Ingeniería Investiga

Vol. 8, e1396, año 2026
ISSN – Online: 2708-3039
DOI: https://doi.org/10.47796/ing.v8i00.1396

Artículo de revisión

Integración de DevSecOps en la seguridad de la información para el desarrollo de software: Una revisión sistemática

Integrating DevSecOps into Information Security for Software Development: A Systematic Review

Jhonatan Efrain Monzon Llanos[1]* | Pamela Dominga Alayo Gamboa[2] | Alberto Carlos Mendoza de los Santos[3]

Afiliación:

[1,2,3]Escuela de Ingeniería de Sistemas, Universidad Nacional de Trujillo, La libertad, Perú

Autor de correspondencia: E-mail: *jmonzon@unitru.edu.pe

Información del artículo:

Recibido: 08/09/2025
Aceptado: 09/01/2026
Publicado: 19/02/2026

Resumen

El incremento de amenazas cibernéticas y la necesidad de integrar la seguridad en el ciclo de vida del software han impulsado la adopción de DevSecOps como evolución de DevOps. El presente estudio tuvo como objetivo analizar los beneficios, desafíos y tendencias de la integración de herramientas DevSecOps en la seguridad de la información aplicada al desarrollo de software. Con este propósito, se utilizó la metodología PRISMA y se llevaron a cabo búsquedas en bases de datos académicas como arXiv, Google Académico, SCOPUS e IEEE Xplore. Esto posibilitó la selección y el análisis de 18 artículos significativos publicados entre 2020 y 2025. Los hallazgos muestran que implementar DevSecOps permite identificar vulnerabilidades de manera anticipada, mejora la capacidad de los sistemas para resistir y fomentar una cultura compartida de seguridad entre los grupos de desarrollo, sin embargo, persisten retos vinculados con la compatibilidad de herramientas, el desarrollo organizacional y la necesidad de marcos integradores. Es así que, DevSecOps se constituye como un enfoque clave para optimizar la continuidad y confiabilidad del software. Aunque, su sostenibilidad a largo plazo está condicionada a futuras investigaciones que indaguen sobre modelos predictivos, métricas de evaluación y la utilización de tecnologías como inteligencia artificial y automatización avanzada.

Palabras clave: DevSecOps; seguridad de la información; desarrollo de software; automatización; CI/CD.

Abstract

The increase in cyber threats and the need to integrate security into the software lifecycle have driven the adoption of DevSecOps as an evolution of DevOps. This study aimed to analyze the benefits, challenges, and trends of integrating DevSecOps tools into information security applied to software development. For this purpose, the PRISMA methodology was used and searches were carried out in academic databases such as arXiv, Google Scholar, SCOPUS, and IEEE Xplore. This enabled the selection and analysis of 18 significant articles published between 2020 and 2025. The findings show that implementing DevSecOps allows for the early identification of vulnerabilities, improves the resilience of systems, and fosters a shared security culture among development groups. However, challenges related to tool compatibility, organizational development, and the need for integrative frameworks persist. Thus, DevSecOps is emerging as a key approach to optimizing software continuity and reliability. However, its long-term sustainability depends on future research investigating predictive models, evaluation metrics, and the use of technologies such as artificial intelligence and advanced automation.

Keywords: DevSecOps; information security; software development; automation; CI/CD.

1. Introducción

La seguridad de la información constituye un pilar fundamental en el ciclo de vida del desarrollo de software (SDLC), especialmente en entornos que enfrentan el incremento de amenazas cibernéticas. Diversos estudios muestran que la integración temprana de prácticas de seguridad permite reducir riesgos y costos asociados a vulnerabilidades (Aljohani et al., 2023; David et al., 2024). De igual forma, Elder et al. (2021) resaltan la importancia de alinear la seguridad con los objetivos empresariales, mientras que investigaciones recientes refuerzan la importancia de metodologías más estructuradas y rigurosas revisiones para el fortalecimiento de la protección del software (Saeed et al., 2025; Kudriavtseva & Gadyatskaya, 2023). De esta manera, DevOps ha evolucionado hacia DevSecOps, incorporando la seguridad de forma continua y automatizada de la seguridad dentro de los procesos de integración y entrega continua (CI/CD).

El análisis estático y dinámico se evidencia como un pilar para detectar vulnerabilidades tempranas. Los trabajos de Christakis et al. (2022) y Jerónimo et al. (2024) respaldan los enfoques híbridos SAST-DAST por su mayor precisión, además Stanciu y Ciocârlie (2023) destacan que automatizar las revisiones de código mejora la eficiencia y la calidad. Aun así, integrar controles de seguridad en metodologías ágiles sigue siendo retador por las dependencias y la compatibilidad entre herramientas; esto exige marcos más flexibles y automatización avanzada (Sinan et al., 2025). Finalmente, adoptar DevSecOps implica un cambio cultural de manera que, la seguridad debe ser compartida por todos los equipos (Sermpezis et al., 2024).

Por otro lado, las tendencias culturales evidencian un progreso hacia enfoques inteligentes que amplían las prácticas convencionales. Según Ahsan y Anwer (2024), la incorporación de metaheurísticas, machine learning y aprendizaje por refuerzo optimizan el testing automatizado, favoreciendo la identificación de patrones avanzados y vulnerabilidades novedosas. De un modo similar, Riaz et al. (2025) afirman que el avance del DevSecOps no solo mejora la calidad del software, sino que también hace que los sistemas sean más resistentes ante ataques en entornos críticos.

A pesar del creciente número de estudios sobre DevSecOps, la evidencia disponible se encuentra dispersa en investigaciones centradas en herramientas específicas, entornos particulares o aplicaciones sectoriales, lo que dificulta la identificación de tendencias integrales sobre su impacto real en la seguridad del desarrollo de software. Asimismo, existe limitada sistematización de los beneficios, desafíos organizacionales y avances tecnológicos emergentes asociados a su implementación, especialmente en contextos de adopción progresiva dentro de organizaciones de distintos niveles de madurez tecnológica.

De En este contexto, el objetivo del presente estudio es realizar una revisión sistemática de la literatura que permita identificar y analizar las principales tendencias, beneficios, limitaciones y oportunidades futuras relacionadas con la integración de herramientas DevSecOps en la seguridad de la información aplicada al desarrollo de software, proporcionando una visión integradora que contribuya a orientar futuras investigaciones y estrategias de implementación en entornos organizacionales

2. Metodología

El estudio empleó la metodología PRISMA (Page et al., 2021), la que ofrece un marco riguroso para estructurar las etapas de búsqueda, evaluación y análisis de literatura científica, asegurando que el proceso sea claro y replicable. Su aplicación permitió recopilar información actual y relevante relacionada con la integración de herramientas DevSecOps en la seguridad de la información para el desarrollo de software.

2.1. Estrategia de búsqueda

La búsqueda sistemática se realizó durante julio y agosto de 2025 en cuatro bases de datos académicas: Google Académico, SCOPUS, arXiv e IEEE Xplore. Se emplearon palabras clave específicas relacionadas con DevSecOps y

seguridad de la información, combinadas mediante operadores booleanos para optimizar la precisión de los resultados. La Tabla 1 presenta el detalle de las bases de datos consultadas, términos de búsqueda utilizados y el número de documentos identificados y seleccionados.

Tabla 1

Términos de búsqueda en base de datos

Base De Datos	Términos De Búsqueda	Resultados	Seleccionados
Google Académico	("software development" OR “SDLC”) AND ("security automation" OR "automated security tools" OR “SAST” OR “DAST” OR “IAST”) AND ("software security" OR "information security")	135	6
Scopus	TITLE-ABS-KEY ((“software development" OR SDLC) AND (“security automation" OR "automated security tools" OR “SAST” OR “DAST” OR “IAST”) AND (“software security" OR "information security”)) AND PUBYEAR > 2019 AND PUBYEAR < 2026 AND (LIMIT-TO (LANGUAGE, "English”) OR LIMIT-TO (LANGUAGE, "Spanish”))	15	4
arXiv	(“(“software development" OR “SDLC”) AND ("security automation" OR "automated security tools" OR “SAST” OR “DAST” OR “IAST”) AND "software security" OR "information security")	19	2
IEEE Xplore	((("software development" OR “SDLC”) AND ("security automation" OR "automated security tools" OR “SAST” OR “DAST” OR “IAST”) AND ("software security" OR "information security")))	21	6

2.2. Criterios de inclusión y exclusión

Para garantizar la calidad y pertinencia de los estudios incluidos en esta revisión sistemática, se definieron criterios de inclusión y exclusión específicos que se muestran en la tabla 2.

Tabla 2

Criterios de inclusión, exclusión y calidad

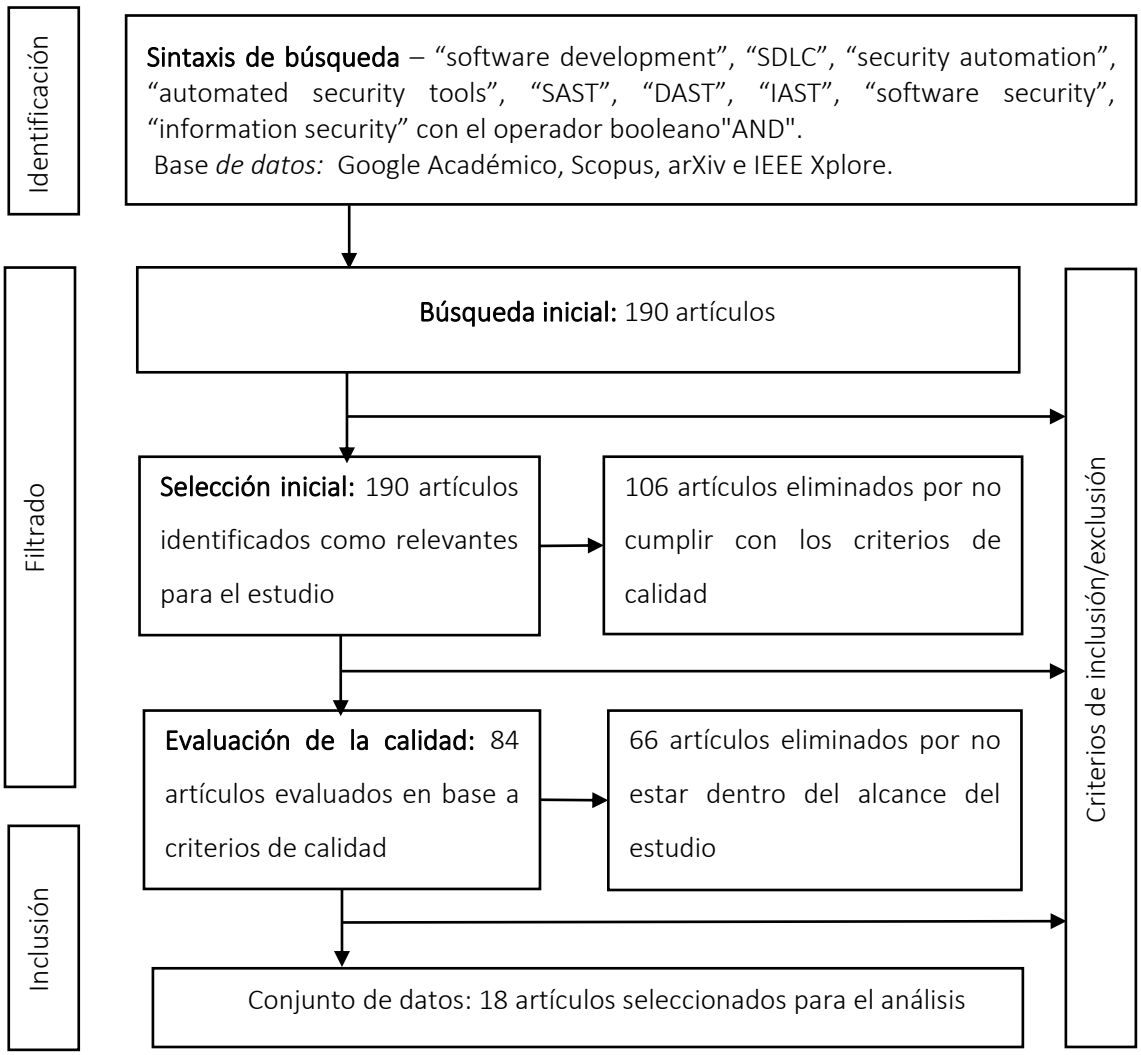
Criterio	Descripción
Inclusión	Publicaciones entre los años 2020 y 2025.
	Estudios disponibles en texto completo en español o inglés.
	Documentos accesibles en línea desde bases de datos seleccionadas.
Exclusión	Documentos sin relación directa con las herramientas aplicadas a la seguridad de la información.
	Publicaciones anteriores a 2020.
	Contenido duplicado o sin respaldo académico.
	Ensayos de opinión, notas breves o artículos no revisados por pares.
	Áreas de medicina.
Calidad	Relevancia temática.
	Claridad metodológica.
	Fundamentación teórica y referencias confiables.
	Accesibilidad y publicación en fuentes académicas.

2.3. Proceso de recolección de la información

Se utilizó el diagrama de flujo de la metodología PRISMA, como se ilustra en la Figura 1, con el objetivo de asegurar la relevancia de los documentos identificados. De este modo, tras ejecutar la estrategia de búsqueda y aplicar los criterios de inclusión y exclusión detallados, se seleccionaron los artículos pertinentes a la línea de investigación.

Figura 1

Secuencia metodológica mediante el diagrama de flujo PRISMA



2.4. Evaluación del riesgo de sesgo de los estudios seleccionados

Con la intención de mejorar la validez interna de la revisión sistemática que se ha realizado, se ha llevado a cabo una evaluación cualitativa de la propensión al sesgo exhibida en los estudios incluidos, teniendo en cuenta las especificidades que tienen lugar en el dominio de la investigación en ingeniería de software y sobre seguridad de la información. Dada la ausencia de un instrumento estandarizado que sea ampliamente utilizado, se ha adoptado un enfoque en base a criterios que han sido utilizados por trabajos anteriores en lo que respecta a revisiones sistemáticas (Martelleur & Hamza, 2022; Kudriavtseva & Gadyatskaya, 2023).

Entre otros, se ha analizado el posible sesgo de publicación, y se ha observado que en la mayoría de los estudios seleccionados se presentan resultados provenientes de fuentes

académicas consolidadas y establecidas, como son IEEE Xplore, Scopus y arXiv. Sin embargo, también se indica que existe cierta tendencia hacia la publicación de resultados positivos en la adopción de DevSecOps, mientras que los trabajos que evidencian fracasos o resultados negativos son menos frecuentes, lo que podría suponer una limitación en la correcta representación del fenómeno.

En segundo lugar, se realizó el sesgo de metodología, ya que una buena parte de los artículos tratados se basa en revisiones sistemáticas, estudio de caso o propuesta de marcos conceptuales. Si bien tales tipos de artículo aportan evidencia importante, las evidencias presentadas por algunos de los trabajos analizados son de escasa validez empírica, al encontrarse en entornos controlados que pueden comprometer la validez externa del hallazgo en contextos industriales concretos.

Por otra parte, se descubrió que el sesgo de contexto está presente, ya que algunos estudios abarcan aplicaciones web, entornos web en la cloud, o sectores concretos como la parte financiera, pero consideraremos, en menor medida, la implementación de DevSecOps en organizaciones de menor tamaño o en sectores industriales diferentes en los que su uso pueda ser diferente. Así la concentración del tema en los trabajos revisados puede afectar la percepción del impacto real de las prácticas DevSecOps en entornos distintos.

Finalmente, se detectó el sesgo de contexto asociado a herramientas y tecnologías, ya que algunos estudios evalúan una solución específica (p. ej., un SAST o DAST concreto) y favorecen un tipo concreto de enfoque técnico frente a otros. A pesar de estas limitaciones, la consideración de criterios de inclusión estrictos, la incorporación de las diversas fuentes académicas y la homogeneidad de cómo los hallazgos se presentan y esbozan entre estudios analizados contribuyen a mitigar el impacto global del riesgo de sesgo en los resultados de la revisión actual.

3. Resultados

A continuación, en la Tabla 2 se presenta el análisis de los 18 artículos seleccionados, destacando los principales resultados relacionados con la integración de herramientas y prácticas de seguridad de la información, lo cual permite identificar tendencias, beneficios y limitaciones encontradas en la literatura.

Tabla 2

Resultados de los documentos analizados

N°	Autores y año	Resultados
1	Czekster (2024)	Evidencia que la integración de Secure DevOps con gestión de riesgos (Risk Assessment y Threat Modelling) logra reforzar la seguridad durante todo el desarrollo del software, previendo amenazas. Se fomenta la importancia de incorporar seguridad desde las primeras fases en CI/CD y metodologías ágiles, junto con capacitación y cultura organizacional, para elevar la robustez y calidad del producto.
2	Nikolov y Aleksieva-Petrova (2023)	Se evalúa que DevSecOps profundiza la protección del software con automatización y detección de posibles vulnerabilidades desde una etapa temprana, formando una cultura ágil colaborativa. Además, se enfatiza la importancia de las tecnologías de nube y contenedores, así como los desafíos regulatorios, confirmando a DevSecOps como un campo emergente de ciberseguridad.

Tabla 2 (Continuación/1)

3	Lu et al. (2024)	Se introduce una técnica de detección automática, basada en la caracterización de múltiples versiones mejora la seguridad y confiabilidad del software. El enfoque integra machine learning y minería de datos para la detección de amenazas de forma autónoma. Los resultados experimentales evidencian un aumento en la eficiencia y precisión frente a métodos tradicionales, fortaleciendo la protección de los sistemas.
4	Piskachev et al. (2022)	Las herramientas de análisis estático, aunque cada vez más reconocidas, tienen un uso limitado; los desarrolladores prefieren opciones gratuitas y enfrentan barreras culturales y de procesos para su integración. Los chequeos automáticos de seguridad rara vez se realizan en cada entrega. Estos resultados destacan la necesidad de fortalecer la madurez organizacional para integrar efectivamente herramientas DevSecOps y mejorar la seguridad en el desarrollo de software.
5	Islam et al. (2024)	El estudio indica que las herramientas tradicionales corrigen solo el 36.2% del código vulnerable. Ante dicha limitación se diseñó T5-RCGCN, una herramienta basada en la integración de language models y graph networks para detectar la causa de las vulnerabilidades. Tal enfoque obtuvo una mejora del 28.9% en la corrección de código, poniendo en evidencia cómo la IA mejora la identificación y corrección temprana en DevSecOps.
6	Wadhams et al. (2024)	Se utilizaron herramientas SAST desde etapas tempranas del desarrollo para proteger información sensible. Se diseñó un proceso automatizado que integra resultados de análisis estático con plataformas de gestión de incidencias, mejorando la detección y control de vulnerabilidades. La implementación con SonarQube en GitLab generó beneficios como retroalimentación rápida y cultura de seguridad, pese a falsos positivos y curva de aprendizaje
7	Chen et al. (2022)	Los datos del estudio indican que es fundamental implementar medidas de seguridad a lo largo de todo el ciclo de vida del software. Para ello se han aplicado y valorado técnicas como el análisis de composición de software (SCA), las pruebas interactivas de seguridad de aplicaciones (IAST), así como las pruebas estáticas (SAST) y dinámicas (DAST). Los resultados destacan que su puesta en marcha temprana y continua permite descubrir vulnerabilidades y mejorar la seguridad del software.
8	Ami et al. (2024)	Pese a que las herramientas SAST se emplean con frecuencia para identificar vulnerabilidades, la investigación subraya que los expertos afrontan retos vinculados a sus restricciones, facilidad de uso y alineación con lo que esperan los equipos de desarrollo. Asimismo, se detectan falencias en los modelos existentes, lo que hace necesaria la innovación hacia soluciones más eficaces y con un nivel de adopción superior.
9	Kushwaha et al. (2024)	Según el estudio, al incorporar métodos de protección en los procedimientos DevSecOps, por ejemplo, SAST y DAST (Codacy, OWASP ZAP y GitHub Advanced Security Scanning), sobre Azure DevOps, se favorece la identificación y resolución de vulnerabilidades en fases iniciales. Subrayan que, gracias a la utilización de pruebas automatizadas y modelado combinado de amenazas, se mejora la seguridad en las aplicaciones web del sector financiero mientras se mantiene la agilidad en la entrega ininterrumpida de software.

Tabla 2 (Continuación/2)

10	Martelleur y Hamza (2022)	Mediante una revisión sistemática de literatura se identificaron trece categorías de herramientas de seguridad para DevSecOps, organizadas en torna a siete fases del desarrollo y cinco prácticas clave de seguridad. El estudio también identifica doce desafíos y dieciséis recomendaciones que contribuyen a enfrentar los retos al seleccionar y aplicar herramientas de seguridad en entornos de automatización y entrega continua.
11	Nguyen (2020)	Mediante una revisión multivocal, se analizan más de 300 herramientas de automatización de seguridad y privacidad, clasificadas en 18 categorías y 119 actividades. El estudio evidencia que la investigación actual se centra en actividades de seguridad ligadas al desarrollo de software, destacando la escalabilidad y trazabilidad como factores clave en la adopción de la automatización para mejorar la seguridad y el cumplimiento normativo.
12	Meliala et al. (2024)	El estudio combina un análisis de literatura y exploración de mercado para proponer un marco capaz de integrar pruebas de seguridad dentro de los flujos de trabajo CI/CD. Describen los tipos de prueba que resultan relevantes en las distintas fases del pipeline y los factores que influyen en la selección e integración de herramientas, ofreciendo recomendaciones para profesionales que buscar fortalecer la seguridad en los procesos de entrega de software.
13	De Jesús Dominguez-García et al. (2023)	La revisión sistemática de literatura permitió clasificar 18 técnicas de prueba, 88 herramientas y cuatro modelos de desarrollo seguro centrados en aplicaciones web, tomando como base el modelo de amenazas STRIDE. Los hallazgos muestran una evolución que va desde pruebas de penetración tradicionales hasta soluciones basadas en inteligencia artificial, sirviendo como base práctica para la práctica profesional y la investigación académica.
14	Valdés-Rodríguez et al. (2023)	El análisis revela deficiencias en la integración temprana de seguridad en las etapas de conceptualización, desarrollo y verificación de métodos ágiles. Sin embargo, hay una tendencia positiva hacia su adopción. Se propone una metodología basada en estos hallazgos para equipos inexpertos en marcos ágiles, sentando bases para investigaciones futuras en desarrollos móviles y tecnologías innovadoras.
15	Dasanayake et al. (2024)	La revisión sistemática hace una búsqueda sobre cómo poner en práctica DevSecOps en aplicaciones de negociación financiera con respecto a codificar de manera segura, pruebas sin ayuda, monitoreo continuo y respuesta ante problemas. Se encuentran dificultades con el cumplimiento de las normas y también con la protección de datos importantes junto a restricciones en las amenazas más nuevas . El estudio indica que la integración de DevSecOps es clave para mejorar la seguridad de sistemas de negociación en un ámbito cibernético muy hostil.
16	Wang y Yan (2021)	Con el apoyo de las capacidades de BurpSuite, se proponen técnicas de pruebas de seguridad automatizadas, con el fin de maximizar la eficacia de testing de software. Los resultados obtenidos muestran que, con el uso flexible de dicha herramienta se logra un incremento tanto seguridad como eficiencia en la localización de vulnerabilidades, ayudando a incrementar el nivel de seguridad de los sistemas. Las ventajas se resumen en: Facilidad de uso, eficiencia y flexibilidad.

Tabla 2 (Continuación/3)

17	Santos et al. (2021)	El estudio compara la revisión manual de código con herramientas SAST tradicionales y con versiones de SAST mejoradas mediante IA y aprendizaje automático. Demuestran que con la integración de ML/IA se identifica de manera rápida y precisa las vulnerabilidades, por lo cual se recomienda su uso en la industria para potenciar la seguridad durante el desarrollo del software.
18	Bryhynets et al. (2025)	El estudio se centra en el uso de SAST con la herramienta Snyk Code e introduce el modelo General Application Vulnerability Rate (GAVR) fundamentado en CVSS 3.1. El caso analizado muestra la detección de fallos como XSS así como problemas de validación de certificados. El GAVR hace posible un enfoque más dinámico de evaluación. Los resultados subrayan que al integrar SAST junto con metodologías automatizadas y cuantificación estructurada de riesgos disminuye riesgos además de aumentar la eficiencia en la remediación.

Síntesis temática de los hallazgos

Del análisis comparativo de los 18 estudios seleccionados emergen cuatro categorías principales:

Integración temprana de herramientas automatizadas

La mayoría de los estudios coinciden en que la incorporación de herramientas SAST, DAST, IAST y SCA en fases iniciales del SDLC permite reducir vulnerabilidades de forma anticipada. Se observa especial énfasis en la automatización dentro de pipelines CI/CD como mecanismo de retroalimentación temprana.

Impacto de la inteligencia artificial en la detección de vulnerabilidades

Varios estudios recientes incorporan técnicas de machine learning y modelos híbridos que mejoran la precisión del análisis estático y dinámico, reduciendo falsos positivos y fortaleciendo la priorización de riesgos críticos.

Desafíos organizacionales y culturales

Se identifican barreras relacionadas con madurez organizacional, resistencia al cambio, limitaciones presupuestarias y falta de personal especializado, especialmente en pequeñas y medianas empresas.

Necesidad de marcos integradores y métricas cuantitativas

Diversos estudios proponen modelos conceptuales y métricas como GAVR, evidenciando la necesidad de enfoques estructurados que permitan evaluar cuantitativamente la efectividad de DevSecOps.

4. Discusión

La integración de la seguridad desde fases tempranas del desarrollo de software ha demostrado ser un factor decisivo para reducir vulnerabilidades y fortalecer la resiliencia de los sistemas. Estudios recientes evidencian que la incorporación de prácticas como la gestión de riesgos y el modelado de amenazas en entornos DevSecOps no solo anticipa posibles ataques, sino que también mejora la alineación de los objetivos de seguridad con los de negocio (Czekster, 2024). A ello, se añade la necesidad de transformaciones culturales como también de gestión, puesto que el adoptar DevSecOps requiere equipos colaborativos y ágiles, capaces de asumir la seguridad y responsabilidad compartida a lo largo de todo el ciclo de desarrollo (Nikolov & Aleksieva-Petrova, 2023; Valdés-Rodríguez et al., 2023).

Los recientes avances tecnológicos evidencian que con el empleo de nuevas técnicas y la automatización se logra un cambio importante en cuanto a la manera en que se identifican y gestionan los riesgos. BurpSuite, en este contexto, ayuda a mejorar la exactitud de las

pruebas de seguridad automatizadas (Wang & Yan, 2021), mientras que la aplicación de machine learning en el análisis estático del código ha permitido que la detección de vulnerabilidades mejore, dejando atrás las limitaciones de los métodos manuales (Santos et al., 2021; Islam et al., 2024). Asimismo, al incorporar pruebas de seguridad en pipelines CI/CD se obtiene una retroalimentación temprana, lo que fomenta una postura de seguridad más sólida en ambientes cambiantes de entrega continua (Wadhams et al., 2024; Meliala et al., 2024).

Además, las metodologías de evaluación cuantitativa y los marcos sistemáticos que ofrecen nuevos enfoques se traducen en la adopción de decisiones estratégicas. Modelos como el GAVR, que incluyen probabilidades de explotación para priorizar riesgos críticos (Bryhynets et al., 2025), y revisiones sistemáticas, que han hecho posible la identificación de categorías de herramientas y prácticas eficaces para diferentes contextos, así como los beneficios y las limitaciones de su uso (Martelleur & Hamza, 2022; De Jesús Domínguez-García et al., 2023; Dasanayake et al., 2024). Estos descubrimientos apoyan la noción de que el futuro de DevSecOps no solo se basa en la innovación tecnológica, sino también en marcos integradores que unan metodologías de evaluación estructurada, automatización y la capacidad organizacional para mantener prácticas seguras a largo plazo.

En este sentido, es interesante poner el foco en el contraste que DevSecOps tiene en relación con otras metodologías de desarrollo de software que aplican prácticas de seguridad, con el fin de exponer sus principales diferencias, limitaciones y beneficios en torno a su capacidad de integración, automatización y adaptarse a un entorno de desarrollos de software moderno

La adopción de Arquitecturas Zero Trust (ZTA) aporta beneficios en seguridad y eficiencia operativa, aunque enfrenta limitaciones técnicas y organizacionales. Estos resultados confirman patrones observados en la literatura reciente, pero también evidencian vacíos que requieren nuevas líneas de investigación.

4.1. Comparación DevSecOps y otras metodologías de desarrollo de software con enfoque en seguridad

La forma en la que se incorporó la seguridad dentro de la práctica del desarrollo de software ha sido tradicionalmente atendida por cualquiera de las metodologías históricas de desarrollo de software, cada una de ellas con un distinto nivel de práctica, de integración y de mejora de la eficiencia. En este sentido, DevSecOps se aparta de las metodologías más tradicionales o tradicionales-agiles por presentar un modelo de integración continua, pero sobre todo automatizada que lidia desde el inicio y hasta la obtención de entregas del software, en contraste con las metodologías o modelos que prevén la incorporación de prácticas de control de seguridad de forma lineal o tardía.

El enfoque Secure Software Development Life Cycle (SSDLC) que incorpora lo que sería la seguridad en el ciclo de desarrollo de análisis de requisitos, diseño seguro y las pruebas de seguridad como último proceso de desarrollo de software. Hoy por hoy, este modelo no es sino el modelo que permite elevar la posibilidad de la identificación de riesgos, lo que pasó a ser en el modelo anterior, un modelo más útil que el anterior en relación a las metodologías de desarrollo del software tradicional, pero que ha acabado siendo más rígido y un tanto menos adaptado a las características que suelen tener normalmente los entornos de desarrollo ágil y entrega continuas. Esto ofrece la posibilidad de tener métodos de trabajo con poco margen para la adaptación en contexto más dinámicos (Kudriavtseva & Gadyatskaya, 2023).

Por otro lado, el enfoque tradicional de DevOps enfatiza la velocidad y la automatización del despliegue, dejando a la seguridad en las fases finales de cada proceso. Esto da como resultado la aparición de brechas que pueden ser explotadas de la misma manera que la eficiencia del ciclo empieza a incrementarse, lo que, a su vez, incrementa el costo de rectificación y el riesgo operativo. En contraste, el movimiento de DevSecOps aparece precisamente para subsanar estas limitaciones al introducir controles de seguridad automatizados que van desde SAST, DAST, IAST, hasta análisis de composición de

software, en los pipelines CI/CD, haciendo posible la detección de vulnerabilidades en una forma temprana y continua de forma intrínseca al ciclo de vida del despliegue. (Chen et al., 2022; Wadhams et al., 2024).

De igual manera, las formas de trabajo de la seguridad de la información integradas en metodologías ágiles, llamadas Agile Security, introducen elementos de concienciación de seguridad y revisiones periódicas de seguridad durante cada uno de los sprints. Sin embargo, su eficiencia depende fuertemente de la experiencia del equipo de desarrollo y de los procesos manuales debido a que son limitados en cuanto a la escalabilidad y la consistencia de los controles a la vez que DevSecOps aprovecha el potencial de los enfoques previos para ofrecer prácticas automatizadas y tener menores dependencias manuales y mejorar la trazabilidad de las acciones de seguridad (Martelleur & Hamza, 2022).

En resumen, aunque las metodologías tradicionales o bien ágiles con seguridad incorporada permiten ciertas mejoras, DevSecOps supone un enfoque más holístico del mismo debido a que une automatización, integración continua y responsabilidad compartida. Esta confluencia beneficiada no sólo el descubrimiento y mitigación de vulnerabilidades, sino que también permite alinear la seguridad con las metas de negocio, algo que es primordial en el desarrollo moderno del software por la alta exigencia de cambio, así como la exposición constante a las amenazas de la seguridad cibernética.

4.2. Impacto de la inteligencia artificial y la automatización avanzada en la evolución de DevSecOps

El desarrollo reciente de DevSecOps está asociado a la progresión de la inteligencia artificial (IA) y la automatización avanzada que permiten al software identificar y gestionar vulnerabilidades encontradas a partir de la forma en la que se escribe el software. Pronto, las tecnologías ajustarían las limitaciones de enfoques de DevSecOps tradicionales, especialmente en las vinculadas con el ciclo tardío en detectar fallos, o en la dependencia de procesos manuales.

Varios estudios han mostrado que la imbuición de machine learning en herramientas de análisis (estáticas y dinámicas) mejora la precisión en la detección de vulnerabilidades, disminuyendo falsos positivos/falsos negativos sin comprometer la velocidad de los ciclos de entrega continua (Santos et al., 2021; Islam et al., 2024). La automatización definida durante las pruebas de seguridad en pipelines CI/CD también permite una mejor gestión del riesgo y la priorización de vulnerabilidades críticas encontradas pronto (Wadhams et al., 2024; Bryhynets et al., 2025).

En el ámbito de las organizaciones, la inteligencia artificial aplicada a DevSecOps permite que se potencie la toma de decisiones gracias al uso de análisis predictivo y sistemas de recomendación y, al mismo tiempo, potenciar la cultura de la seguridad para convertirse en una cultura de la seguridad más proactiva. Sin embargo, su adopción plantea retos relacionados con la calidad de los datos, con la necesidad de garantizar la transparencia de los modelos y con la dependencia tecnológica, los cuales hay que tenerlos en cuenta para garantizar una implementación sostenible (Ahsan & Anwer, 2024; Sinan et al., 2025).

En resumen, la inteligencia artificial y la automatización avanzada son motores claves que aceleran el proceso de evolución inmediato de DevSecOps, para facilitar la eficiencia técnica y la capacidad preventiva de los procesos de seguridad, siempre que dicha integración venga complementada con metodologías formando parte de un nivel de madurez organizacional suficiente.

4.3. Obstáculos organizacionales y culturales en la adopción de DevSecOps

No obstante, los beneficios técnicos de la DevSecOps, su práctica se halla aún expuesta a importantes obstáculos sociales y organizativos, sobre todo en empresas de menor tamaño. En este sentido, varios estudios indican que, la resistencia al cambio, la carencia de personal cualificado y la escasa madurez de los procesos de seguridad se hallan entre los obstáculos más

persistentes para la integración de DevSecOps (Nikolov & Aleksieva-Petrova, 2023; Piskachev et al., 2022).

En el caso de las pequeñas y medianas empresas el problema se agrava por las restricciones de recursos financieros e humanos, lo que impide la utilización de herramientas de última generación y la automatización completa de los pipelines CI/CD. Y es que la ausencia de la figura de un profesional de la seguridad lleva a que dicha función se asuma de manera informal por los propios grupos de desarrollo, lo que da lugar a la acumulación de riesgos por causas relacionadas con una mala práctica (omisiones, configuraciones incorrectas y similares) de naturaleza tardía.

Sin embargo, la literatura establece que los problemas mencionados pueden ser solucionados acudiendo a estrategias progresivas, como la adopción progresiva de herramientas automatizadas, el uso de soluciones basadas en código abierto y la formación continua del personal. Asimismo, el fomento de la cultura de la responsabilidad compartida y la progresiva incorporación de controles de seguridad en las primeras etapas del desarrollo permiten a las pequeñas organizaciones mejorar su postura de seguridad y sensibilidad a costa de la agilidad (Valdés-Rodríguez et al., 2023; Sinan et al., 2025).

En resumen, la superación de los daños organizacionales y culturales representa un elemento crítico para la implantación de DevSecOps, siendo que, en las pymes, el uso de la automatización progresiva, la consciencia del equipo y la adaptación de las buenas prácticas son clave para una implantación sostenible.

5. Conclusiones

Con la integración de DevSecOps en el ciclo de vida del desarrollo de software se logra un cambio notable en la forma en que se maneja la seguridad de los datos. La incorporación temprana de controles automatizados, y una mezcla de métodos de análisis dinámico y estático, ha hecho posible la identificación de vulnerabilidades en fases tempranas, lo cual ha disminuido los riesgos y costos relacionados. Demostrando que la seguridad no debe ser considerada como una etapa separada, sino como un elemento transversal y constante en las metodologías ágiles y los pipelines de integración y entrega continua (CI/CD).

También, el uso de métodos basados en aprendizaje automático, modelado de amenazas y análisis automatizado mejora la habilidad para responder a ciberataques en entornos constantemente cambiantes y críticos. Estos avances no solo aumentan la eficiencia técnica, sino que también promueven una cultura en la organización donde la seguridad se considera una responsabilidad que todos deben asumir juntos. Colaborar con los objetivos comerciales, garantizar un control total desde el principio hasta el final y tener procesos estandarizados ayudan a crear un marco estratégico más sólido, lo que mejora la capacidad del software demostrando resiliencia y fiabilidad en situaciones donde hay mucha demanda.

Los resultados analizados abordan la necesidad de seguir profundizando en líneas de investigación futuras, como la investigación de nuevas métricas evaluativas que posibiliten priorizar riesgos críticos, el desarrollo de modelos predictivos que incorporen elementos humanos y técnicos para prever vulnerabilidades, además del uso de tecnologías emergentes como la inteligencia artificial avanzada, la automatización inteligente y las metaheurísticas. Con el apoyo de estas iniciativas el enfoque DevSecOps será más sostenible y aplicable en organizaciones de sectores y tamaños variados, para así mejorar la seguridad del software en contextos cada vez más complejos.

Asimismo, el análisis comparativo incluido en la revisión sistemática demuestra que DevSecOps ofrece ciertas ventajas relevantes respecto de las estrategias tradicionales y ágiles de integración parcial de seguridad ya que promueve la práctica de la automatización e integración continua, lo cual mejora la deteicción y resolución prematura de vulnerabilidades. Sin embargo, la adecuada adopción de la práctica se ve dificultada por dificultades organizacionales y culturales en el caso de las pequeñas y medianas empresas donde los recursos son limitados y, en consecuencia, la especialización en seguridad es escasa. En ese sentido, la posibilidad de integrar

herramientas de forma progresiva, y la mejora de las capacidades a través del entrenamiento de seguridad y las tecnologías emergentes como el uso de inteligencia artificial y la automatización avanzada parecen ser, a corto y medio plazo, claves para generar la adopción de DevSecOps de forma sostenible y escalable

Contribución de los autores

J. E. Monzon: Conceptualización, investigación, metodología, administración del proyecto, recursos, visualización y redacción del borrador original P. D. Alayo: Conceptualización, investigación, metodología, administración del proyecto, recursos y redacción del borrador original A. C. Mendoza: Supervisión, validación, redacción, revisión y edición.

Conflictos de interés

Los autores declaran no tener ningún conflicto de interés relacionado con esta publicación.

6. Referencias bibliográficas

Ahsan, F., & Anwer, F. (2024). A systematic literature review on software security testing using metaheuristics. Automated Software Engineering, 31(44), 1–36. https://doi.org/10.1007/s10515-024-00433-0

Aljohani, M. A., & Alqahtani, S. S. (2023). A unified framework for automating software security analysis in DevSecOps. 2023 International Conference on Smart Computing and Application (ICSCA), 1–6. Hail, Saudi Arabia. https://doi.org/10.1109/ICSCA57840.2023.10087568

Ami, A. S., Moran, K., Poshyvanyk, D., & Nadkarni, A. (2024). "False negative – that one is going to kill you": Understanding industry perspectives of static analysis–based security testing. 2024 IEEE Symposium on Security and Privacy (SP), 3979–3997. San Francisco, CA, United States. https://doi.org/10.1109/SP54263.2024.00019

Bryhynets, A., Haidur, H., Gakhov, S., & Marchenko, V. (2025). Quantitative web application vulnerability assessment using SAST methodology. International Journal of Computing, 24(1). https://doi.org/10.47839/ijc.24.1.3888

Chen, S.-J., Pan, Y.-C., Ma, Y.-W., & Chiang, C.-M. (2022). The impact of the practical security test during the software development lifecycle. 2022 24th International Conference on Advanced Communication Technology (ICACT), 313–316. PyeongChang Kwangwoon_Do, Korea, Republic of. https://doi.org/10.23919/ICACT53585.2022.9728868

Christakis, M., Cottenier, T., Filieri, A., Luo, L., Mansur, M. N., Müller, P., Qadeer, S., Schafhalter, P., Schulte, W., & Tillmann, N. (2022). Input splitting for cloud-based static application security testing platforms. In Proceedings of the 30th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 1367–1378). ACM. https://doi.org/10.1145/3540250.3558944

Czekster, R. M. (2024). Continuous risk assessment in secure DevOps. Aston University. https://doi.org/10.48550/arXiv.2409.03405

Dasanayake, S. D. L. V., Senanayake, J., & Wijayanayake, W. M. J. I. (2024). Devsecops for Continuous Security in Trading Software Application Development: a Systematic Literature Review. Journal of Desk Research Review and Analysis, 2(2), 215-232. https://jdrra.sljol.info/articles/10.4038/jdrra.v2i2.52

David, P., Kushwaha, M. K., & Suseela, G. (2024). DevSecOps in finance: Strengthening the security model of applications. 2024 4th International Conference on Data Engineering and Communication Systems (ICDECS), 1–6. Bangalore, India.

https://doi.org/10.1109/ICDECS59733.2023.10502917

De Jesus Dominguez-García, A., Limón, X., Ocharan-Hernandez, J. O., & Perez-Arriaga, J. C. (2023). Pruebas de seguridad para aplicaciones web: Una revisión sistemática de la literatura. XI Conferencia Internacional de Investigación e Innovación en Ingeniería de Software (CONISOFT), 82–91. León, Guanajuato, México. https://doi.org/10.1109/CONISOFT58849.2023.00020

Elder, S. (2021). Vulnerability detection is just the beginning. arXiv. https://doi.org/10.48550/arXiv.2103.05160

Islam, N. T., Bethany, M., Manuel, D., Jadliwala, M., & Najafirad, P. (2024). Unintentional security flaws in code: Automated defense via root cause analysis. University of Texas at San Antonio. https://doi.org/10.48550/arXiv.2409.00199

Jerónimo, A. H., Moreno, P. M., Camacho, J. A. V., & Vega, G. C. (2024). Techniques of SAST tools in the early stages of secure software development: A systematic literature review. In 2024 IEEE International Conference on Engineering Veracruz (ICEV) (pp. 1–8). IEEE. https://doi.org/10.1109/ICEV63254.2024.10766004

Kudriavtseva, A., & Gadyatskaya, O. (2023). Secure software development methodologies: A multivocal literature review (Version 2). arXiv. https://doi.org/10.48550/arXiv.2211.16987

Kushwaha, M. K., David, P., & Suseela, G. (2024). Automation and DevSecOps: Streamlining security measures in financial system. 2024 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 1–6. Bangalore, India. https://doi.org/10.1109/CONECCT62155.2024.10677271

Lu, J., Li, S., Zhao, R., Guo, W., & Wang, C. (2024). Research on automatic security sensing technology based on multi-version characterization. 2024 Boao New Power System International Forum – Power System and New Energy Technology Innovation Forum (NPSIF), 1012–1017. Qionghai, China. https://doi.org/10.1109/NPSIF64134.2024.10883282

Martelleur, J., & Hamza, A. (2022). Security Tools in DevSecOps : A Systematic Literature Review (Dissertation). Retrieved from https://urn.kb.se/resolve?urn=urn:nbn:se:lnu:diva-118400

Meliala, R., Lim, C., & Andreas, J. (2024). Integrating security testing in CI/CD pipelines: Current trends from literature and market. 2024 Ninth International Conference on Informatics and Computing (ICIC), 1–6. Medan, Indonesia. https://doi.org/10.1109/ICIC64337.2024.10957011

Nguyen, J. (2020). A multivocal literature review of current tools for increasing the degree of automation in the development of secure and privacy compliant applications.

Nikolov, L. A., & Aleksieva-Petrova, A. P. (2023). Investigación-acción sobre el pipeline de DevSecOps. Conferencia Científica Internacional sobre Ciencias de la Computación (COMSCI) 2023, 1–6. Sozopol, Bulgaria. https://doi.org/10.1109/COMSCI59259.2023.10315920

Piskachev, G., Dziwok, S., Koch, T., Merschjohann, S., & Bodden, E. (2022). How far are German companies in improving security through static program analysis tools? 2022 IEEE Secure Development Conference (SecDev), 7–15. Atlanta, GA, United States. https://doi.org/10.1109/SecDev53368.2022.00015

Saeed, H., Shafi, I., Ahmad, J., Khan, A. A., Khurshaid, T., & Ashraf, I. (2025). Review

of techniques for integrating security in software development lifecycle. Computers, Materials & Continua, 82(1), 139–172. https://doi.org/10.32604/cmc.2024.057587

Samavia Riaz, Ayyan Asif, Younus Khan, Muhammad Ibrar, Saira Afzal, Khalid Hamid, Sehar Gul, & Muhammad Waseem Iqbal. (2025). Software Development Empowered and Secured by Integrating A DevSecOps Design . Journal of Computing & Biomedical Informatics, 8(02). Retrieved from https://www.jcbi.org/index.php/Main/article/view/889

Santos, R., Rizvi, S., Cesarone, B., Gunn, W., & McConnell, E. (2021). Reducing software vulnerabilities using machine learning static application security testing. Proceedings of the 2021 International Conference on Software Security and Assurance (ICSSA), 1–6. https://doi.org/10.1109/ICSSA53632.2021.00016

Sermpezis, E., Karapiperis, D., & Tjortjis, C. (2024). Integration of security in the DevOps methodology. In 2024 15th International Conference on Information, Intelligence, Systems & Applications (IISA) (pp. 1–6). IEEE. https://doi.org/10.1109/IISA62523.2024.10786669

Sinan, M., Shahin, M., & Gondal, I. (2025). Integrating security controls in DevSecOps: Challenges, solutions, and future research directions. Journal of Software: Evolution and Process, 37(6), e70029. https://doi.org/10.1002/smr.70029

Stanciu, A.-M., & Ciocârlie, H. (2023). Analyzing code security: Approaches and tools for effective review and analysis. In 2023 International Conference on Electrical, Computer and Energy Technologies (ICECET) (pp. 1–6). IEEE. https://doi.org/10.1109/ICECET58911.2023.10389326

Valdés-Rodríguez, Y., Hochstetter-Diez, J., Díaz-Arancibia, J., & Cadena-Martínez, R. (2023). Towards the Integration of Security Practices in Agile Software Development: A Systematic Mapping Review. Applied Sciences, 13(7), 4578. https://doi.org/10.3390/app13074578

Wadhams, Z., Reinhold, A. M., & Izurieta, C. (2024). Automating static code analysis through CI/CD pipeline integration. Proceedings of the 2024 IEEE International Conference on Software Analysis, Evolution and Reengineering – Companion (SANER-C 2024). https://doi.org/10.1109/SANER-C62648.2024.00021

Wang, J., & Yan, C. (2021). Automation testing of software security based on BurpSuite. Proceedings of the 2021 International Conference of Social Computing and Digital Economy (ICSCDE), 1–6. https://doi.org/10.1109/ICSCDE54196.2021.00025