Recibido: 10/01/2026 Aprobado:15/02/2026

DOI: https://doi.org/10.70208/3007.8245.v6.n1.356

DevSecOps como arquitectura integral para la ingeniería de software moderna: automatización, seguridad continua y resiliencia organizacional

Maria Teodolinda Ortega Ovalle
maria.ortegao@up.ac.pa
https://orcid.org/0009-0000-3629-9751
Universidad de Panamá
Facultad de Informática, Electrónica y Comunicación
Departamento de Informática

RESUMEN

DevSecOps ha emergido como un paradigma fundamental dentro de la ingeniería de software contemporánea, integrando prácticas de desarrollo, operaciones y seguridad en un flujo continuo y automatizado. Este artículo presenta una revisión conceptual y documental sobre la evolución teórica y práctica de DevSecOps como una arquitectura integral para organizaciones que buscan resiliencia, escalabilidad y protección frente a amenazas cada vez más sofisticadas. El estudio examina los principios esenciales del modelo, incluyendo integración continua, entrega continua, automatización de pruebas, infraestructura como código y monitoreo inteligente. Asimismo, analiza la incorporación de controles de seguridad desde las primeras etapas del ciclo de vida del software, en contraste con enfoques tradicionales donde la seguridad se añadía al final del proceso. La metodología combina una revisión documental sistemática con un análisis comparativo de los modelos DevOps, SecOps y DevSecOps, lo que permite identificar diferencias estructurales, operativas y culturales. Los hallazgos muestran que DevSecOps no solo reduce vulnerabilidades, sino que también mejora la eficiencia organizacional mediante pipelines automatizados y prácticas colaborativas. Se presentan aplicaciones prácticas en sectores como banca, salud y servicios gubernamentales, donde la seguridad es crítica. El estudio también discute limitaciones relacionadas con la cultura organizacional, los costos de adopción y la escasez de talento especializado. En conjunto, el análisis concluye que DevSecOps constituye un enfoque indispensable para organizaciones que buscan desarrollar software seguro, ágil y sostenible en entornos altamente dinámicos.

Palabras Claves: DevSecOps, integración continua, seguridad informática, automatización, ingeniería de software

DevSecOps as an Integrated Architecture for Modern Software Engineering: Automation, Continuous Security, and Organizational Resilience

ABSTRACT

DevSecOps has emerged as a foundational paradigm in contemporary software engineering, integrating development, operations, and security practices into a continuous and automated workflow. This article presents a conceptual and documentary review of the theoretical and practical evolution of DevSecOps as an integrated architecture for organizations seeking resilience, scalability, and protection against increasingly sophisticated threats. The study examines core principles of the model, including continuous integration, continuous delivery, automated testing, infrastructure as code, and intelligent monitoring. It also analyzes the incorporation of security controls from the earliest stages of the software development lifecycle, contrasting this approach with traditional models in which security was added at the end of the process. The methodology combines a systematic documentary review with a comparative analysis of DevOps, SecOps, and DevSecOps models, enabling the identification of structural, operational, and cultural differences. Findings indicate that DevSecOps not only reduces vulnerabilities but also enhances organizational efficiency through automated pipelines and collaborative practices. Practical applications are highlighted in sectors such as banking, healthcare, and government services, where security is critical. The study also discusses limitations related to organizational culture, adoption costs, and shortages of specialized talent. The analysis concludes that DevSecOps constitutes an essential approach for organizations aiming to develop secure, agile, and sustainable software in highly dynamic environments.

Keywords: DevSecOps, continuous integration, cybersecurity, automation, software engineering

INTRODUCCIÓN

La ingeniería de software contemporánea atraviesa un punto de inflexión marcado por la creciente complejidad de los sistemas, la aceleración de los ciclos de entrega y la intensificación de las amenazas cibernéticas. En este escenario, los enfoques tradicionales de desarrollo y seguridad han demostrado ser insuficientes para responder a las demandas actuales de agilidad, escalabilidad y protección integral. Durante décadas, la separación estructural entre desarrollo, operaciones y seguridad generó procesos fragmentados, tiempos de entrega prolongados y vulnerabilidades que solo se identificaban en etapas tardías del ciclo de vida del software (Kim et al., 2022). Esta desconexión impulsó la transición hacia modelos colaborativos como DevOps, cuyo propósito fue integrar desarrollo y operaciones para mejorar la eficiencia, la calidad y la estabilidad de los sistemas (Bass et al., 2023).

Sin embargo, la ausencia de un componente de seguridad desde las primeras fases del proceso reveló limitaciones críticas en un entorno donde los ataques cibernéticos se han vuelto más sofisticados, persistentes y automatizados. Estudios recientes muestran que la velocidad de entrega propia de DevOps puede amplificar riesgos si no se incorporan controles de seguridad continuos y automatizados (Bautista Ramos & Yoo, 2025; Rose et al., 2020). Esta brecha dio origen a DevSecOps, una evolución natural que integra la seguridad como un elemento transversal en todas las etapas del ciclo de vida del software.

DevSecOps no solo incorpora herramientas automatizadas de análisis de vulnerabilidades, pruebas estáticas y dinámicas, y monitoreo continuo, sino que también promueve una cultura organizacional basada en la colaboración, la responsabilidad compartida y la gestión proactiva del riesgo (Shahin et al., 2017; Koneru, 2021). La automatización se convierte en un pilar fundamental, permitiendo que los controles de seguridad se ejecuten de manera continua sin ralentizar los procesos de integración y entrega continua (CI/CD). Este cambio de paradigma transforma la seguridad en un habilitador de la innovación, en lugar de un obstáculo o una etapa final reactiva.

La relevancia de DevSecOps ha aumentado significativamente en sectores donde la seguridad es crítica, como banca, salud, comercio electrónico y servicios gubernamentales. La adopción de arquitecturas basadas en microservicios, contenedores y computación en la nube ha ampliado la

superficie de ataque, lo que exige mecanismos de protección más dinámicos y adaptativos (Microsoft, 2023; Red Hat, 2023). En este contexto, DevSecOps ofrece un marco integral que combina prácticas técnicas, metodológicas y culturales para fortalecer la resiliencia organizacional. Además, el incremento de regulaciones internacionales en materia de protección de datos como GDPR, HIPAA y normativas latinoamericanas emergentes ha impulsado a las organizaciones a adoptar modelos que garanticen cumplimiento normativo desde el diseño, alineándose con principios de security by design y Zero Trust (Rose et al., 2020).

A pesar de sus beneficios, la implementación de DevSecOps presenta desafíos significativos. Entre ellos destacan la resistencia cultural, la falta de talento especializado, los costos iniciales de adopción y la complejidad de integrar herramientas heterogéneas en pipelines automatizados (Erich et al., 2017; Mohammed et al., 2025). Estas barreras justifican la necesidad de estudios que analicen el modelo desde una perspectiva integral, comparándolo con enfoques previos y evaluando su impacto en la seguridad, la eficiencia y la sostenibilidad del desarrollo de software. Investigaciones recientes han demostrado que la integración temprana de seguridad incluyendo enfoques como Shift Left Security reduce significativamente el costo de remediación y mejora la calidad del software (Kaithe, 2025; Rajapakse et al., 2021).

El propósito de este artículo es analizar DevSecOps como arquitectura integral dentro de la ingeniería de software moderna, examinando sus fundamentos conceptuales, su evolución histórica, sus componentes técnicos y sus implicaciones organizacionales. Se plantea como pregunta central: ¿cómo contribuye DevSecOps a mejorar la seguridad, eficiencia y resiliencia en el desarrollo de software contemporáneo? Para responderla, se desarrolla un marco teórico actualizado, una metodología basada en revisión documental sistemática y análisis comparativo, y una discusión de resultados que integra evidencia reciente (2017–2025). Asimismo, se incluyen gráficos conceptuales y tablas comparativas que facilitan la comprensión de los elementos clave del modelo.

Finalmente, el artículo se estructura de la siguiente manera: el marco teórico presenta los fundamentos y la evolución del enfoque; la metodología describe el proceso de revisión y análisis; los resultados sintetizan los hallazgos principales; la discusión interpreta los resultados en relación con la literatura;

las aplicaciones prácticas muestran su utilidad en distintos sectores; las limitaciones identifican los desafíos del modelo; y las conclusiones ofrecen una síntesis crítica y proponen líneas futuras de investigación.

Marco Teórico

Antecedentes históricos del desarrollo y la seguridad

La evolución de DevSecOps no puede comprenderse sin analizar primero la trayectoria histórica del desarrollo de software y la seguridad informática. Durante décadas, los modelos tradicionales como el ciclo en cascada separaban estrictamente las fases de análisis, diseño, implementación, pruebas y despliegue, lo que generaba procesos lentos y vulnerabilidades detectadas tardíamente (Pressman & Maxim, 2022). La seguridad, en este contexto, se concebía como una etapa final, reactiva y centrada en auditorías posteriores al desarrollo.

Con la llegada de las metodologías ágiles a principios de los años 2000, surgió la necesidad de ciclos más cortos, entregas incrementales y colaboración continua. Sin embargo, la seguridad continuó siendo un componente aislado, lo que provocó brechas significativas en sistemas cada vez más expuestos a amenazas (Kim et al., 2022). La creciente complejidad de las arquitecturas distribuidas, la adopción masiva de la computación en la nube y el incremento de ataques automatizados impulsaron la transición hacia modelos más integrados.

Tabla 1. Comparación entre DevOps, SecOps y DevSecOps.

CARACTERÍSTICA	DEVOPS	SECOPS	DEVSECOPS
ENFOQUE PRINCIPAL	Velocidad y entrega	Seguridad y monitoreo	Integración continua de seguridad
PARTICIPACIÓN	Desarrollo + Operaciones	Equipos de seguridad	Desarrollo + Operaciones + Seguridad
AUTOMATIZACIÓN	Alta	Media	Muy alta
SEGURIDAD	Al final del ciclo	Reactiva	Proactiva y continua
HERRAMIENTAS TÍPICAS	Jenkins, Docker	SIEM, IDS	SAST, DAST, IaC scanners
CULTURA	Colaborativa	Especializada	Colaborativa con enfoque de riesgo

Evolución conceptual: de DevOps a DevSecOps

DevOps emergió como un enfoque que integra desarrollo (Dev) y operaciones (Ops) para mejorar la velocidad, calidad y estabilidad del software. Su objetivo principal es eliminar silos organizacionales mediante automatización, integración continua (CI) y entrega continua (CD) (Bass et al., 2023). No obstante, DevOps no incorporaba explícitamente la seguridad como parte del flujo de trabajo.

DevSecOps surge como una extensión natural que incorpora la seguridad (Sec) como un componente transversal. Su premisa fundamental es que la seguridad debe integrarse desde el diseño, mantenerse durante todo el ciclo de vida y automatizarse en la mayor medida posible, alineándose con principios de seguridad continua y responsabilidad compartida (Rajapakse et al., 2021). Este enfoque se conoce como shift-left security, que implica mover los controles de seguridad hacia las primeras etapas del desarrollo (Kaithe, 2025).

Tabla 2. Evolución conceptual del desarrollo de software

MODELO	CARACTERÍSTICAS	LIMITACIONES	APORTE A DEVSECOPS
CASCADA	Secuencial, rígido	Vulnerabilidades detectadas tardíamente	Base histórica del ciclo de vida
ÁGIL	Iterativo, colaborativo	Seguridad aislada	Promueve colaboración y entregas rápidas
DEVOPS	Automatización, CI/CD	Sin seguridad integrada	Integra desarrollo y operaciones
DEVSECOPS	Seguridad continua, automatización avanzada	Requiere madurez organizacional	Modelo integral con seguridad desde el diseño

La literatura reciente identifica varios principios esenciales. En primer lugar, la seguridad se concibe como una responsabilidad compartida, donde todos los equipos participan en la identificación, mitigación y prevención de riesgos, lo cual coincide con los hallazgos de estudios sobre integración de seguridad en entornos DevOps y DevSecOps (Rajapakse et al., 2021). En segundo lugar, la automatización de controles constituye un pilar central, incluyendo análisis estático, dinámico, escaneo de dependencias, infraestructura como código (IaC) y pruebas de seguridad automatizadas, prácticas

ampliamente documentadas en investigaciones sobre pipelines seguros y automatización de seguridad (Koneru, 2021).

Otro principio clave es la integración continua de seguridad, donde los controles se ejecutan en cada commit, merge o despliegue, alineándose con el enfoque de seguridad continua propuesto en marcos modernos de DevSecOps (Bautista Ramos & Yoo, 2025). Asimismo, la observabilidad y el monitoreo continuo permiten detectar anomalías y responder a incidentes mediante herramientas SIEM, métricas de seguridad y alertas inteligentes, elementos que se consideran esenciales en arquitecturas Zero Trust y entornos nativos en la nube (Rose et al., 2020). Finalmente, la infraestructura como código (IaC) facilita la gestión de entornos reproducibles y seguros, lo que resulta fundamental para mantener coherencia, trazabilidad y control en sistemas distribuidos (Microsoft, 2023).

Arquitectura técnica de DevSecOps

La arquitectura DevSecOps se compone de múltiples capas interconectadas que integran herramientas, procesos y prácticas de seguridad dentro de los flujos de CI/CD. Estas capas incluyen análisis de código, pruebas automatizadas, escaneo de vulnerabilidades, gestión de configuraciones, monitoreo continuo y mecanismos de respuesta, conformando un ecosistema que permite detectar y mitigar riesgos de manera temprana y sostenida (Shahin et al., 2017; Guda, 2024).

Relación entre DevSecOps y metodologías ágiles

DevSecOps se integra naturalmente con metodologías ágiles debido a su énfasis en iteraciones cortas, retroalimentación continua, colaboración multidisciplinaria y entrega incremental. En entornos Scrum y Kanban, los pipelines automatizados permiten validar aspectos de seguridad en cada sprint, garantizando que los incrementos de software cumplan con requisitos funcionales y no funcionales sin ralentizar el proceso de desarrollo (Bass et al., 2023). La literatura destaca que esta integración mejora la calidad del software y reduce el costo de remediación al incorporar controles de seguridad desde etapas tempranas del ciclo de vida (Kaithe, 2025).

Herramientas clave en DevSecOps

La literatura identifica herramientas agrupadas por función:

Tabla 3. Herramientas DevSecOps por categoría

CATEGORÍA	HERRAMIENTAS	FUNCIÓN
SAST	SonarQube, Checkmarx	Análisis de código
DAST	OWASP ZAP, Burp Suite	Pruebas dinámicas
SCA	Dependabot, Snyk	Escaneo de dependencias
IAC	Terraform, Ansible	Infraestructura automatizada
CONTENEDORES	Docker, Kubernetes	Orquestación
MONITOREO	Prometheus, Grafana	Observabilidad

Cultura organizacional en DevSecOps

La cultura es un componente crítico. Los estudios destacan la importancia de la responsabilidad compartida, la comunicación abierta, la capacitación continua, la madurez en seguridad y la gestión del cambio. Sin estos elementos, la adopción técnica fracasa, ya que la integración de seguridad en entornos DevOps depende tanto de factores humanos como de herramientas y procesos (Erich et al., 2017).

Desafíos y barreras de adopción

La literatura identifica varios desafíos, entre ellos la resistencia cultural, la falta de talento especializado, los costos iniciales elevados, la integración de herramientas heterogéneas y la complejidad en entornos multinube. Estos desafíos justifican la necesidad de estudios como el presente, especialmente considerando que la adopción de DevSecOps requiere cambios estructurales, técnicos y organizacionales (Rajapakse et al., 2021; Mohammed et al., 2025).

Tendencias emergentes en DevSecOps

Entre las tendencias más relevantes se encuentran la inteligencia artificial para seguridad predictiva, la aplicación de Zero Trust a pipelines, la automatización del cumplimiento normativo, la infraestructura inmutable y el shift-left extremo. Estas tendencias están redefiniendo el futuro del modelo y se alinean con la evolución de arquitecturas modernas y prácticas avanzadas de ingeniería de software (Bass et al., 2023; Rose et al., 2020; Guda, 2024).

METODOLOGÍA

La metodología empleada en este estudio se fundamenta en un enfoque cualitativo de carácter documental, orientado a analizar la evolución, los componentes y las implicaciones del modelo DevSecOps dentro de la ingeniería de software contemporánea. Dado que DevSecOps es un paradigma en constante desarrollo, se adoptó una revisión sistemática de literatura reciente, siguiendo lineamientos adaptados del protocolo PRISMA, con el fin de identificar tendencias, brechas, desafíos y buenas prácticas reportadas en investigaciones académicas y documentos técnicos especializados (Kitchenham et al., 2022). Esta aproximación permite garantizar rigor, transparencia y reproducibilidad en el proceso de selección, evaluación y síntesis de las fuentes.

Enfoque metodológico

El estudio adopta un enfoque descriptivo–analítico que combina la sistematización conceptual con el examen crítico de los elementos operativos del modelo DevSecOps. Este enfoque permite describir las características fundamentales del paradigma, analizar su relación con modelos previos como DevOps y SecOps, y evaluar su impacto en la seguridad, la eficiencia y la resiliencia organizacional. La dimensión descriptiva facilita la organización de conceptos clave, mientras que la dimensión analítica permite establecer comparaciones, identificar patrones y valorar la aplicabilidad del modelo en distintos contextos tecnológicos. Esta combinación metodológica asegura una comprensión integral del fenómeno estudiado, en línea con estudios previos sobre evolución de prácticas DevOps y DevSecOps (Erich et al., 2017).

Diseño de la revisión documental

La revisión documental se desarrolló siguiendo un proceso estructurado en cuatro fases, adaptado del protocolo PRISMA, con el propósito de asegurar coherencia metodológica y trazabilidad.

Identificación

Se recopilaron artículos científicos, libros especializados, informes técnicos y documentos de organismos internacionales publicados entre 2015 y 2025. Las bases de datos consultadas fueron IEEE Xplore, ACM Digital Library, Scopus, ScienceDirect y SpringerLink. También se incorporaron

whitepapers de empresas tecnológicas como Red Hat, GitLab y Microsoft, debido a su relevancia en la implementación práctica de DevSecOps.

Selección

Se aplicaron criterios de inclusión y exclusión para garantizar la pertinencia y calidad de las fuentes. Los criterios de inclusión abarcaron publicaciones entre 2015 y 2025, estudios relacionados con DevOps, SecOps o DevSecOps, investigaciones sobre automatización, CI/CD, IaC o seguridad continua, y documentos con metodología explícita o evidencia empírica. Los criterios de exclusión comprendieron publicaciones anteriores a 2015, artículos sin revisión por pares (excepto whitepapers técnicos) y documentos con enfoque exclusivamente comercial.

Evaluación

Las fuentes seleccionadas fueron evaluadas mediante criterios de calidad metodológica, relevancia temática y claridad conceptual. Se priorizaron estudios que aportaran comprensión profunda del modelo DevSecOps, así como investigaciones que presentaran casos de uso, métricas de seguridad o análisis comparativos. Esta evaluación se apoyó en literatura reciente que examina desafíos y soluciones en la adopción de DevSecOps (Rajapakse et al., 2021; Mohammed et al., 2025).

Síntesis

La información recopilada se organizó en categorías temáticas que facilitaron la estructuración de los resultados: evolución histórica del modelo, componentes técnicos (CI/CD, IaC, automatización de seguridad), cultura organizacional y colaboración, herramientas y prácticas recomendadas, desafíos y limitaciones, y aplicaciones sectoriales. Esta categorización permitió integrar los hallazgos de manera coherente y facilitar su análisis crítico.

Técnicas de análisis.

Se emplearon dos técnicas principales para el tratamiento de la información. El análisis comparativo permitió contrastar DevOps, SecOps y DevSecOps, identificando diferencias en enfoque, herramientas, cultura y resultados esperados. Esta técnica facilitó la construcción de tablas comparativas y modelos conceptuales que sintetizan las características distintivas de cada enfoque, en concordancia con estudios sobre arquitecturas modernas y seguridad en entornos distribuidos (Microsoft, 2023).

El análisis de contenido se utilizó para identificar patrones, conceptos recurrentes y tendencias emergentes en la literatura. Esta técnica permitió extraer información relevante sobre automatización, seguridad continua, integración de herramientas y prácticas organizacionales, fortaleciendo la interpretación de los hallazgos.

Alcance y delimitación.

El estudio se centra exclusivamente en organizaciones que desarrollan software mediante metodologías ágiles, entornos con pipelines CI/CD, arquitecturas basadas en contenedores, microservicios y nube, y prácticas de seguridad automatizada. No se incluyen modelos tradicionales en cascada ni estudios previos a 2015, debido a su limitada pertinencia para el estado actual del paradigma DevSecOps. Esta delimitación asegura que los hallazgos reflejen el contexto tecnológico contemporáneo.

Consideraciones éticas.

La investigación se basa exclusivamente en fuentes secundarias de acceso público y académico. No se recopilaron datos personales ni se realizaron experimentos con usuarios, por lo que no se identifican riesgos éticos. Se respetaron los derechos de autor mediante el uso adecuado de citas y referencias en formato APA 7.

RESULTADOS

Los resultados obtenidos a partir del análisis documental sistemático permiten identificar patrones, tendencias y hallazgos clave sobre la adopción, implementación y efectividad del enfoque DevSecOps en organizaciones contemporáneas. La síntesis de la literatura revisada (2015–2025) revela que DevSecOps ha evolucionado desde un conjunto de prácticas emergentes hacia un marco integral que combina automatización, cultura colaborativa y seguridad continua. Los resultados se presentan organizados en cuatro dimensiones principales: integración técnica, cultura organizacional, impacto en seguridad y eficiencia, y adopción sectorial. Además, se integran modelos conceptuales y comparaciones con marcos alternativos de desarrollo seguro, con el fin de fortalecer el análisis crítico.

Integración técnica y automatización.

La literatura coincide en que la automatización constituye el núcleo operativo de DevSecOps. Los estudios analizados destacan que la integración de herramientas de análisis estático (SAST), análisis

dinámico (DAST), escaneo de infraestructura como código (IaC) y monitoreo continuo permite detectar vulnerabilidades en etapas tempranas, reduciendo significativamente los costos de remediación (Koneru, 2021). Esta integración se articula dentro de pipelines CI/CD que incorporan controles de seguridad desde el diseño hasta la operación, consolidando un flujo continuo de verificación.

Tabla 4. Herramientas más utilizadas en pipelines DevSecOps (2022–2025)

CATEGORÍA	HERRAMIENTAS COMUNES	FUNCIÓN PRINCIPAL
SAST	SonarQube, Checkmarx	Análisis de código fuente
DAST	OWASP ZAP, Burp Suite	Pruebas de seguridad en ejecución
IAC SCANNERS	Terraform Scan, Checkov	Validación de configuraciones
CI/CD	Jenkins, GitLab CI, GitHub Actions	Integración y entrega continua
CONTENEDORES	Docker, Kubernetes	Orquestación y despliegue
MONITOREO	Prometheus, ELK Stack, Grafana	Observabilidad y alertas

Modelo conceptual del pipeline DevSecOps

El análisis documental permitió sintetizar un modelo conceptual que representa la arquitectura del pipeline DevSecOps. Este modelo se estructura en tres capas interdependientes. La primera capa corresponde al desarrollo, donde se incorporan análisis SAST, gestión de dependencias y validación de IaC. La segunda capa corresponde a la integración y entrega, donde se ejecutan pruebas DAST, escaneo de contenedores y validación de políticas. La tercera capa corresponde a la operación, donde se aplican monitoreo continuo, detección de anomalías y auditoría automatizada. Este modelo evidencia que DevSecOps funciona como una arquitectura continua que integra seguridad en todas las fases del ciclo de vida del software.

Cultura organizacional y colaboración

Uno de los hallazgos más consistentes es que DevSecOps no puede implementarse únicamente desde una perspectiva técnica. La literatura enfatiza que la cultura organizacional es un factor determinante

para el éxito del modelo (Erich et al., 2017). Los estudios revisados identifican tres elementos culturales clave: responsabilidad compartida, comunicación continua y aprendizaje organizacional. La evidencia muestra que las organizaciones que adoptan prácticas de colaboración multidisciplinaria logran mayor madurez en seguridad y reducen la fricción entre equipos, lo que se traduce en ciclos de entrega más estables y seguros.

Impacto en seguridad y eficiencia

Los resultados muestran que DevSecOps genera mejoras significativas tanto en seguridad como en eficiencia operativa. Los estudios analizados reportan reducciones del 40–60 % en vulnerabilidades críticas cuando se integran controles automatizados desde el inicio del ciclo de vida (Bautista Ramos & Yoo, 2025), disminución del tiempo de despliegue gracias a pipelines CI/CD optimizados, incremento en la resiliencia organizacional en entornos de nube y microservicios, y mejora en el cumplimiento normativo mediante auditorías automatizadas (Microsoft, 2023).

Tabla 5. Impacto comparado entre DevOps y DevSecOps

INDICADOR	DEVOPS	DEVSECOPS	MEJORA OBSERVADA
VULNERABILIDADES CRÍTICAS	Alta	Baja	-40 % a -60 %
TIEMPO DE DESPLIEGUE	Medio	Bajo	-25 % a -35 %
CUMPLIMIENTO NORMATIVO	Parcial	Alto	+30 %
RESILIENCIA OPERATIVA	Media	Alta	+40 %

Estos resultados muestran que DevSecOps no solo mejora la seguridad, sino que también optimiza la eficiencia operativa, lo que lo convierte en un enfoque especialmente valioso para organizaciones que requieren ciclos de entrega rápidos y seguros.

Comparación crítica con otros marcos de seguridad

El análisis comparativo permitió contrastar DevSecOps con marcos consolidados de seguridad en el desarrollo de software, como NIST SSDF, OWASP SAMM, Microsoft SDL e ISO/IEC 27034. NIST SSDF proporciona un conjunto de prácticas estructuradas para integrar seguridad en el ciclo de vida, pero no define mecanismos de automatización continua, lo que limita su aplicabilidad en entornos altamente dinámicos. OWASP SAMM ofrece un marco de madurez que permite evaluar capacidades organizacionales, pero su enfoque es más evaluativo que operativo. Microsoft SDL se centra en prácticas de diseño seguro, aunque no incorpora de manera explícita la integración con pipelines CI/CD. ISO/IEC 27034 establece lineamientos de seguridad en aplicaciones, pero su adopción suele ser más lenta debido a su carácter normativo.

En comparación, DevSecOps se distingue por su énfasis en la automatización, la integración continua y la colaboración entre equipos. Mientras los marcos tradicionales ofrecen lineamientos y buenas prácticas, DevSecOps proporciona un enfoque operativo que permite implementar seguridad de manera continua y escalable. Esta diferencia explica por qué DevSecOps ha sido adoptado con mayor rapidez en sectores donde la velocidad y la seguridad son críticas.

Adopción sectorial

La revisión documental evidencia que DevSecOps ha sido adoptado con mayor rapidez en sectores donde la seguridad es crítica. Entre los sectores con mayor adopción entre 2015 y 2025 se encuentran banca y fintech, salud, gobierno, comercio electrónico y telecomunicaciones. En estos sectores, la necesidad de proteger datos sensibles, cumplir regulaciones estrictas y gestionar infraestructuras distribuidas impulsa la adopción de prácticas de seguridad continua (Microsoft, 2023; Red Hat, 2023).

Tendencias emergentes identificadas

Los resultados también revelan tendencias clave que están moldeando el futuro de DevSecOps. La integración de inteligencia artificial para seguridad predictiva permite detectar anomalías y patrones de ataque mediante modelos de machine learning (Guda, 2024). El shift-left extremo incorpora la seguridad incluso antes de escribir código, mediante análisis de diseño (Kaithe, 2025). La infraestructura inmutable reduce riesgos al reemplazar entornos en lugar de modificarlos. La aplicación de Zero Trust

a pipelines CI/CD exige autenticación y validación continua de cada componente (Rose et al., 2020). La automatización del cumplimiento normativo permite generar evidencia de auditoría en tiempo real, alineándose con tendencias de automatización avanzada en ingeniería de software (Bass et al., 2023).

Síntesis general de resultados

Los hallazgos permiten concluir que DevSecOps mejora la seguridad de manera significativa, reduce tiempos y costos operativos, exige una cultura organizacional madura y depende fuertemente de la automatización. Se adapta especialmente bien a entornos de nube, contenedores y microservicios, aunque presenta desafíos de adopción relacionados con talento especializado, integración de herramientas y resistencia cultural. La comparación con otros marcos de seguridad evidencia que DevSecOps no reemplaza a estos modelos, sino que los complementa al proporcionar una arquitectura operativa que permite implementar sus lineamientos de manera continua y automatizada.

DISCUSIÓN

La discusión de los resultados permite comprender con mayor profundidad el papel que desempeña DevSecOps dentro de la ingeniería de software contemporánea. Los hallazgos muestran que este enfoque trasciende la dimensión técnica y se consolida como un marco sociotécnico integral, donde convergen automatización, cultura organizacional y gobernanza de seguridad. Esta perspectiva coincide con la literatura reciente, que destaca que la integración continua de seguridad, desarrollo y operaciones es indispensable para enfrentar la creciente complejidad de los sistemas modernos y la sofisticación de las amenazas cibernéticas (Rajapakse et al., 2021).

Integración técnica y automatización como eje central

Los resultados confirman que la automatización constituye el núcleo operativo de DevSecOps. La incorporación de herramientas SAST, DAST, SCA, IaC y monitoreo continuo permite detectar vulnerabilidades en etapas tempranas, lo cual se alinea con el principio de shift-left security ampliamente documentado (Kaithe, 2025). Esta automatización reduce tiempos de detección y remediación, disminuye la dependencia de procesos manuales y fortalece la resiliencia organizacional, especialmente en entornos basados en nube y microservicios (Microsoft, 2023). En este sentido, los resultados evidencian que la madurez técnica es un factor determinante para la adopción sectorial.

Cultura organizacional como factor crítico de adopción

La dimensión cultural emerge como un elemento decisivo. Los resultados muestran que la responsabilidad compartida, la comunicación continua y el aprendizaje organizacional son condiciones necesarias para la implementación efectiva de DevSecOps. Esto coincide con estudios que identifican la falta de madurez cultural como una de las principales barreras para su adopción (Erich et al., 2017). Aunque las herramientas automatizadas son esenciales, su impacto depende de la capacidad de los equipos para colaborar, comprender riesgos y adoptar prácticas de seguridad de manera proactiva. DevSecOps, por tanto, no debe entenderse únicamente como un conjunto de tecnologías, sino como un cambio estructural en la forma de concebir la seguridad dentro de las organizaciones.

Impacto en seguridad, eficiencia y cumplimiento normativo

Los resultados también muestran mejoras significativas en seguridad y eficiencia, lo cual coincide con estudios que reportan reducciones del 40–60 % en vulnerabilidades críticas tras la adopción de DevSecOps (Bautista Ramos & Yoo, 2025). La integración de controles automatizados en cada etapa del ciclo de vida del software contribuye a una mayor trazabilidad y facilita el cumplimiento de estándares internacionales. Esto es especialmente relevante en sectores regulados como banca, salud y gobierno, donde la seguridad es tanto un requisito técnico como legal (Microsoft, 2023). La evidencia sugiere que DevSecOps no solo mejora la postura de seguridad, sino que también optimiza procesos y reduce costos asociados a fallas tardías.

Adopción sectorial y brechas tecnológicas

La adopción sectorial observada muestra que DevSecOps se consolida principalmente en industrias donde la seguridad es crítica y la presión regulatoria es elevada. Esto coincide con investigaciones que señalan que la exposición a riesgos cibernéticos acelera la adopción del modelo (Red Hat, 2023). Sin embargo, la menor adopción en sectores como educación y manufactura revela brechas de infraestructura, recursos y madurez tecnológica. Estas diferencias abren oportunidades para investigaciones futuras orientadas a adaptar DevSecOps a contextos con capacidades limitadas, así como para desarrollar modelos de madurez sectorial más inclusivos.

Tendencias emergentes y evolución del modelo

Las tendencias emergentes identificadas como la integración de inteligencia artificial, la adopción de Zero Trust en pipelines CI/CD, la infraestructura inmutable y la automatización del cumplimiento normativo indican que DevSecOps continúa evolucionando hacia modelos más inteligentes, predictivos y autónomos. Estas transformaciones refuerzan la idea de que DevSecOps no es un estado final, sino un proceso de mejora continua que se adapta a las dinámicas tecnológicas globales (Guda, 2024; Rose et al., 2020; Bass et al., 2023).

Síntesis crítica

En conjunto, la discusión evidencia que DevSecOps constituye un enfoque integral que combina tecnología, cultura y procesos para fortalecer la seguridad y eficiencia del desarrollo de software. No obstante, su adopción requiere superar desafíos relacionados con talento especializado, resistencia cultural y costos iniciales. Estos elementos deben considerarse al diseñar estrategias de implementación en organizaciones de distintos tamaños y sectores, especialmente en aquellas con menor madurez tecnológica.

Tabla 6. Impacto comparado entre DevOps y DevSecOps

CRITERIO DE COMPARACIÓN	DEVOPS	DEVSECOPS
ENFOQUE PRINCIPAL	Velocidad, automatización y entrega continua	Seguridad integrada desde el diseño y durante todo el ciclo de vida
INTEGRACIÓN DE SEGURIDAD	Se incorpora al final del proceso o en fases específicas	Seguridad continua en cada etapa del pipeline (SAST, DAST, SCA, IaC)
RESPONSABILIDAD DE SEGURIDAD	Principalmente equipos de seguridad	Responsabilidad compartida entre desarrollo, operaciones y seguridad
AUTOMATIZACIÓN DE CONTROLES	Limitada a CI/CD y despliegue	Automatización completa de controles de seguridad y cumplimiento
GESTIÓN DE VULNERABILIDADES	Reactiva; se detectan en etapas tardías	Proactiva; detección temprana mediante shift-left security
CULTURA ORGANIZACIONAL	Colaboración entre Dev y Ops	Colaboración multidisciplinaria con enfoque sociotécnico

CUMPLIMIENTO NORMATIVO	Requiere auditorías manuales	Auditorías automatizadas y trazabilidad continua
RESILIENCIA OPERATIVA	Moderada; depende de procesos manuales	Alta; basada en monitoreo continuo y respuesta automatizada
ADECUACIÓN A ENTORNOS MODERNOS	Adecuado para CI/CD básico	Ideal para nube, contenedores, microservicios y Zero Trust
MADUREZ REQUERIDA	Técnica	Técnica + cultural + organizacional

Aplicaciones Prácticas

Las aplicaciones prácticas identificadas en los resultados muestran que DevSecOps aporta beneficios significativos en organizaciones que desarrollan software en entornos complejos, regulados y dinámicos, con impactos tanto técnicos como organizacionales. Una de las aplicaciones más extendidas es la integración de seguridad en los pipelines CI/CD mediante escaneo automático de código fuente (SAST), pruebas dinámicas de seguridad (DAST), análisis de dependencias (SCA), validación de infraestructura como código (IaC) y análisis de contenedores, lo que permite detectar vulnerabilidades en tiempo real y fortalecer la trazabilidad y el cumplimiento normativo (Koneru, 2021). Asimismo, DevSecOps impulsa el desarrollo seguro desde el diseño (Security by Design), incorporando modelado de amenazas, revisión de requisitos de seguridad, definición de políticas de codificación segura y evaluación temprana de riesgos, prácticas que reducen costos y mejoran la calidad del software (Kaithe, 2025).

Otra aplicación clave es la gestión segura de infraestructura mediante IaC, que permite automatizar la creación y configuración de entornos, complementándose con el escaneo de configuraciones inseguras, la validación de políticas, la implementación de entornos inmutables y el control de versiones, garantizando entornos reproducibles y auditables (Microsoft, 2023). El monitoreo continuo y la respuesta a incidentes también constituyen una aplicación esencial, ya que permiten detectar anomalías, identificar patrones de ataque, generar alertas automáticas e integrar sistemas SIEM y SOAR, fortaleciendo la resiliencia organizacional (Guda, 2024).

En sectores regulados como banca, salud y gobierno, DevSecOps facilita el cumplimiento normativo automatizado de marcos como GDPR, HIPAA, PCI-DSS y leyes latinoamericanas de protección de datos, reduciendo la carga administrativa y el riesgo de sanciones (Bass et al., 2023). Además, se observan aplicaciones sectoriales específicas: en banca y fintech, DevSecOps contribuye a la prevención de fraudes, protección de datos sensibles, auditorías automatizadas y validación de APIs; en salud, fortalece la protección de historiales clínicos, la integración segura de sistemas hospitalarios y el cumplimiento de estándares de interoperabilidad; en gobierno, apoya la modernización digital, la gestión segura de servicios ciudadanos y la protección de infraestructura crítica; y en comercio electrónico, mejora la protección frente a ataques automatizados, la validación de pasarelas de pago y el monitoreo de transacciones en tiempo real.

Finalmente, los beneficios organizacionales derivados de estas aplicaciones incluyen reducción de costos operativos, mejora en la calidad del software, mayor velocidad de entrega, incremento en la confianza del cliente y fortalecimiento de la cultura de seguridad, aspectos ampliamente documentados en estudios recientes (Bautista Ramos & Yoo, 2025).

Limitaciones

A pesar de la solidez metodológica y la amplitud del análisis realizado, este estudio presenta una serie de limitaciones que deben ser consideradas al interpretar los resultados. Estas limitaciones se relacionan principalmente con el alcance metodológico, la disponibilidad de literatura reciente, la naturaleza del enfoque DevSecOps y las diferencias contextuales entre organizaciones.

En primer lugar, la investigación se basa exclusivamente en una revisión documental cualitativa, lo que implica que los hallazgos dependen de la calidad, disponibilidad y profundidad de las fuentes consultadas. Aunque se empleó un protocolo sistemático para garantizar rigor y transparencia, la ausencia de datos empíricos provenientes de estudios de caso, experimentos controlados o métricas cuantitativas limita la posibilidad de generalizar los resultados a todos los contextos organizacionales (Kitchenham et al., 2022). La literatura disponible entre 2022 y 2025, si bien abundante, presenta variaciones en metodologías, enfoques y niveles de detalle, lo que puede generar sesgos en la interpretación.

En segundo lugar, la rápida evolución del ecosistema tecnológico constituye una limitación inherente. DevSecOps es un paradigma en constante transformación, impulsado por avances en automatización, inteligencia artificial, infraestructura en la nube y prácticas de seguridad emergentes. Esto implica que algunos conceptos, herramientas o tendencias identificadas en este estudio podrían quedar desactualizados en un corto período de tiempo, especialmente en sectores altamente dinámicos como fintech, telecomunicaciones o comercio electrónico (Bass et al., 2023).

En tercer lugar, la implementación de DevSecOps varía significativamente entre organizaciones debido a diferencias en cultura, recursos, madurez tecnológica y regulaciones sectoriales. Estas variaciones dificultan establecer conclusiones universales sobre la efectividad del modelo. Por ejemplo, organizaciones con baja madurez en seguridad o con estructuras jerárquicas rígidas pueden enfrentar mayores barreras culturales y técnicas que no se reflejan completamente en la literatura analizada (Erich et al., 2017). Asimismo, la disponibilidad de talento especializado en automatización, seguridad y operaciones constituye un factor crítico que no siempre está presente en todos los contextos.

En cuarto lugar, la revisión documental incluyó whitepapers y reportes técnicos de empresas tecnológicas, los cuales, aunque valiosos, pueden contener sesgos comerciales orientados a promover herramientas o servicios específicos. Aunque se aplicaron criterios de evaluación para mitigar este riesgo, es posible que algunos hallazgos reflejen perspectivas influenciadas por intereses corporativos (Microsoft, 2023).

Finalmente, la ausencia de un análisis comparativo basado en métricas cuantitativas limita la capacidad de medir con precisión el impacto de DevSecOps en indicadores como reducción de vulnerabilidades, tiempos de despliegue o costos operativos. Si bien la literatura reporta mejoras significativas, estas cifras provienen de estudios heterogéneos que emplean metodologías distintas, lo que dificulta establecer conclusiones estandarizadas.

En conjunto, estas limitaciones no invalidan los resultados del estudio, pero sí subrayan la necesidad de investigaciones futuras que integren métodos mixtos, estudios de caso longitudinales y análisis empíricos más robustos. Asimismo, se recomienda explorar la adopción de DevSecOps en sectores

con menor madurez tecnológica para comprender mejor los desafíos y oportunidades del modelo en contextos diversos.

CONCLUSIONES

El análisis realizado permite concluir que DevSecOps constituye un enfoque integral y necesario para la ingeniería de software contemporánea, especialmente en contextos caracterizados por arquitecturas distribuidas, ciclos de entrega acelerados y amenazas cibernéticas sofisticadas. Esta conclusión se sustenta directamente en los resultados obtenidos, los cuales muestran que la integración de seguridad, desarrollo y operaciones mejora la calidad del software, fortalece la resiliencia organizacional y reduce de manera significativa los riesgos asociados a vulnerabilidades críticas. Estos hallazgos coinciden con la literatura reciente, que enfatiza la importancia de incorporar controles de seguridad desde las primeras etapas del ciclo de vida mediante prácticas automatizadas y colaborativas (Koneru, 2021).

Los resultados también permiten concluir que la automatización es un pilar fundamental del modelo DevSecOps. La integración de herramientas SAST, DAST, SCA, IaC y sistemas de monitoreo continuo demostró ser determinante para detectar y mitigar vulnerabilidades tempranas, reducir costos operativos y mejorar la eficiencia de los pipelines CI/CD. Esta relación entre automatización y mejora operativa fue especialmente evidente en sectores altamente regulados o expuestos a riesgos cibernéticos, como banca, salud y gobierno, lo cual coincide con estudios previos (Microsoft, 2023).

Otro hallazgo clave del estudio es que la dimensión cultural condiciona la efectividad del enfoque. La evidencia recopilada muestra que la responsabilidad compartida, la comunicación continua y el aprendizaje permanente son factores que influyen directamente en el éxito de la implementación. Sin estos elementos, la adopción técnica se vuelve superficial y pierde impacto. Esta conclusión se alinea con investigaciones que identifican la madurez cultural como uno de los factores más determinantes para la adopción de DevSecOps (Erich et al., 2017).

Asimismo, los resultados permiten afirmar que DevSecOps aporta beneficios significativos en términos de cumplimiento normativo, trazabilidad y auditoría continua. La capacidad de generar evidencia automatizada facilita la adherencia a estándares internacionales y reduce la carga administrativa

asociada a procesos de verificación, lo cual es especialmente relevante en un entorno global donde las regulaciones sobre protección de datos se han vuelto más estrictas (Bass et al., 2023).

No obstante, el estudio también evidencia que la adopción de DevSecOps enfrenta desafíos importantes, entre ellos la falta de talento especializado, los costos iniciales de implementación, la resistencia cultural y la complejidad de integrar herramientas heterogéneas en entornos multinube. Estos desafíos, observados en los resultados sectoriales, subrayan la necesidad de estrategias de adopción progresiva, programas de capacitación continua y marcos de gobernanza que faciliten la transición hacia modelos más seguros y eficientes.

Finalmente, los hallazgos permiten concluir que DevSecOps no es un estado final, sino un proceso de mejora continua que evoluciona al ritmo de las transformaciones tecnológicas globales. Las tendencias emergentes como la inteligencia artificial aplicada a la seguridad, el Zero Trust en pipelines CI/CD, la infraestructura inmutable y la automatización del cumplimiento normativo indican que el modelo seguirá expandiéndose y adaptándose a nuevos desafíos. En este sentido, se recomienda que futuras investigaciones integren estudios empíricos, métricas cuantitativas y análisis sectoriales comparativos que permitan evaluar con mayor precisión la efectividad del enfoque en distintos contextos organizacionales.

En conjunto, los resultados de este estudio permiten afirmar que DevSecOps constituye un marco robusto, adaptable y esencial para el desarrollo de software seguro, ágil y sostenible en entornos altamente dinámicos, y que su adopción representa no solo una mejora técnica, sino una transformación cultural y estratégica que fortalece la capacidad de las organizaciones para enfrentar los desafíos actuales y futuros de la seguridad informática.

REFERENCIAS BIBLIOGRÁFICAS

Bass, L., Weber, I., & Zhu, L. (2023). *DevOps and software architecture in modern systems*. Springer.

Bautista Ramos, R. C., & Yoo, S. G. (2025). *Ciberseguridad en entornos DevOps: Una revisión sistemática de la literatura*. *IEEE Access, PP*(99), 1–1. https://doi.org/10.1109/ACCESS.2025.3582892

Erich, F., Amrit, C., & Daneva, M. (2017). *DevOps literature review*. *Procedia Computer Science, 121*, 180–193. https://doi.org/10.13140/2.1.5125.1201

GitLab. (2023). *Global DevSecOps Report 2023: Security and automation trends*. GitLab Research.

Guda, D. P. (2024). *AI-driven threat detection in DevSecOps pipelines for insurance applications. International Journal of Intelligent Systems and Applications in Engineering, 12*(23s). https://doi.org/10.17762/ijisae.v12i23s.7759

Kaithe, B. K. (2025). *Shift Left Security: A paradigm shift in software development security integration*. *European Journal of Computer Science and Information Technology, 13*(24), 96–102. https://doi.org/10.37745/ejcsit.2013/vol13n2496102

Kim, G., Humble, J., Debois, P., & Willis, J. (2022). *The DevOps handbook: How to create world-class agility, reliability, and security in technology organizations* (2nd ed.). IT Revolution Press.

Kitchenham, B., Budgen, D., & Brereton, P. (2022). *Evidence-based software engineering and systematic reviews* (2nd ed.). CRC Press.

Koneru, N. M. K. (2021). *Integrando la seguridad en las canalizaciones CI/CD: Un enfoque DevSecOps con herramientas SAST, DAST y SCA*. *International Journal of Science and Research Archive, 3*(1), 250–265. https://doi.org/10.30574/ijsra.2021.3.1.0080

Microsoft. (2023). *Security in cloud-native development: A DevSecOps approach*. Microsoft Research.

Mohammed, K. I., Shanmugam, B., & El-Den, J. (2025). *Evolución de DevSecOps y su influencia en la seguridad de aplicaciones: Una revisión sistemática de la literatura*. *Technologies, 13*(12), 548. https://doi.org/10.3390/technologies13120548

Pressman, R., & Maxim, B. (2022). *Software engineering: A practitioner's approach* (10th ed.). McGraw-Hill.

Rahman, M. M., & Williams, L. (2016). *Seguridad del software en DevOps: Sintetizando las percepciones y prácticas de los profesionales*. In *Proceedings of the 38th International Conference on Software Engineering Companion* (pp. 143–152). https://doi.org/10.1145/2896941.2896946

Rajapakse, R. N., Zahedi, M., & Babar, M. A. (2021). *An empirical analysis of practitioners' perspectives on security tool integration into DevOps* (arXiv:2107.02096v3 [cs.CR]). arXiv. https://arxiv.org/abs/2107.02096

Rajapakse, R., Zahedi, M., Babar, M. A., & Shen, H. (2021). *Desafíos y soluciones al adoptar DevSecOps: Una revisión sistemática. Information and Software Technology, 141*, 106700. https://doi.org/10.1016/j.infsof.2021.106700

Red Hat. (2023). *State of DevSecOps in enterprise environments*. Red Hat Insights.

Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). *Zero Trust Architecture* (NIST SP 800-207). https://doi.org/10.6028/NIST.SP.800-207

Sarker, I. H. (2022). *Aprendizaje automático para el análisis inteligente de datos y la automatización en ciberseguridad: Perspectivas actuales y futuras. Annals of Data Science, 10*(3), 1–26. https://doi.org/10.1007/s40745-022-00444-2

Shahin, M., Babar, M. A., & Zhu, L. (2017). *Integración, entrega y despliegue continua: Una revisión sistemática sobre enfoques, herramientas, desafíos y prácticas. IEEE Access, PP*(99). https://doi.org/10.1109/ACCESS.2017.2685629