOps para el fortalecimiento de la calidad del software	Implementar pipelines CI/CD automáticos con pruebas integradas; adoptar infraestructura como código y contenerización para estandarizar entornos; mejorar la documentación de procesos; establecer medición continua de métricas de calidad ISO 25010.
		Analizar la relación entre CI/CD y la mantenibilidad del software		Variable dependiente: Calidad del software			La automatización CI/CD contribuye a mejorar la mantenibilidad del software		
		Evaluar la relación entre infraestructura como código y contenedores con la eficiencia del software		Dimensiones: mantenibilidad, eficiencia y seguridad			La infraestructura como código y la contenerización mejoran la eficiencia del sistema		
		Examinar el impacto del despliegue automatizado en la seguridad del software		Métricas de calidad ISO 25010			La adopción de prácticas DevOps fortalece la seguridad del software		

Fuente: Elaboración propia (2026).

REFERENCIAS BIBLIOGRÁFICAS

25010:2011, I. (. (s.f.). Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models. International Organization for Standardization. Obtenido de https://www.iso.org/standard/35733.html

Forsgren, N, & Kersten, M. (s.f.). DevOps metrics. Communications of the ACM, 61(4), 44-48. doi:https://doi.org/10.1145/3159169

Forsgren, N, Storey, M. A., Maddila, C, Zimmermann, T., Houck, B., & Butler, J. (2024). *2024 State of DevOps Report. Google Cloud DevOps Research and Assessment (DORA).* Obtenido de https://dora.dev/

Hernández-Sampieri, R., Fernández-Collado, C., & Baptista-Lucio, P. (2014). *Metodología de la investigación.* McGraw-Hill Education.

López, L., Burgués, X., Martínez-Fernández, S., Vollmer, A. M., Behutiye, W., Karhapää, P., . . . Oivo, M. (2022). Quality measurement in agile and rapid software development: A systematic mapping. Journal of Systems and Software, 186, 111187. doi:https://doi.org/10.1016/j.jss.2021.111187

Lwakatare, L. E., Kylmäkoski, T., Kuvaja, P., & Oivo, M. (2019). DevOps in practice: A multiple case study of five companies. Information and Software Technology, 114, 217-230. doi:https://doi.org/10.1016/j.infsof.2019.06.010

Mishra, A., & Otaiwi, Z. (2020). DevOps and software quality: A systematic mapping. Computer Science Review, 38, 100308. doi:https://doi.org/10.1016/j.cosrev.2020.100308

Mohammad, S. M. (2017). Improve software quality through practicing DevOps. Proceedings of the 2017 International Conference on Advances in ICT for Emerging Regions (ICTer), 42-47. Obtenido de https://www.researchgate.net/publication/322515647_Improve_software_quality_through_practicing_DevOps

Peña, J. J., Ávila, M., García-Mireles, & G. A. (2024). DevOps y la medición de la calidad del producto de software: Hallazgos preliminares. *RISTI - Revista Ibérica de Sistemas e Tecnologías de Informação, 53, 37-52.* doi:https://doi.org/10.17013/risti.53.37-52

Rahman, A. A. U, & Williams, L. (2016). Software security in DevOps: Synthesizing practitioners' perceptions and practices. *2016 IEEE/ACM International Workshop on Continuous Software Evolution and Delivery (CSED), 70-76.* doi:https://doi.org/10.1145/2896941.2896946

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering, 14(2), 131-164. doi:https://doi.org/10.1007/s10664-008-9102-8

Rzig, D. E., Hassan, F., & Kessentini, M. (2022). An empirical study on ML DevOps adoption trends, efforts, and benefits analysis. Information and Software Technology, 149, 106939. doi:https://doi.org/10.1016/j.infsof.2022.106939

Senapathi, M., & Buchan, J. . (2018). DevOps capabilities, practices, and challenges: Insights from a case study. Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering (EASE 2018), 57-67. doi:https://doi.org/10.1145/3210459.3210465

Şenkal, C., Yolaçan, B., Herkiloğlu, K., & Yılmaz, A. E. . (2023). Assessment of DevOps pipeline with quality perspective and application of ISO/IEC 25010 gap analysis. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 16(1), 81-86. doi:https://doi.org/10.54525/tbbmd.1118200

Yin, R. K. (2003). *Case study research: Design and methods (3rd ed.).* Sage Publications.

Zhang, Y., Huang, J., Wang, X., Li, X., & Zhang, Y. (2023). A framework for automating the measurement of DevOps Research and Assessment (DORA) metrics. Proceedings of the 38th IEEE/ACM International Conference on Automated Software Engineering (ASE), 1847-1859. Obtenido de https://www.researchgate.net/publication/376432131_A_Framework_for_Automating_the_Measurement_of_DevOps_Research_and_Assessment_DORA_Metrics

Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. IEEE Software, 33(3), 94–100. https://doi.org/10.1109/MS.2016.68

Mishra, A., & Otaiwi, Z. (2020). DevOps and software quality: A systematic mapping. Computer Science Review, 38, 100308. https://doi.org/10.1016/j.cosrev.2020.100308

García-Mireles, G. A., Peña Olivero, N., & Ávila-George, H. (2024). DevOps y la medición de la calidad del producto de software: Hallazgos preliminares. RISTI – Revista Ibérica de Sistemas e Tecnologías de Informação, (53), 37-52. https://doi.org/10.17013/risti.53.37-52

López, L., Burgués, X., Martínez-Fernández, S., Vollmer, A. M., Behutiye, W., Karhapää, P., Franch, X., Rodríguez, P., & Oivo, M. (2022). Quality measurement in agile and rapid software development: A systematic mapping. Journal of Systems and Software, 186, 111187. https://doi.org/10.1016/j.jss.2021.111187

Prates, L., Faustino, J., Silva, M., & Pereira, R. (2019). DevSecOps metrics. En S. Wrycza & J. Maślankowski (Eds.), Information Systems: Research,

Development, Applications, Education.
Lecture Notes in Business Information Processing (pp. 77–90). Springer.b
https://repositorio.iscte-iul.pt/handle/10071/25425

Orozco-Garcés, C.-E., Pardo-Calvache, C.-J., & Suescún-Monsalve, E. (2022). Metrics model to complement the evaluation of DevOps in software companies. Revista Facultad de Ingeniería, 31(62), e201.
https://doi.org/10.19053/01211129.v31.n62.2022.14766

ISO/IEC 25010. (2011). Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — System and software quality models. International Organization for Standardization.

Redgate. (2020). The State of Database DevOps 2020. Redgate Software Ltd.
https://www.red-gate.com/devops-report-2020

Aguirre Flórez, M. Á. (2025). Gestión de calidad, desarrollo sostenible y DevOps en tecnología: Práctica empresarial… [Proyecto de grado, Universidad Tecnológica de Pereira]. Repositorio Institucional UTP.
https://repositorio.utp.edu.co/bitstreams/533dd0f7-c1b2-47ab-9983-6406a285cb34/download

Amaro, R., Pereira, R., & da Silva, M. M. (2023). Capabilities and metrics in DevOps: A design science study. Information & Management, 60(5), 103809.
https://doi.org/10.1016/j.im.2023.103809

Erich, F., Amrit, C., & Daneva, M. (2017). A qualitative study on DevOps from a management perspective. Journal of Systems and Software, 141, 1–16.
https://doi.org/10.1016/j.jss.2018.03.065

Forsgren, N., Humble, J., & Kim, G. (2018). Accelerate: The science of lean software and DevOps – excerpt. IT Revolution Press. https://itrevolution.com/wp-content/uploads/2022/06/ACC_excerpt.pdf

Gómez-Aguirre, M., Hernández Alarcón, R., Solís Carmona, E., & Álvarez Valentín, H. (2023). ¿Qué es DevOps? Definición y características. Revista Innova Ingeniería, 4(1), 45–56.
https://innovaingenieria.uagro.mx/innova/index.php/innova/article/download/134/86/

ISO/IEC. (2023). ISO/IEC 25010:2023 — Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. International Organization for Standardization. https://www.iso.org/standard/86783.html

Mohan, V., & Othmane, L. B. (2020). Security practices in DevSecOps: An industry perspective. IEEE Security & Privacy, 18(4), 72–77. https://doi.org/10.1109/MSEC.2020.2965411

Offerman, T., Blinde, R., Stettina, C. J., & Visser, J. (2022). A study of adoption and effects of DevOps practices. 2022 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), 1–9. https://doi.org/10.1109/ICE/ITMC54999.2022.9845162

Rodríguez, P., Haghighatkhah, A., Lwakatare, L. E., Teppola, S., Suomalainen, T., Eskeli, J., Karvonen, T., Kuvaja, P., Verner, J. M., & Oivo, M. (2017). Continuous deployment of software-intensive products and services: A systematic mapping study. Journal of Systems and Software, 123, 263–291. https://doi.org/10.1016/j.jss.2015.12.015

Leite, L. A. F., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2019). A survey of DevOps concepts and challenges. ACM Computing Surveys, 52(6), Article 127. https://doi.org/10.1145/3359981

Atlas.ti Team. (n.d.). Confidencialidad y privacidad en la investigación. Guía de investigación cualitativa. https://atlasti.com/es/guias/guia-investigacion-cualitativa-parte-1/confidencialidad-y-privacidad

Tatineni, S. (2023). Applying DevOps practices for quality and reliability improvement in cloud-based systems. TIJER – International Research Journal, 10(11), a374–a380. https://tijer.org/TIJER/papers/TIJER2311042

DevOps Institute. (2021). DevOps Foundation® (DOFD) exam study guide (v3.3). PeopleCert. https://www.devopsinstitute.com/wp-content/uploads/2021/03/DOFD-v3.3-English-Exam-Study-Guide_01Mar2021.pdf

ANEXOS

ANEXO 1. INSTRUMENTO PARA LA RECOLECCIÓN DE DATOS

Nivel de Implementación de Prácticas DevOps y Calidad del Producto de Software

Estimado/a participante:

El presente cuestionario forma parte de una investigación académica desarrollada en el marco de la Maestría en Gestión de Tecnologías de la Información. Su objetivo es evaluar el nivel de implementación de prácticas DevOps y su relación con la calidad del producto de software en una entidad financiera.

La información recopilada será utilizada exclusivamente con fines académicos. Sus respuestas son confidenciales y anónimas, y no permitirán identificar a personas ni a la institución. No existen respuestas correctas o incorrectas; le solicitamos responder con sinceridad de acuerdo con su experiencia y percepción.

El cuestionario utiliza una **escala tipo Likert de cinco puntos**, donde:

- 1 = Nunca
- 2 = Casi nunca
- 3 = Algunas veces
- 4 = Casi siempre
- 5 = Siempre

Agradecemos de antemano su valiosa colaboración.

1. Automatización Integración y Despliegue Continuo (CI/CD) *

	Nunca	Casi nunca	Algunas veces	Casi siempre	Siempre
Los cambios que los desarrolladores hacen en el código se integran automáticamente en un repositorio central compartido por todo el equipo.	○	○	○	○	○
Cada vez que se hace un cambio en el código, se ejecutan automáticamente procesos que compilan, prueban y validan el software sin intervención manual.	○	○	○	○	○
Las pruebas automatizadas validan de forma completa los cambios más importantes y críticos del sistema antes de llegar a producción.	○	○	○	○	○
Los errores en el código se detectan automáticamente en las primeras etapas del proceso, antes de que el software llegue a los usuarios finales.	○	○	○	○	○
Los resultados de las pruebas y validaciones automáticas son visibles y accesibles para todos los miembros del equipo técnico.	○	○	○	○	○

2. Nivel de implementación de Infraestructura como Código (IaC) *

	Nunca	Casi nunca	Algunas veces	Casi siempre	Siempre
La infraestructura tecnológica (servidores, redes, bases de datos) se define y despliega usando archivos de código o plantillas automatizadas en lugar de configuraciones manuales.	◯	◯	◯	◯	◯
Los ambientes de desarrollo, calidad y producción se crean de forma idéntica utilizando las mismas definiciones de código, garantizando que funcionen igual.	◯	◯	◯	◯	◯
Los cambios en la infraestructura se registran y versionan en un sistema de control (como Git), de la misma manera que se hace con el código del software.	◯	◯	◯	◯	◯
La documentación técnica de la infraestructura se mantiene actualizada y está sincronizada con las definiciones de código que realmente se utilizan.	◯	◯	◯	◯	◯
Los cambios propuestos en la infraestructura pasan por un proceso de revisión técnica por otros miembros del equipo antes de ser aplicados.	◯	◯	◯	◯	◯

3. Nivel de adopción de Contenerización *

	Nunca	Casi nunca	Algunas veces	Casi siempre	Siempre
Las aplicaciones y servicios se empaquetan y despliegan de forma estandarizada en contenedores (tecnologías como Docker o similares).	○	○	○	○	○
El uso de contenedores ha logrado que las aplicaciones se comporten de manera consistente sin importar si están en desarrollo, calidad o producción.	○	○	○	○	○
Las nuevas versiones del software se despliegan de forma más rápida y confiable gracias al uso de contenedores.	○	○	○	○	○
Existe una gestión efectiva de los contenedores que permite controlar automáticamente aspectos como el escalamiento, la disponibilidad y el balanceo de carga.	○	○	○	○	○
El monitoreo continuo de los contenedores permite al equipo detectar y diagnosticar problemas operativos antes de que afecten a los usuarios.	○	○	○	○	○

4. Mantenibilidad del Software *

	Nunca	Casi nunca	Algunas veces	Casi siempre	Siempre
Cuando se necesita hacer cambios en el software, estos se pueden implementar sin causar problemas o errores inesperados en otras partes del sistema.	○	○	○	○	○
La forma en que está organizado el código permite localizar rápidamente las partes que necesitan modificarse y entender cómo funcionan.	○	○	○	○	○
Cuando se encuentran errores en el sistema, estos se pueden identificar, diagnosticar y corregir de forma eficiente sin comprometer la estabilidad general.	○	○	○	○	○
El equipo gestiona de forma continua la deuda técnica del sistema mediante actividades planificadas de mejora y refactorización del código.	○	○	○	○	○
El código fuente incluye documentación técnica que ayudan a comprender su funcionamiento y facilitan el trabajo de mantenimiento futuro.	○	○	○	○	○

5. Eficiencia del desempeño del software *

	Nunca	Casi nunca	Algunas veces	Casi siempre	Siempre
El sistema mantiene tiempos de respuesta adecuados para el negocio, incluso cuando enfrenta diferentes niveles de carga de trabajo.	◯	◯	◯	◯	◯
El software utiliza de manera eficiente los recursos tecnológicos disponibles, sin desperdiciarlos.	◯	◯	◯	◯	◯
El desempeño del sistema se mantiene estable y consistente durante los períodos de mayor demanda operativa del negocio.	◯	◯	◯	◯	◯
Cuando se necesita mayor capacidad de procesamiento, el sistema puede crecer de forma proporcional agregando más recursos tecnológicos.	◯	◯	◯	◯	◯
Los indicadores de desempeño del sistema cumplen de manera consistente con los niveles de servicio que se han establecido para el negocio.	◯	◯	◯	◯	◯

6. Seguridad del software *

	Nunca	Casi nunca	Algunas veces	Casi siempre	Siempre
Las vulnerabilidades de seguridad en el software se identifican y corrigen de forma oportuna antes de que el sistema llegue a producción.	○	○	○	○	○
Los mecanismos de autenticación y control de acceso aseguran que solo los usuarios autorizados puedan acceder a los recursos según el nivel de privilegios que les corresponde.	○	○	○	○	○
Las prácticas de seguridad (análisis de código, revisión de dependencias, pruebas de seguridad) están integradas en todas las etapas del proceso de desarrollo.	○	○	○	○	○
Las configuraciones de seguridad del sistema protegen de manera efectiva la confidencialidad, integridad y disponibilidad de la información.	○	○	○	○	○
Se realizan auditorías y evaluaciones de seguridad de forma regular para verificar que se cumplen las políticas y controles establecidos.	○	○	○	○	○