

Gestión de la ciberseguridad en entornos de desarrollo ágiles:

Claves para una integración efectiva

Mag. Gabriel Arias Rivero¹

gaboar08@gmail.com

Fecha de recepción: 19 de marzo de 2026

Fecha de aceptación: 31 de agosto de 2026

ARK CAICyT: <https://id.caicyt.gov.ar/ark://ob3t9v4g6>

Resumen

Este artículo se propone explorar y analizar las tendencias y estrategias actuales relacionadas con la integración de la ciberseguridad en los entornos de desarrollo de software basados en metodologías ágiles, partiendo de la hipótesis de trabajo que postula que la integración sistémica y temprana de estos aspectos no solo es viable, sino necesaria y beneficiosa para los proyectos. Para ello, una vez revisada la literatura académica y profesional relevante, se identifican prácticas claves y desafíos persistentes para una efectiva integración. El trabajo se complementa con la exploración de experiencias sobre implementación de DevSecOps. Se evidencia como resultados que, la integración temprana de la seguridad en fases de especificación y diseño de los proyectos más la inclusión de tareas de ciberseguridad específicamente bien definidas, cambiando conceptualmente desde esquemas actuales de revisión/verificación de la misma al final del proceso hacia esquemas shift-left, se visualizan como las tendencias claves y prácticas más efectivas. La automatización del testing de software e inclusión de la seguridad en los flujos de trabajo o pipelines CI/CD, junto a la definición precisa de historias de usuario que incluyan criterios de seguridad en conjunto con la designación de nuevos roles security champions en los equipos, complementan las herramientas disponibles.



Esta obra está bajo una [Licencia Creative Commons Atribución-NoComercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/)

¹ Universidad de la Empresa. Montevideo, Uruguay.
ORCID: <https://orcid.org/0009-0002-5592-3737>

Finalmente, el modelo DevSecOps se consolida como el marco teórico-práctico predominante para equilibrar agilidad y seguridad. Este nuevo esquema requiere de un cambio cultural por parte de las organizaciones y equipos técnicos, que permita abordar los proyectos con un enfoque que integre la seguridad tempranamente desde la planificación, apoyándose en la automatización y la colaboración interdisciplinaria, sin comprometer la velocidad de entrega.

Palabras clave: Ciberseguridad, metodologías ágiles, DevSecOps, integración de seguridad, desarrollo de software.

Resumo

Este artigo propõe-se a explorar e analisar as tendências e estratégias atuais relacionadas à integração da cibersegurança nos ambientes de desenvolvimento de software baseados em metodologias ágeis, partindo da hipótese de trabalho que postula que a integração sistêmica e precoce desses aspectos não é apenas viável, mas necessária e benéfica para os projetos. Para isso, após a revisão da literatura acadêmica e profissional relevante, identificam-se práticas-chave e desafios persistentes para uma integração efetiva. O trabalho é complementado pela exploração de experiências sobre a implementação de DevSecOps. Evidencia-se como resultado que a integração precoce da segurança nas fases de especificação e design dos projetos, somada à inclusão de tarefas de cibersegurança especificamente bem definidas — mudando conceitualmente dos esquemas atuais de revisão/verificação ao final do processo para esquemas shift-left —, consolida-se como a tendência principal e a prática mais eficaz. A automação dos testes de software e a inclusão da segurança nos fluxos de trabalho ou pipelines CI/CD, juntamente com a definição precisa de histórias de usuário que incluam critérios de segurança e a designação de novos papéis de security champions nas equipes, complementam as ferramentas disponíveis. Finalmente, o modelo DevSecOps se consolida como o arcabouço teórico-prático predominante para equilibrar agilidade e segurança. Esse novo esquema exige uma mudança cultural por parte das organizações e equipes técnicas, permitindo abordar os projetos com um enfoque que integre a segurança desde o planejamento, apoiando-se na automação e na colaboração interdisciplinar, sem comprometer a velocidade de entrega.

Palavras-chave: Cibersegurança, metodologias ágeis, DevSecOps, integração de segurança, desenvolvimento de software.

Abstract

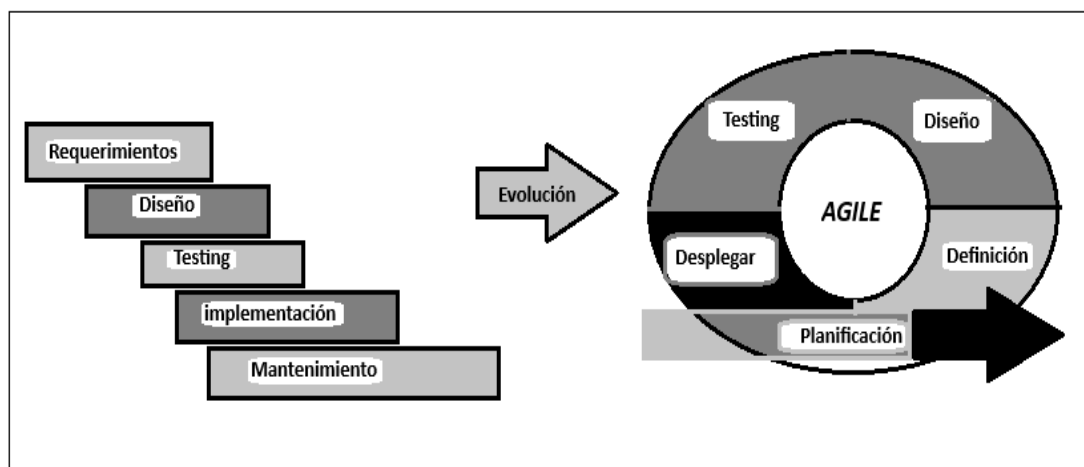
This article aims to explore and analyze current trends and strategies related to integrating cybersecurity into agile software development environments, based on the working hypothesis that the systemic and early integration of these aspects is not only viable, but necessary and beneficial for projects. To this end, following a review of the relevant academic and professional literature, key practices and persistent challenges for effective integration are identified. The study is complemented by an exploration of experiences regarding DevSecOps implementation. Results demonstrate that early security integration during the project specification and design phases, along with the inclusion of clearly defined cybersecurity tasks—conceptually shifting from current end-of-process review/verification models toward shift-left approaches—emerge as key trends and the most effective practices. Automated software testing and the inclusion of security within CI/CD pipelines, alongside the precise definition of user stories with built-in security criteria and the designation of security champions within teams, complement the available toolkit. Finally, the DevSecOps model consolidates itself as the predominant theoretical-practical framework to balance agility and security. This new approach demands a cultural shift within organizations and technical teams, enabling projects to be addressed with a mindset that integrates security early from the planning stage, relying on automation and interdisciplinary collaboration without compromising delivery speed.

Keywords: Cybersecurity, agile methodologies, DevSecOps, security integration, software development.

Introducción

Actualmente, la adopción generalizada de metodologías ágiles como Scrum y Kanban o sus variantes híbridas, están generando una profunda transformación en los procesos de desarrollo y mantenimiento de software (Sharma, 2025; Thool y Brown, 2025), superando las tradicionales metodologías lineales o secuenciales (modelo en cascada) de la ingeniería de software, tan utilizados antaño (IEEE, 2024).

Figura 1 - Modelo en Cascada vs Agile.



Fuente: Elaboración propia.

Con estos nuevos enfoques se intenta priorizar la flexibilidad, la colaboración y la entrega continua de valor (Bedoya et al., 2024). En paralelo con la adopción de estos nuevos paradigmas de la ingeniería de software, la integración de aspectos de seguridad como un componente fundamental en el proceso de desarrollo emerge como un tema de actual vigencia. También es relevante que esta integración no ponga en peligro las virtudes y mejoras aportadas por estos nuevos enfoques de reducción de tiempos y entrega continua de valor.

Entonces, para responder a nuestra hipótesis en cuanto a que la integración temprana y de manera sistémica de los aspectos relacionados con la ciberseguridad en los procesos de desarrollo ágiles de software es viable, necesaria y beneficiosa para los proyectos, este artículo se propone como objetivo examinar las estrategias y prácticas existentes para la gestión de la ciberseguridad en el marco de estos entornos, identificando las tendencias más efectivas y beneficios documentados en la literatura reciente, además de visualizar los desafíos que persisten en su implementación práctica.

Nuevos estándares de desarrollo y tensiones organizacionales

Se indica que la adopción por parte de las empresas, de metodologías con enfoques ágiles, superan el 70% en Estados Unidos, mientras que en China se estima que la cifra ronde ya el 90% (Sharma, 2025; Khwaja et al., 2025). Esto muestra una penetración y adopción muy

generalizada por parte de la industria. Sin embargo, se hace referencia a que en el proceso de introducción de estas metodologías ágiles se generan tensiones significativas en la cultura de las organizaciones y los equipos técnicos (Sharma, 2025; IEEE, 2024). Organizaciones y equipos tradicionalmente afiliados a arquitecturas organizacionales y procesos internos de enfoques secuenciales, documentación exhaustiva y controles rígidos al final de las etapas, que entran en dificultades de adaptación. Estas dificultades y desafíos reportados aumentan, cuando se incorpora una nueva dimensión en los proyectos: los requisitos de ciberseguridad (Sharma, 2025). Requisitos de altísima relevancia, teniendo en cuenta la evolución, asiduidad y complejidad creciente de los ciberataques y la ciberdelincuencia (Sharma, 2025; Bedoya et al., 2024).

Este estado de situación, que agrega el surgimiento de normativas que endurecen los marcos regulatorios en cuanto a la seguridad, hace imperativo el considerar la integración de los temas de ciberseguridad de manera temprana y efectiva en los ciclos de desarrollo de software (Sharma, 2025; Bedoya et al., 2024), sin sacrificar con ello la velocidad de entrega ni la calidad del entregable (Sharma, 2025).

Por tanto, prácticas actuales dentro de los proyectos, como el tomar *deuda de seguridad*, similares a las decisiones de toma de *deudas técnicas* que están correlacionados a las viejas prácticas de postergación de la atención de los requisitos propios de seguridad hacia el final de los ciclos, generan acumulación de riesgos, mayor exposición a ellos e incremento sustancial de costos (Sharma, 2025). Se debe recordar que generar una *deuda* dentro del sprint para un requisito o tema a resolver, implica que se decide postergar la resolución en el momento (Jhala, 2022), desplazándola para más adelante en el cronograma del proyecto o incluso la decisión de resolver ahora, pero utilizando una solución de menor calidad técnica, planificando su resolución completa o mejor posteriormente.

Esta situación es análoga a lo producida cuando surgen cambios que impactan en el diseño de una solución, al detectarse fallas importantes, por ejemplo, sobre el final del ciclo de vida del software o también como en situaciones donde se realiza alguna introducción de cambios en las especificaciones originales (los requerimientos) sobre el final del proceso de desarrollo, una vez que ya se había diseñado, desarrollado y probado el software; en todos estos casos la introducción de cambios en ese momento del ciclo de desarrollo es altamente costosa, con respecto a las estimaciones iniciales (Pressman y Maxim, 2020).

DevOps, Pipeline CI/CD y la incorporación de la dimensión seguridad

La importancia de incorporar aspectos relacionados con la seguridad en los proyectos de desarrollo de software, así como también en las tareas y planificaciones de operación de los sistemas ha contribuido a la extensión del clásico enfoque *DevOps* (Desarrollo y Operaciones), el cual sintetiza el proceso de desarrollo y operaciones del software, como continuo, colaborativo y de ser posible, automatizado (Pressman y Maxim, 2020). Este enfoque está dando paso a uno más integrativo de todos los aspectos, incluidos los de la seguridad, como nuevo estándar de buenas prácticas en la industria (Kim et al., 2021). Además, se suma la automatización de tareas a lo largo del ciclo de desarrollo y mantenimientos, los llamados *pipelines CI/CD* (integraciones continuas del código producido en repositorios comunes que, a través de automatizaciones en la fusión, compilación, testing y generación de entregables, permiten ciclos de entregas continuas del software).

DevSecOps como nuevo enfoque integrativo

Podemos ver entonces a *DevSecOps*, como la evolución directa de *DevOps* que incorpora además la dimensión y requerimientos de seguridad como una responsabilidad compartida e incorporada en todo el ciclo de vida del desarrollo de software, en lugar de tratarla como una fase de verificación final previa al despliegue (Pressman y Maxim, 2020).

Metodologías ágiles y gestión de la seguridad

El proceso de desarrollo ágil de software, se presenta como una alternativa válida y razonable contrapuesta a la ingeniería de software convencional compuesta mayormente de modelos llamados en cascada (Beck, 2000). Cuando los entornos del negocio y los requisitos cambian rápidamente y el software como producto debe lanzarse al mercado a un ritmo elevado, la agilidad en el proceso de desarrollo y mantenimiento se revela adecuado y ventajoso (Martin, 2002). Las metodologías ágiles, no solo aportan un conjunto de herramientas a la ingeniería de software, sino que combinan un enfoque, directrices de trabajo y filosofía de desarrollo, centradas en la adecuada respuesta al cambio (Pressman y Maxim, 2020).

Cuadro 1 – Diferencias de enfoques entre Ágil y Cascada.

Dimensiones	Enfoque Ágil	Enfoque en Cascada
Manejo del cambio	Veloz y continuo	Planificado. Dependiente del cronograma
Entregas y escalabilidad	Tempranas, continuas y en breves lapsos, definición de sprints	Establecidas, en hitos, lapsos estimados previamente, tareas, Gantt
Simplicidad, artefactos generados	Mínima documentación, concentración en construcción y entrega en periodos breves (sprint)	Documentación, análisis y estimación de tiempos, cumplimiento de cronograma de proyecto
Equipos, motivación y organización	Autoorganizados, definición de roles, seguimiento diario, análisis retrospectivo, integración de todas las fases en cada sprint	Jerarquías establecidas, seguimientos definidos, concentración en las tareas y etapas del proyecto

Fuente: Elaboración propia.

Dentro de las distintas metodologías ágiles existentes, se destacan Scrum y Kanban. El primero es un marco de trabajo ágil estructurado en iteraciones de duración fija de tiempo llamadas *sprints* (construcción, prueba y entrega) y el segundo hace énfasis en la gestión del flujo de trabajo con mayor fluidez que Scrum, y se basa en un sistema de tarjetas (tareas, problemas e incidentes) en un tablero (a trabajar, trabajando, finalizados) y con limitación del trabajo en curso (Sharma, 2025; Pressman y Maxim, 2020).

Estas metodologías ágiles favorecen la integración de la gestión de la seguridad al incorporarla como un aspecto más a lo largo de la construcción del software. Lo hace bajo los lineamientos *shift-left* (Sharma, 2025), es decir, bajo la práctica que consiste en mover tareas críticas detectadas, hacia las fases iniciales del ciclo de vida del proyecto, con el objetivo principal de detectar y corregir errores cuando aún son de bajo costo y de fácil solución. Superando de esta manera el enfoque de etapa final donde realizar el control y verificación de estos aspectos, habitualmente utilizado en los modelos iterativos de cascada.

Metodología de Investigación

El trabajo se fundamentó en dos pilares. Por un lado, en la revisión analítica de la literatura académica y profesional publicada entre 2024 y 2025 y para ello, se consultó fuentes de datos académicas (IEEE Xplore, arXiv, Google Académico) y fuentes profesionales especializadas en ciberseguridad y desarrollo de software. Por otro, se aportó en diálogo con la literatura disponible, observaciones y señalamientos en base a la trayectoria profesional del autor, de más de veinticinco años en el sector de las Tecnologías de la Información trabajando en proyectos de consultoría y desarrollo de tecnología para el sector bancario, en Uruguay y diversos países de América Latina y el Caribe. Este diálogo permanente entre los datos cualitativos aportados en la literatura y la perspectiva otorgada por las observaciones del autor, amplió y enriqueció la capacidad de análisis y comprensión de los fenómenos analizados.

Para la estrategia de búsqueda se combinaron términos como ("seguridad informática" OR "ciberseguridad") AND ("metodologías ágiles" OR "agile" OR "scrum" OR "kanban") AND ("DevSecOps" OR "integración de seguridad") y rangos de fechas de publicación ($\geq$ 2024 AND $\leq$ 2026) y la misma se realizó en los meses de febrero-marzo del 2026.

Se priorizaron publicaciones con revisión por pares e informes técnicos de instituciones relevantes del sector TI, en particular enfocadas a los procesos de desarrollo de software y la ciberseguridad, así como información pública aportada por instituciones y empresas reconocidas que incorporan como trasfondo estudios en entornos de la industria. Por cuestiones de enfoque y abordabilidad del objetivo de este trabajo, se seleccionaron los materiales más relevantes según criterios de focalización en el tema, tratamiento evolutivo e integrativo del mismo, prestigio académico del autor o institución asociada, además de la mayor proximidad de publicación dentro del rango de tiempo definido, descartándose aquellos que cumpliesen en menor medida estos criterios. Para complementar los fundamentos teóricos, se suma también aportes de la bibliografía clásica y de base de la ingeniería de software con énfasis en los enfoques ágiles, en ediciones que van desde el 2002 al 2020.

El análisis se estructuró en torno a tres ejes:

- . Modelos de integración de seguridad (DevSecOps, shift-left).
- . Prácticas, estrategias y tendencias específicas (herramientas, procesos, roles).
- . Desafíos y limitaciones reportados en la implementación del enfoque.

Resultados

Cuadro 2 – Cuadro de caracterización de fuentes.

Autor y Año	Título del Estudio	Tipo de Fuente	Metodología	Objetivo Principal
Sharma, S. (2025)	Strategies for Integrating Security into the Software Development Lifecycle. <i>The American Journal of Engineering and Technology</i>	Artículo	Revisión y síntesis de datos cualitativos.	Estrategias para la integración de aspectos de la seguridad en el ciclo de vida del software.
Khwaja, M., Bharanidharan, S., Jamal, E. (2025).	Evolution of DevSecOps and Its Influence on Application Security: A Systematic Literature Review	Artículo	Revisión evolutiva y sistémica. Síntesis con datos cualitativos.	Impacto y evolución de DevSecOps y la integración de aspectos de la seguridad.
Bedoya, M., Palacios, S., Díaz-López, D., Laverde, E., Nespoli, P. (2024)	Enhancing DevSecOps practice with Large Language Models and Security Chaos Engineering	Artículo	Revisión. Análisis de prácticas. Datos cualitativos.	Enfoque DevSecOps con LLM e integración de la seguridad. Cultura organizacional. Revisión y actualización de buenas practicas
DistinThool, A., Brown, C. (2025)	Integrating DAST in Kanban and CI/CD: A Real-World Security Case Study	Artículo	Revisión teórica. Revisión de casos. Datos cualitativos.	Metodologías Agiles, Kanban e integración de la seguridad. Cultura organizacional.
Security Compass. (2025)	Top Six Security Best Practices For Agile Development Environments.	Blog	Síntesis de prácticas, revisión de casos.	Divulgación, gestión de la seguridad. Actualidad en empresas.
IEEE. (2024)	Security Integration in Agile Methodology: A Systematic Review.	Artículo	Revisión teórica. Revisión de casos. Síntesis de temas.	Metodologías Agiles e integración de la seguridad. Cultura organizacional.
Pressman, R., Maxim, B. (2020)	Software Engineering: A Practitioner's Approach (9th ed.).	Libro de texto. Material universitario (clásico) de ingeniería de software.	Revisión teórica. Síntesis de la materia.	Fundamentos de la ingeniería de software. Metodologías de desarrollo de software. Etapas del ciclo de vida del software. Buenas prácticas.
Martin, R. C. (2002).	Agile software development, principles, patterns, and practices	Libro de texto.	Revisión teórica. Síntesis de la materia.	Fundamentos del enfoque Ágil. Ingeniería de software. Prácticas y patrones de diseño.
Beck, K. (2000).	Extreme programming explained: Embrace change.	Libro de texto	Síntesis de la materia.	Principios y fundamentos del desarrollo Ágil. Metodología.
Jhala, V. (2022).	Hands-on DevSecOps: CI/CD security for	Libro de texto.	Manual técnico práctico.	Buenas prácticas DevSecOps.

	cloud-native applications.			Automatización de SAST, DAST y SCA dentro de pipelines CI/CD.
Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021).	The DevOps handbook: How to create world-class agility, reliability, and security in technology organizations	Libro de texto	Síntesis de prácticas. Manual. técnico.	Prácticas y fundamentos de desarrollo DevOps, pipelines CI/CD, automatizaciones, SAST, DAST y SCA.

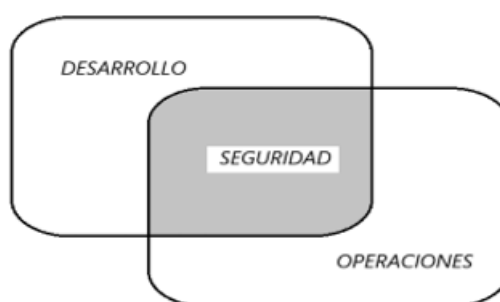
Fuente: Elaboración propia.

El modelo DevSecOps como marco integrador exitoso.

Se señala a DevSecOps como el marco más utilizado y conveniente para integrar agilidad y seguridad (Sharma, 2025; Bedoya et al., 2024). Este enfoque permite incorporar de manera natural los aspectos de seguridad a las dimensiones ya reconocidas del desarrollo y las operaciones, logrando de esta manera complementar y robustecer el ciclo de desarrollo y entrega del software (Sharma, 2025; Khwaja et al., 2025). Además, se entiende que DevSecOps extiende el modelo preexistente DevOps incorporando la seguridad no solo como un aspecto más del ciclo, sino como una responsabilidad compartida entre desarrollo y operaciones (Khwaja et al., 2025), desde etapas tempranas, o sea, desde el inicio del ciclo de vida.

Este enfoque, en definitiva, promueve la cultura de la seguridad (IEEE, 2024), entendida como globalidad y no solo como parte individual dentro los proyectos informáticos, además de transmitir la necesidad de realizar esfuerzos en cuanto a la automatización de controles en cada fase (Thool y Brown, 2025), desde la planificación hasta la entrega, extendiéndola si fuera posible también al monitoreo en producción.

Figura 2 – Marco integrador DevSecOps



Fuente: Elaboración propia.

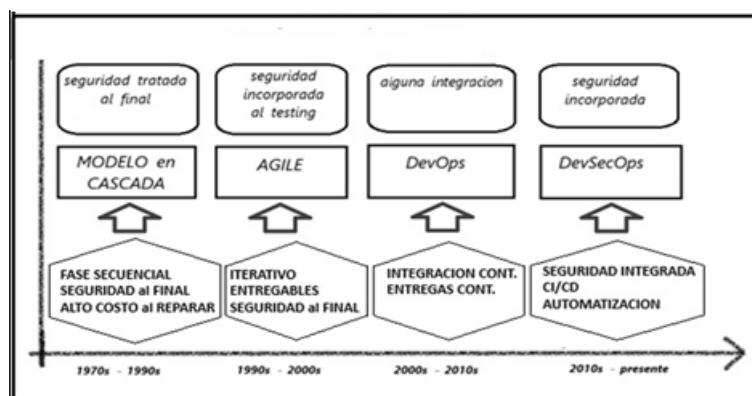
Prácticas efectivas de integración

Por otro lado, del análisis y casos exitosos se suman en las referencias bibliográficas un conjunto de prácticas recurrentes:

Integración temprana de los aspectos en seguridad (esquema Shift-Left)

Se coincide que situar las actividades de seguridad al principio del ciclo de desarrollo, o sea durante la planificación del sprint y el diseño de la solución (esquema shif-left), reduce significativamente la probabilidad de vulnerabilidades en etapas tardías, además de ahorrar en tiempo de retrabajo o de corrección de fallas detectadas (Sharma, 2025; Bedoya et al., 2024; Security Compass, 2025). Esto supone la definición de requisitos de seguridad en la planificación de cada etapa antes de escribir código y por supuesto la realización de pruebas en cada sprint.

Figura 3 – Esquema de seguridad Shifting Left



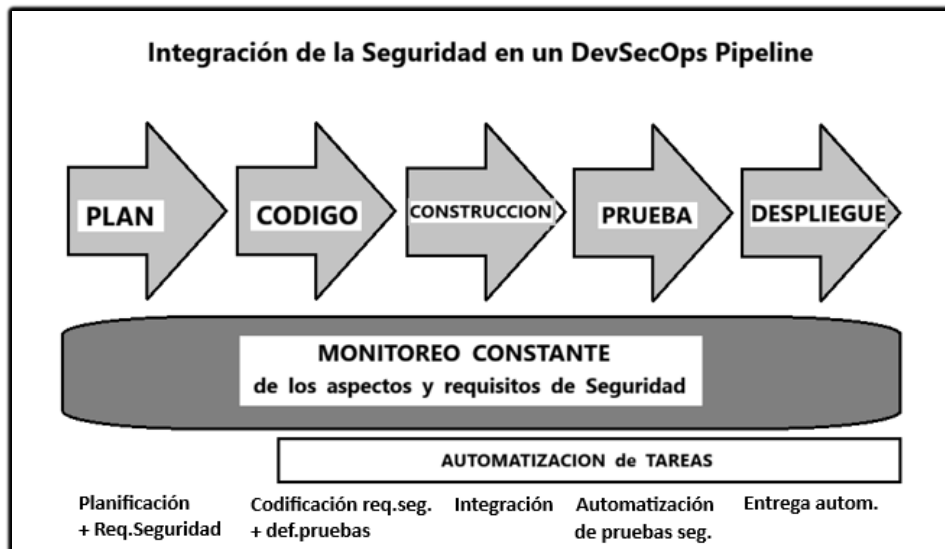
Fuente: Elaboración propia.

Automatización de pruebas en pipelines CI/CD

Otra práctica indicada es la utilización de herramientas de prueba automatizada en los flujos o pipelines de integración y despliegue continuo (Khwaja et al., 2025; Bedoya et al., 2024; Thool y Brown, 2025). Entre ellas se distinguen: SAST (*Static Application Security Testing*) o análisis estático de código fuente en cada commit, DAST (*Dynamic Application Security Testing*), pruebas dinámicas en entornos de especialmente acondicionados y simulación de ataques externos y SCA (*Software Composition Analysis*) análisis de

dependencias y bibliotecas de terceros para identificar vulnerabilidades conocidas (Jhala, 2022; Kim et al., 2021).

Figura 4 – Seguridad incorporada al modelo: DevSecOps



Fuente: Elaboración propia.

Además, se señalan algunos desafíos a enfrentar en estos aspectos ya que, aunque los equipos muestran disposición a adoptar estas herramientas, persisten dificultades relacionadas con la integración técnica y la interpretación de resultados (Khwaja et al., 2025; Bedoya et al., 2024).

Requisitos y criterios de seguridad incluidos en las historias de usuario.

Incluir condiciones explícitas de seguridad dentro de los criterios de aceptación de las historias de usuario (requerimientos del sistema) garantiza que cada funcionalidad sea verificada desde una perspectiva de seguridad (Security Compass, 2025). También, incluir los requisitos de seguridad como tareas a ser desplegadas en cada etapa o sprint, asegura una identificación y priorización de los aspectos de seguridad a lo largo del proyecto.

Designación de roles de especificidad para la seguridad (los llamados security champions).

Un punto novedoso que surge de la revisión, es la aparición de figuras o roles desempeñados dentro de los equipos de trabajo, denominados como "campeones de seguridad". Este nuevo rol es llevado adelante por un miembro del equipo de desarrollo con formación

adicional en seguridad y que actúa como nexo entre el equipo y los expertos (Sharma, 2025; IEEE, 2024). Es de destacar que cumple una función importantísima desde el punto de vista del cambio de cultura que se debe llevar a cabo en las organizaciones y equipos de trabajo para adoptar estos enfoques de cultura de la seguridad. Por tanto, aparece recurrentemente como facilitador de la adopción cultural. Estos roles promueven buenas prácticas, realizan revisiones entre pares y ayudan a priorizar hallazgos.

Desafíos persistentes.

A pesar de los avances realizados, en base a los criterios y prácticas reseñadas anteriormente, se identifican algunos obstáculos aun significativos:

Resistencia cultural.

La percepción de la seguridad como un factor de bloqueo o freno persiste en los equipos y organizaciones donde no se ha internalizado la responsabilidad compartida ni la cultura de la seguridad (Sharma, 2025; Khwaja et al., 2025). Se debe romper con dialécticas de enfrentamiento de unos contra otros (áreas de desarrollo contra sectores de seguridad) transformándolas en una relación cooperadora “ganar-ganar”. Otro óptimo debería ser la integración de áreas para mancomunar trabajo y optimizar costos, convirtiendo a todos los involucrados en parte del proceso de construcción del software y la consecuente generación de valor.

Brecha de habilidades.

Muchos desarrolladores carecen de formación básica en seguridad y aún más específica, en codificación segura, lo que introduce vulnerabilidades evitables (Sharma, 2025; Khwaja et al., 2025).

Equilibrio entre velocidad y rigor.

Tensión para cumplir con los plazos definidos para el sprint y su entrega, así como aplicar controles de seguridad exhaustivos sigue siendo un desafío no resuelto completamente. Parte del cambio organizacional o de cultura que se debe abordar, se encuentra en reconocer los aspectos de la ciberseguridad como requerimientos y el consecuente reconocimiento de esfuerzos y tiempos necesarios para su desarrollo (Sharma, 2025; Bedoya et al., 2024).

Dificultades en la priorización de aspectos de la seguridad.

El encontrar un orden priorizado de los aspectos y requisitos de seguridad dentro del proyecto (Khwaja et al., 2025), así como cuando surgen dificultades no previstas inicialmente, generan tensiones y puntos de mejora a trabajar.

Cuadro 3 – Síntesis de los principales hallazgos.

Conceptos claves	Prácticas exitosas
El modelo <i>DevSecOps</i> como marco integrador exitoso.	Este enfoque permite incorporar de manera natural los aspectos de seguridad a las dimensiones ya reconocidas del desarrollo y las operaciones, logrando de esta manera complementar y robustecer el ciclo de desarrollo y entrega del software.
Integración temprana de los aspectos en seguridad (esquema <i>shift-left</i>).	Situar las actividades de seguridad al principio del ciclo de desarrollo, o sea durante la planificación del <i>sprint</i> y el diseño de la solución— reduce significativamente la probabilidad de vulnerabilidades en etapas tardías.
Automatización de pruebas en <i>pipelines CI/CD</i> .	Utilización de herramientas de prueba automatizada en los pipelines de integración y despliegue continuo.
Requisitos y criterios de seguridad incluidos en las historias de usuario (como parte de los requerimientos).	Inclusión de condiciones explícitas de seguridad dentro de los requerimientos y criterios de aceptación de las historias de usuario.
Designación de roles de especificidad para la seguridad (<i>security champions</i>)	Aparición de figuras o roles desempeñados dentro de los equipos de trabajo, denominados como campeones de seguridad.
Persistencia de desafíos a superar.	Trabajar sostenidamente sobre: • Resistencia cultural. • Brecha de habilidades. • Equilibrio entre velocidad y rigor. • Dificultades en la priorización de aspectos de la seguridad.

Fuente: Elaboración propia.

Discusión

Los hallazgos recientemente expuestos parecen confirmar la hipótesis de partida planteada: la integración temprana y sistémica de la ciberseguridad en los procesos de desarrollo ágiles de software es viable, necesaria y beneficiosa. Requiere de un enfoque

multidimensional que combine delicada y decididamente, tecnología, procesos y, sobre todo, cultura organizacional.

Por tanto, el cambio de cultura para incorporar la ciberseguridad como otro elemento fundamental de los proyectos y las organizaciones, es fundamental para una integración que sea finalmente exitosa (Sharma, 2025; Khwaja et al., 2025). Tenemos que, la paciencia y gradualidad de avances en la implantación de estos enfoques de seguridad, deben ser proporcionales a los tamaños de las organizaciones.

La preeminencia del modelo DevSecOps observada en la literatura refleja un consenso en la industria sobre la necesidad de superar el enfoque tradicional de la seguridad como etapa final y de salida de un proyecto (Khwaja et al., 2025; Bedoya et al., 2024). No obstante, la mera incorporación de herramientas sin el correspondiente cambio de cultura en la organización y los equipos de trabajo conduce a resultados subóptimos. La automatización de pruebas en pipelines CI/CD (Khwaja et al., 2025; Bedoya et al., 2024; Thool y Brown, 2025), si bien es una condición esperable, no es suficiente si los equipos de trabajo no están capacitados para interpretar y actuar sobre los hallazgos.

La importancia de incluir criterios de seguridad en los requerimientos y las historias de usuario (Security Compass, 2025), así como identificar tempranamente requisitos de seguridad (Sharma, 2025; Bedoya et al., 2024), conecta directamente con el principio ágil de entregar valor de manera iterativa y continua. Al definir la seguridad como parte del valor entregado en cada sprint, se alinean los incentivos y se facilita la trazabilidad para auditorías, incluido el cumplimiento normativo.

Los desafíos identificados, especialmente la resistencia cultural (Sharma, 2025; Khwaja et al., 2025) y la brecha de habilidades (Sharma, 2025; IEEE, 2024) apuntan a dos áreas de mejora prioritaria. La primera, que es fundamental, requiere una estrategia de consolidación con enfoque gradual, pero persistente, de capacitación y concientización. La segunda, demanda formación continua y contextualizada (IEEE, 2024), que incluye la incorporación de profesionales formados en ciberseguridad en las plantillas de los equipos de trabajo.

Una limitación que emerge en el presente trabajo, es que no se dispone en la literatura revisada, de información aportada por estudios longitudinales y cuantitativos, que releven datos sobre algunos indicadores objetivo, por ejemplo, reducción de vulnerabilidades en producción, estadísticas de tiempo de reparación de fallas encontradas en producción, comparativos en los tiempos empleados dentro de los sprints con y sin requisitos de seguridad incluidos, etc. Futuras

investigaciones podrían abordar estos tipos de estudios en base a organizaciones con integraciones realizadas y maduras.

De la discusión de resultados se reafirman los puntos clave destacados para una integración efectiva que vehiculice una posible gestión de la ciberseguridad en entornos de desarrollo ágiles.

Al transitar este enfoque de integración, se identifican claramente beneficios asociados a una exitosa gestión.

Cuadro 4 –Beneficios identificados con la integración.

Contemplación efectiva y real de los aspectos de ciberseguridad en el software construido.
Reducción de costos en los proyectos de desarrollo, con repercusiones de baja de costos también en producción.
Cumplimiento normativo.
Mayor confianza en el producto final y repercusiones positivas en la calidad del mismo.
Evolución cultural dentro de las organizaciones y equipos de trabajo.

Fuente: Elaboración propia.

No obstante, la persistencia de desafíos y limitaciones a la práctica van desde el fundamental cambio cultural dentro de las organizaciones, que aspiran a integrar la ciberseguridad en sus prácticas de desarrollos ágiles, a las más íntimas, como la internalización de los conceptos de seguridad para que los equipos asuman responsabilidades en la materia o la falta de talentos formados técnicamente en ciberseguridad.

Conclusiones

La gestión de la ciberseguridad ha evolucionado desde posiciones reactivas y periféricas dentro de los proyectos, hacia modelos más integrados.

En particular, luego de la aparición de las llamadas metodologías ágiles y su expansión dentro de la comunidad técnica y las organizaciones, este enfoque integrativo se hizo de

sustancial importancia sobre todo respecto a los requerimientos de ciberseguridad, el enfrentamiento a la complejidad y riesgos crecientes de los ciberataques y a la sofisticación de la ciberdelincuencia, incluida la potenciación de esta última de mano de la Inteligencia Artificial.

Una práctica que se ha convertido en un verdadero estándar de la industria es *DevOps*, entendida como la integración proactiva de las actividades de desarrollo y operaciones en el ciclo de vida del software, por tanto, la transformación e incorporación de la seguridad en este esquema se revela fundamental, pasando hacia un enfoque *DevSecOps*. Esta integración también de los aspectos de ciberseguridad, en todas las fases del ciclo de vida del software, desde la planificación hasta la puesta en producción, incluida la tendencia a la automatización de las pruebas, redundancia en protección, mitigación de riesgos y mayores beneficios traducidos en tiempo y costo.

La ciberseguridad integrada en entornos ágiles de desarrollo no es un punto de arribo o un destino final. Por el contrario, es un proceso continuo de aprendizaje y adaptación, donde la colaboración interdisciplinaria y la mejora iterativa constituyen la base para construir software seguro y resiliente desde el origen.

6. Referencias

Beck, K. (2000). *Extreme programming explained: Embrace change*. Addison-Wesley.

Bedoya, M., Palacios, S., Díaz-López, D., Laverde, E., Nespoli, P. (2024).

Enhancing DevSecOps practice with Large Language Models and Security Chaos Engineering. <https://doi.org/10.1007/s10207-024-00909-w>

IEEE. (2024). Security Integration in Agile Methodology: A Systematic Review. *2024 IEEE SmartBlock4Africa Conference*. <https://ieeexplore.ieee.org/document/10779565>

Jhala, V. (2022). *Hands-on DevSecOps: CI/CD security for cloud-native applications*.

Packt Publishing.

Khwaja, M., Bharanidharan, S., Jamal, E. (2025). Evolution of DevSecOps and

Its Influence on Application Security: A Systematic Literature Review.

<https://doi.org/10.3390/technologies13120548>

Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021). *The DevOps handbook:*

How to create world-class agility, reliability, and security in technology organizations (2.^a ed.). IT Revolution Press.

Martin, R. C. (2002). *Agile software development, principles, patterns, and practices.*

Prentice Hall.

Pressman, R., Maxim, B. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.).

McGraw-Hill Education.

Security Compass. (2025). Top Six Security Best Practices For Agile Development

Environments. *Security Compass Blog*. <https://www.securitycompass.com/blog/best-practices-agile-development-environments/>

Sharma, S. (2025). Strategies for Integrating Security into the Software Development

Lifecycle. *The American Journal of Engineering and Technology*,

7(04), 151–156. <https://doi.org/10.37547/tajet/Volume07Issue04-20>

Thool, A., Brown, C. (2025). Integrating DAST in Kanban and CI/CD: A Real-World

Security Case Study. *arXiv preprint arXiv:2503.21947*.

<https://doi.org/10.48550/arXiv.2503.21947>