



**Recibido:** 15, enero 2026  
**Aceptado:** 21, enero 2026  
**Publicado:** 23, enero 2026  
**Como citar:** A. Guerra Mera et al., "Diseño y desarrollo de software en la era de la IA: retos éticos, técnicos y operativo", *ARKSIS-Journal*, vol. 1, no. 1, pp. 65-76, doi: 10.63957/arksis.v1i1.0005

## Diseño y desarrollo de software en la era de la IA: retos éticos, técnicos y operativos

### Software design and development in the age of AI: ethical, technical and operational challenges

Alexandra Guerra Mera <sup>ID</sup><sup>a</sup>, José Balseca Manzano <sup>ID</sup><sup>a</sup>

<sup>a</sup>Carrera de Sistemas de Información, Escuela de Hábitat, Infraestructura y Creatividad, Pontificia Universidad Católica del Ecuador, Ambato 180207, Ecuador.

**Autor de correspondencia:** aguerra@pucesa.edu.ec

Creative Commons CC BY 4.0



#### Resumen

El diseño y desarrollo de software ha evolucionado hacia ecosistemas donde la Inteligencia Artificial (IA), particularmente la IA generativa y los modelos de lenguaje (LLMs), intervienen en actividades del ciclo de vida del software (SDLC) como ingeniería de requisitos, construcción, pruebas, despliegue y operación. Este estudio tuvo como objetivo identificar y sintetizar los principales retos técnicos, éticos y operativos asociados al uso de IA en el SDLC, así como las prácticas de control que permiten gestionarlos de manera responsable. Se aplicó un proceso de revisión basado en PRISMA para la identificación, cribado y elegibilidad de estudios, con criterios explícitos orientados a ingeniería de software (excluyendo dominios no relacionados). Los hallazgos muestran beneficios consistentes en productividad y soporte cognitivo (p. ej., generación/transformación de código, apoyo a pruebas y documentación), pero también riesgos recurrentes: introducción de vulnerabilidades por sugerencias no verificadas, alucinaciones, exposición de datos, dependencia tecnológica y degradación de habilidades. En el plano ético, se evidencian desafíos de trazabilidad, sesgo y rendición de cuentas, mientras que en el plano operativo se intensifica la necesidad de prácticas de MLOps/LLMOps, monitoreo, validación continua y gobernanza del uso de herramientas. Se concluye que el impacto positivo de la IA en ingeniería de software depende de marcos éticos aplicables, controles técnicos verificables y disciplina operativa sostenida.

**Palabras Clave:** Inteligencia artificial, ingeniería de software, ética tecnológica, gobernanza operativa.

## **Abstract**

Software design and development has evolved into ecosystems where Artificial Intelligence (AI), particularly generative AI and Language Models (LLMs), are involved in software development life cycle (SDLC) activities such as requirements engineering, construction, testing, deployment, and operation. This study aimed to identify and synthesize the main technical, ethical, and operational challenges associated with the use of AI in the SDLC, as well as the control practices that allow for their responsible management. A PRISMA-based review process was applied for the identification, screening, and eligibility of studies, with explicit criteria focused on software engineering (excluding unrelated domains). The findings show consistent benefits in productivity and cognitive support (e.g., code generation/transformation, test support, and documentation), but also recurring risks: introduction of vulnerabilities through unverified suggestions, hallucinations, data exposure, technological dependence, and skills degradation. On an ethical level, challenges related to traceability, bias, and accountability are evident, while on an operational level, the need for MLOps/LLMOps practices, monitoring, continuous validation, and governance of tool use is intensified. It is concluded that the positive impact of AI in software engineering depends on applicable ethical frameworks, verifiable technical controls, and sustained operational discipline.

**Keywords:** Artificial intelligence, software engineering, technology ethics, operational governance.

## **1. Introducción**

El diseño y desarrollo de software dejó de ser un ejercicio centrado solo en código, requisitos y despliegues. La Inteligencia Artificial (IA), en especial la IA generativa, ya forma parte de decisiones de arquitectura, escritura de código, pruebas, documentación y soporte a operaciones. En la práctica, el equipo de desarrollo ya no conversa únicamente con frameworks y repositorios: también conversa con modelos que sugieren soluciones, completan módulos y proponen refactorizaciones, lo cual cambia la dinámica del trabajo y la trazabilidad de las decisiones técnicas dentro del ciclo de vida del software [1], [2].

Esta nueva realidad trae beneficios evidentes, como aceleración de tareas rutinarias y apoyo cognitivo en actividades complejas. Sin embargo, también instala retos que cruzan tres frentes: técnico, ético y operativo. En el frente técnico, la integración de IA exige criterios más estrictos de validación, ya que el resultado “correcto” puede esconder fallas de seguridad, sesgos de implementación o supuestos no declarados. Estudios empíricos recientes muestran que, ante preguntas de programación, las respuestas de modelos de lenguaje no siempre priorizan prácticas seguras, lo cual incrementa el riesgo de introducir vulnerabilidades cuando el equipo adopta recomendaciones sin verificación sistemática [3], [4].

En el frente ético, la discusión ya no se limita a “usar IA o no usar IA”, sino a cómo se gobierna su uso dentro de productos y procesos. La entrada en vigor del EU AI Act (1 de agosto de 2024) empujó a muchas organizaciones hacia la lógica de cumplimiento; aun así, la literatura advierte que el cumplimiento normativo no equivale

a una práctica ética completa, por lo que hacen falta medidas adicionales que protejan equidad, justicia y responsabilidades a lo largo del ciclo de vida del sistema [4].

En el frente operativo, la adopción de IA dentro de software en producción obliga a fortalecer prácticas de MLOps y de operación continua: control de versiones de datos y modelos, monitoreo de deriva, reproducibilidad, escalabilidad, evaluación post-despliegue y criterios de madurez. Una revisión sistemática reciente identifica buenas prácticas, desafíos recurrentes y modelos de madurez para MLOps, lo cual evidencia que el problema no termina con “entrenar y desplegar”, sino con sostener la confiabilidad del sistema durante su vida útil [5].

Todo esto reconfigura, además, una pieza clásica del desarrollo: la ingeniería de requisitos. Cuando el software incorpora componentes de IA, aparecen requisitos nuevos y más difíciles de especificar (calidad de datos, explicabilidad, confianza, sesgos, límites de desempeño y condiciones de operación). La evidencia mapea vacíos en herramientas y enfoques tradicionales para capturar requisitos de sistemas con IA, lo cual obliga a integrar explícitamente dimensiones de datos y ética desde etapas tempranas [6].

En este contexto, el presente artículo posiciona la Inteligencia Artificial como un catalizador de productividad y, a la vez, como un factor que incrementa la complejidad socio-técnica del desarrollo de software. A partir de la lectura crítica de la literatura especializada y de la escritura analítica de sus hallazgos, se examina cómo la IA redefine prácticas de diseño, construcción y operación, y qué controles permiten equilibrar velocidad con confiabilidad, cumplimiento con ética y automatización con responsabilidad. En consecuencia, la investigación se orienta por la siguiente pregunta: ¿Qué retos éticos, técnicos y operativos emergen en el diseño y desarrollo de software con IA, y qué prácticas permiten gestionarlos de forma responsable a lo largo del ciclo de vida?

De este modo, el desarrollo de software asistido por IA exige una mirada integral que articule controles técnicos verificables, gobernanza ética y disciplina operativa. Solo así la productividad se traduce en valor sostenible: soluciones confiables, seguras y auditables, capaces de evolucionar sin comprometer la responsabilidad ni la confianza de quienes las utilizan.

## **2. Métodos y materiales**

Esta revisión se desarrolló siguiendo PRISMA 2020, estructurando el proceso en cuatro fases: identificación, cribado, elegibilidad e inclusión. La Tabla 1 presenta el diagrama de flujo PRISMA con el número de registros recuperados, los filtros aplicados, los duplicados eliminados y los motivos de exclusión, garantizando trazabilidad y replicabilidad del procedimiento [7].

La recopilación de literatura se centró en Scopus como fuente principal, debido a su cobertura multidisciplinaria y a sus capacidades para rastreo de citas y control de resultados por filtros bibliográficos. La elección de Scopus se respaldó en evidencia reciente que documenta diferencias de cobertura entre bases y destaca el valor de bases robustas para minimizar sesgos de muestreo en revisiones sistemáticas [8] [9].

La búsqueda se realizó en Scopus como fuente principal por su cobertura y trazabilidad bibliométrica. La última búsqueda se ejecutó entre los años 2023 hasta 2025.

- a) Para evitar resultados no pertinentes, se empleó una ecuación orientada explícitamente a ingeniería de software y fases del SDLC, combinando términos de IA generativa/LLMs con actividades del ciclo de vida (requisitos, codificación, refactorización, pruebas, despliegue, operación/MLOps).
- b) Se aplicaron criterios de inclusión/exclusión por (i) pertinencia temática (IA aplicada al SDLC/ingeniería de software), (ii) tipo documental (artículos revisados por pares y, si procede, actas de conferencia en ingeniería de software), (iii) rango temporal 2023–2025, (iv) idioma (español/inglés) y (v) disponibilidad de texto completo para evaluación. El cribado se efectuó en dos etapas: primero por título/resumen y luego por lectura completa, registrando motivos de exclusión para trazabilidad., como se presenta en la Tabla 2.

Tabla 1. Criterios de exclusión y resultados.

	<b>Criterios de exclusión</b>	<b>Resultados</b>
Años	2023-2025	87
Idioma	Inglés / Español	77
Acceso Libre	Libre acceso	60
Tipo de Documento	Artículos	48
Palabras clave	Algorithm Developer, Artificial Intelligence, Artificial Intelligence System Developer	37
Fuente	Journal	37
Area	Computer Science	7

Línea de consulta:

```
TITLE-ABS-KEY ( ( "software engineering" OR "software development" OR "software lifecycle" OR SDLC OR DevOps OR MLOps OR "requirements engineering" OR testing OR "code generation" OR refactoring OR deployment OR monitoring ) AND ( "generative AI" OR "large language model*" OR LLM OR "AI-assisted programming" OR "code assistant" ) ) AND PUBYEAR > 2022 AND PUBYEAR < 2026 AND ( LIMIT-TO ( LANGUAGE , "English" ) OR LIMIT-TO ( LANGUAGE , "Spanish" ) ) AND ( LIMIT-TO ( DOCTYPE , "ar" ) ) AND ( LIMIT-TO ( SUBJAREA , "COMP" ) )
```

La Tabla 2 presenta los criterios de inclusión y exclusión aplicados durante el proceso de cribado, con el propósito de asegurar que los estudios seleccionados mantengan pertinencia temática, consistencia metodológica y calidad documental respecto al análisis del diseño y desarrollo de software en la era de la IA. En ella se detallan los filtros utilizados (rango temporal, acceso abierto, área de investigación, tipo de obra, contenido, fuente, idioma y control de duplicados), lo cual permitió depurar el corpus y conservar únicamente evidencia alineada con los retos éticos, técnicos y operativos del ciclo de vida del software.

Tabla 2. Criterios de inclusión y exclusión.

Criterios	Inclusión	Exclusión
Fecha de publicación	Obras publicadas desde enero de 2023 hasta diciembre de 2025	Anterior al año 2023
Acceso abierto	Artículos de acceso libre	Artículos de acceso no libre
Área de investigación	Ciencias de la Computación, Ingeniería de Software, Sistemas de Información, DevOps/MLOps, Seguridad de software.	Educación/pedagogía, artes/humanidades, agrotecnología, salud clínica, derecho cuando no traten explícitamente actividades del SDLC o ingeniería de software.
Recurso	Artículos científicos completos	Artículos científicos incompletos o que solo presenta resumen
Tipo de obra	Línea de investigación: Desarrollo, TIC	Artículos científicos con otras líneas de investigación
Contenido de resultados	Contenido acerca de desarrollo de software utilizando IA en la era digital	No se trata software de otras áreas
Fuente	Artículos científicos	Resúmenes, revisión sistemática, ensayos
Cantidad	Un artículo científico por autor o autores	Artículos duplicados
Idioma	Español, inglés	Idioma diferente al mencionado

### 3. Resultados

La búsqueda inicial recuperó 85 registros. Tras aplicar filtros por año (2023–2025), idioma, área y tipo documental, se obtuvo un conjunto preliminar de 48 estudios. Posteriormente, el cribado por título y resumen excluyó trabajos fuera del alcance (p. ej., educación, salud, agro, artes u otros dominios sin relación directa con actividades del SDLC). Finalmente, la evaluación a texto completo determinó 7 estudios incluidos por pertinencia directa con ingeniería de software e IA en fases del ciclo de vida.

Los estudios incluidos se agruparon en categorías del SDLC: (i) requisitos, (ii) construcción/refactorización, (iii) pruebas/calidad, (iv) despliegue/operación (DevOps, MLOps/LLMOps) y (v) gobernanza ética y seguridad. Esta clasificación se utilizó para estructurar la síntesis y la discusión.

#### 4. Discusión

Con base en lo anterior y tras aplicar la secuencia de búsqueda y con base en los hallazgos que se consiguieron, la Tabla 3 muestra el análisis de dichos resultados.

Tabla 3. Caracterización de estudios incluidos (IA en el SDLC).

Cita	Referencia
[1]	Agenda de investigación sobre IA generativa aplicada a ingeniería de software; identifica áreas SDLC impactadas y riesgos de fiabilidad.
[3]	Estudio empírico sobre si los LLMs priorizan prácticas seguras al responder preguntas de programación; evidencia brechas de seguridad y necesidad de verificación humana.
[5]	Revisión sistemática de MLOps: buenas prácticas, desafíos y modelos de madurez; enfatiza operación continua, monitoreo y reproducibilidad.
[6]	Mapeo sistemático de ingeniería de requisitos para sistemas de IA; evidencia vacíos en herramientas/enfoques y requisitos no funcionales (explicabilidad, sesgo, confianza).
[11]	Revisión sistemática sobre LLMs para ingeniería de software; clasifica tareas (código, pruebas, reparación, documentación) y discute riesgos/limitaciones.
[12]	Revisión sistemática sobre IA generativa en requisitos; tendencias, retos técnicos (reproducibilidad/controlabilidad) y desafíos de gobernanza.

Los estudios incluidos coinciden en que la IA generativa y los LLMs están impactando múltiples fases del SDLC, pero con riesgos que obligan a reforzar controles. En requisitos, la literatura destaca que la adopción de IA introduce requisitos difíciles de especificar (calidad de datos, explicabilidad, sesgo, límites de desempeño) y que los enfoques tradicionales requieren adaptación para capturar estas dimensiones desde etapas tempranas [6], [12].

En construcción y refactorización, se reportan ganancias de productividad y apoyo cognitivo, pero persisten problemas de confiabilidad (respuestas plausibles pero incorrectas) y necesidad de validación sistemática [1], [11].

En seguridad, la evidencia empírica muestra que los modelos no siempre priorizan prácticas seguras al generar soluciones de programación, lo que incrementa el riesgo de introducir vulnerabilidades si se aceptan recomendaciones sin verificación [2].

En operación, los hallazgos refuerzan que el desafío no termina en “generar” o “integrar” IA: se requiere disciplina de MLOps/LLMOps (versionado, monitoreo de deriva, reproducibilidad, evaluación post-despliegue) para sostener confiabilidad en producción [5], [13].

En conjunto, la discusión sugiere que una adopción responsable equilibra velocidad con confiabilidad mediante: revisión humana, pruebas automatizadas, análisis estático/dinámico, políticas de uso de herramientas, y mecanismos de trazabilidad de decisiones asistidas por IA [14].

A partir del instrumento que detalla la Figura 1, el texto presenta y examina los resultados de la revisión bibliográfica con el fin de identificar las tendencias más

recurrentes y la arquitectura del conocimiento en esta área. Posteriormente, estos hallazgos se contrastan con el objetivo central del estudio, un ejercicio analítico que permite determinar con precisión el grado de pertinencia y la cobertura real del material documental recopilado. Esta fase resulta crítica para validar si la base teórica acumulada es suficiente para sostener las conclusiones de la investigación o si, por el contrario, existen vacíos que comprometen la solidez del análisis final.

#### Documents by subject area

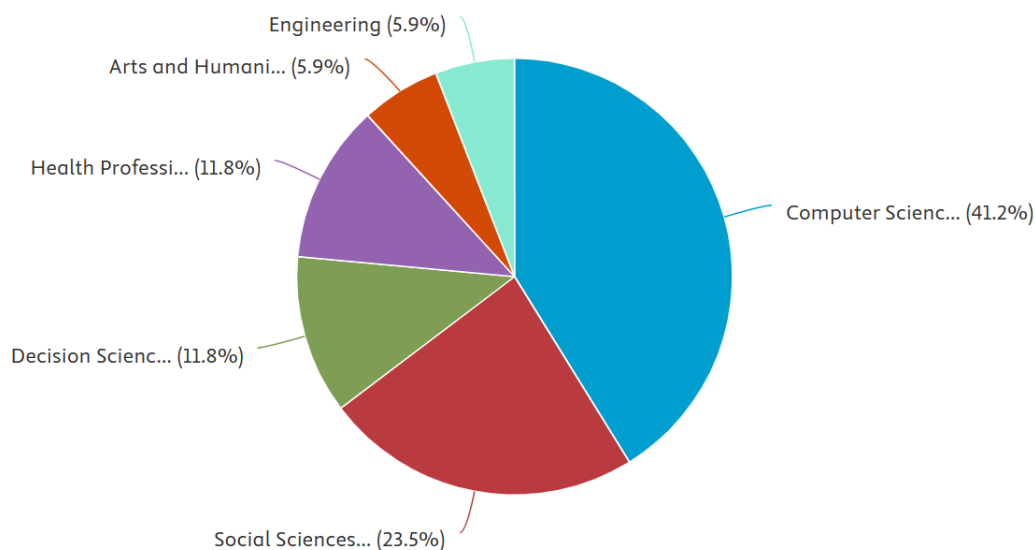


Figura 1. Documentos por área temática

Los datos de la Figura 1 confirman que la literatura sobre Inteligencia Artificial y desarrollo de software gravita con fuerza hacia las Ciencias de la Computación (41,2%), área que robustece los pilares de arquitectura, calidad y seguridad técnica. No obstante, las Ciencias Sociales (23,5%) inyectan una capa necesaria de gobernanza y ética, mientras que las Ciencias de la Decisión y las Profesiones de la Salud (11,8% cada una) enfatizan la precisión operativa en escenarios de alta criticidad.

La Figura 2 revela así que el progreso actual, bajo el impulso de la IA, descansa sobre un tejido interdisciplinario donde convergen la computación, la ingeniería y diversas áreas del saber humano. Esta sinergia es el verdadero motor del avance, pues potencia la innovación y la eficiencia, al tiempo que orienta cada solución hacia las necesidades reales del usuario; en última instancia, la transformación digital plena solo ocurre cuando la ciencia, la técnica y la creatividad se encuentran en un mismo punto.

### Documents by affiliation

Compare the document counts for up to 15 affiliations.

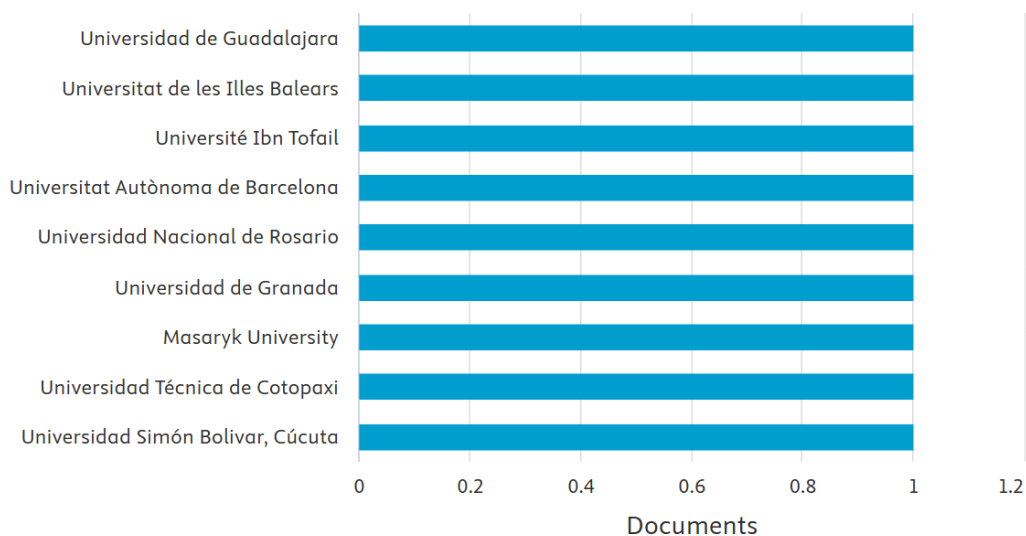


Figura 2. Documentos del patrocinador de la financiación.

La Figura 3 revela una producción científica distribuida de forma equitativa, donde cada afiliación aporta un único documento sin que exista una institución con dominio absoluto. Este fenómeno sugiere que los desafíos éticos, técnicos y operativos del desarrollo de software con Inteligencia Artificial se analizan desde una vasta pluralidad de contextos académicos. Esta fragmentación, lejos de ser una debilidad, fortalece una visión comparativa global y subraya la urgencia de establecer marcos comunes de gobernanza que unifiquen criterios ante una tecnología que no conoce fronteras institucionales.

### Documents by year

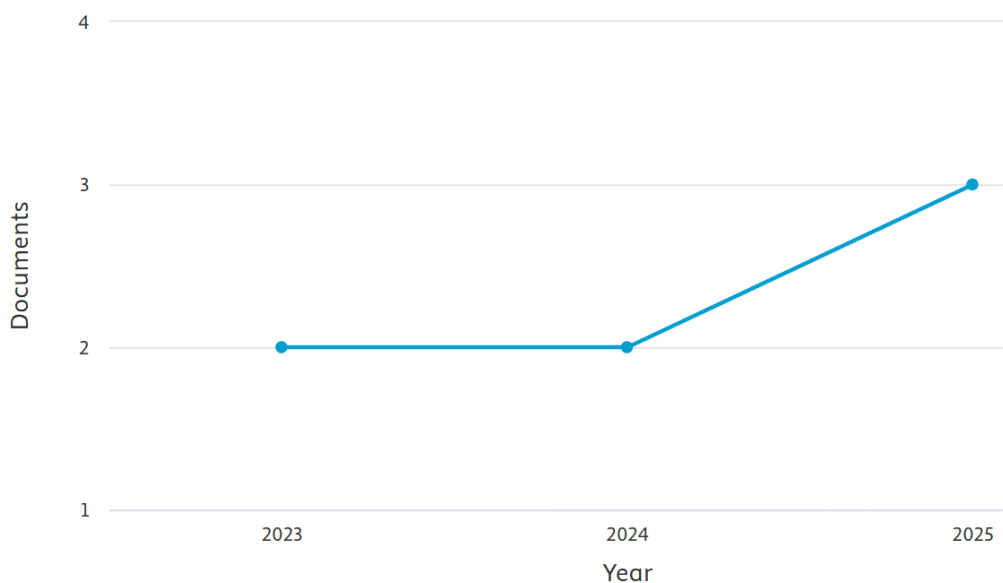


Figura 3. Documentos por año según la fuente.

La Figura 4 revela un incremento reciente en el volumen de publicaciones, con una progresión de dos documentos en 2023, dos en 2024 y tres en 2025. Esta tendencia ascendente confirma que los dilemas éticos, técnicos y operativos en el desarrollo de software bajo la influencia de la IA se han intensificado en el último trienio. Como respuesta directa a esta complejidad, la comunidad científica ha multiplicado su producción y ha acelerado la actualización de los marcos de gobernanza, un fenómeno que demuestra la urgencia por regular y comprender una tecnología que evoluciona a un ritmo superior al de nuestras estructuras institucionales tradicionales.

#### Documents by country or territory

Compare the document counts for up to 15 countries/territories.

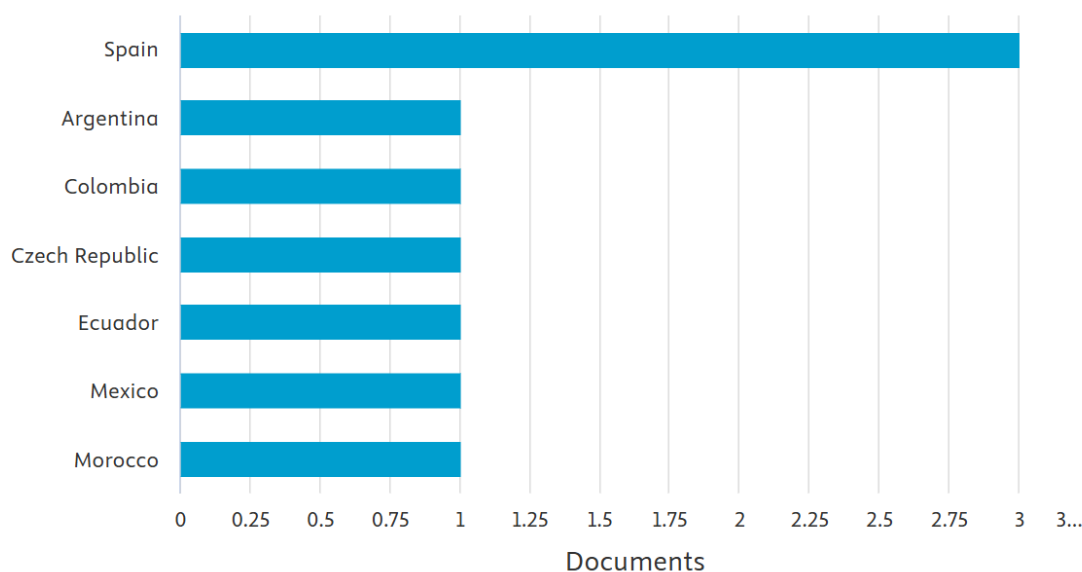


Figura 4. Documentos por país o territorio.

El gráfico confirma el liderazgo de España en este campo con tres documentos, mientras que Argentina, Colombia, República Checa, Ecuador, México y Marruecos contribuyen con una publicación cada uno. Esta distribución geográfica demuestra que los desafíos éticos, técnicos y operativos asociados al software con Inteligencia Artificial son objeto de estudio a escala global. El mapa de la producción científica revela un impulso europeo predominante, pero también resalta una participación emergente y activa de América Latina, lo cual garantiza una visión pluralista en la búsqueda de soluciones para la gobernanza tecnológica.

#### Conclusiones

La literatura revisada evidencia desafíos recurrentes: (i) sesgos y discriminación heredados de datos/modelos, (ii) opacidad y falta de explicabilidad en decisiones asistidas por IA, (iii) responsabilidad y rendición de cuentas cuando la herramienta influye en decisiones de diseño, (iv) privacidad y

uso de datos sensibles (p. ej., incluir código o información propietaria en prompts), y (v) propiedad intelectual y atribución al reutilizar contenido sugerido por modelos. Estos aspectos demandan gobernanza, políticas de uso y evaluación ética continua durante el SDLC.

La tendencia temporal registrada entre 2023 y 2025 revela un incremento constante en el volumen de publicaciones, fenómeno que refleja una maduración acelerada de esta disciplina en la academia. En consecuencia, la práctica profesional actual exige el abandono de "recetas" estáticas en favor de marcos de trabajo dinámicos y actualizados para el diseño, la calidad y la operación de sistemas. Solo a través de esta flexibilidad metodológica será posible responder a los cambios vertiginosos de la tecnología y garantizar que los estándares de excelencia técnica evolucionen a la par de las capacidades de la Inteligencia Artificial.

La investigación actual prioriza las fases de implementación, pruebas y mantenimiento, mientras que las etapas de requisitos y diseño todavía presentan vacíos críticos de conocimiento. Por este motivo, el artículo sostiene que una adopción responsable de la Inteligencia Artificial exige el fortalecimiento de la ingeniería de requisitos para estos sistemas específicos, junto con el establecimiento de criterios verificables desde las fases iniciales del proyecto. Solo a través de esta vigilancia temprana es posible prevenir fallos estructurales y asegurar que los principios de transparencia y seguridad se integren de forma nativa en la arquitectura de la solución.

En operación, el reto principal es sostener confiabilidad y control: (i) incorporación de prácticas MLOps/LLMOps (versionado, despliegue controlado, rollback), (ii) monitoreo de deriva y degradación del rendimiento, (iii) trazabilidad de decisiones asistidas por IA (qué se generó, quién aprobó, con qué evidencia), (iv) gestión de incidentes y vulnerabilidades asociadas a recomendaciones automáticas, y (v) capacitación del equipo para evitar dependencia crítica. La evidencia sugiere que sin disciplina operativa, los beneficios iniciales se convierten en riesgo sostenido.

En el eje ético, el liderazgo europeo que refleja la distribución por países se alinea con un fuerte impulso hacia la regulación y el cumplimiento legal. No obstante, la literatura especializada advierte que el acatamiento de las normas no equivale a una práctica ética integral; la ley es el marco mínimo, pero no el destino final. Por este motivo, las organizaciones deben implementar medidas adicionales que aseguren la equidad, la transparencia y una rendición de cuentas efectiva, elementos que garantizan que el sistema no solo sea legal, sino también justo para todos los usuarios.

A nivel técnico, los principales problemas identificados se concentran en: (i) alucinaciones y soluciones plausibles pero incorrectas, (ii) generación de código inseguro o con malas prácticas si no se valida (p. ej., inyecciones, manejo deficiente de errores), (iii) calidad del código (cohesión, deuda técnica, mantenibilidad), (iv) pruebas insuficientes o inconsistentes cuando se confía solo en el asistente, y (v) riesgo de filtración de información (prompts con secretos,

endpoints o lógica sensible). Por ello, se refuerza la necesidad de verificación humana, pruebas automatizadas y controles de seguridad integrados al pipeline.

### Agradecimientos

Los autores agradecen a las instituciones y comunidades académicas que contribuyeron al desarrollo de esta investigación mediante el acceso a recursos bibliográficos y bases de datos especializadas, en particular *Scopus*, que permitió estructurar la revisión sistemática. Asimismo, se reconoce el apoyo institucional de la Pontificia Universidad Católica del Ecuador, Sede Ambato, por el acompañamiento académico brindado para el análisis crítico de los retos éticos, técnicos y operativos vinculados al diseño y desarrollo de software en la era de la Inteligencia Artificial, así como por facilitar espacios y herramientas para la elaboración del manuscrito.

### Declaración de conflicto de intereses

Los autores no han declarado ningún posible conflicto de intereses en relación con esta investigación, la autoría y/o la publicación de este artículo.

### Contribución de los autores

Los autores declaran sus contribuciones en la siguiente matriz.

	Alexandra Guerra Mera	José Balseca Manzano
Conceptualización		
Análisis formal		
Investigación		
Metodología		
Recursos		
Validación		
Redacción - revisión y edición		

### Referencias

- [1] A. Nguyen-Duc *et al.*, "Generative Artificial Intelligence for Software Engineering—A Research Agenda," *Softw. Pract. Exp.*, Jul. 2025, doi: 10.1002/spe.70005.
- [2] M. L. Siddiq, J. C. S. Santos, S. Devareddy, and A. Muller, "SALLM: Security Assessment of Generated Code," in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering Workshops*, Oct. 2024, pp. 54–65. doi: 10.1145/3691621.3694934.

- [3] A. Sajadi, B. Le, A. Nguyen, K. Damevski, and P. Chatterjee, "Do LLMs consider security? an empirical study on responses to programming questions," *Empir Softw Eng*, vol. 30, no. 4, p. 101, Jul. 2025, doi: 10.1007/s10664-025-10658-6.
- [4] S. Westerstrand, "Fairness in AI systems development: EU AI Act compliance and beyond," *Inf. Softw. Technol.*, vol. 187, p. 107864, Nov. 2025, doi: 10.1016/j.infsof.2025.107864.
- [5] M. Zarour, H. Alzabut, and K. T. Al-Sarayreh, "MLOps best practices, challenges and maturity models: A systematic literature review," *Inf. Softw. Technol.*, vol. 183, p. 107733, Jul. 2025, doi: 10.1016/j.infsof.2025.107733.
- [6] K. Ahmad, M. Abdelrazek, C. Arora, M. Bano, and J. Grundy, "Requirements engineering for artificial intelligence systems: A systematic mapping study," *Inf. Softw. Technol.*, vol. 158, p. 107176, Jun. 2023, doi: 10.1016/j.infsof.2023.107176.
- [7] M. J. Page *et al.*, "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews," *BMJ*, vol. 372, p. n71, Mar. 2021, doi: 10.1136/bmj.n71.
- [8] V. K. Singh, P. Singh, M. Karmakar, J. Leta, and P. Mayr, "The journal coverage of Web of Science, Scopus and Dimensions: A comparative analysis," *Scientometrics*, vol. 126, no. 6, pp. 5113–5142, 2021.
- [9] P. Mongeon and A. Paul-Hus, "The journal coverage of Web of Science and Scopus: a comparative analysis," *Scientometrics*, vol. 106, no. 1, pp. 213–228, Jan. 2016, doi: 10.1007/s11192-015-1765-5.
- [10] J. Horáková, "What can we learn from artificial intelligence about video art? Research on the Vasulka Live Archive; [¿Qué podemos aprender de la inteligencia artificial sobre el videoarte? Investigación sobre el archivo Vasulka Live]," *Artnodes*, vol. 2025-June, no. 36. 2025. doi: 10.7238/artnodes.v0i36.430699.
- [11] X. Hou *et al.*, "Large Language Models for Software Engineering: A Systematic Literature Review," *ACM Trans. Softw. Eng. Methodol.*, vol. 33, no. 8, pp. 1–79, Nov. 2024, doi: 10.1145/3695988.
- [12] H. Cheng *et al.*, "Generative AI for Requirements Engineering: A Systematic Literature Review," *Softw. Pract. Exp.*, vol. 56, no. 2, pp. 141–170, 2026, doi: 10.1002/spe.70029.
- [13] C. K. Tantithamthavorn, F. Palomba, F. Khomh, and J. J. Chua, "MLOps, LLMOps, FMOPs, and Beyond," *IEEE Softw.*, vol. 42, no. 1, pp. 26–32, Jan. 2025, doi: 10.1109/MS.2024.3477014.
- [14] M. Souppaya, K. Scarfone, and D. Dodson, "Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities," National Institute of Standards and Technology, NIST Special Publication (SP) 800-218, Feb. 2022. doi: 10.6028/NIST.SP.800-218.