Inteligencia Artificial Aplicada al Desarrollo de Software en Contextos Empresariales: Una Revisión Sistemática de Literatura

Mesias Orlando Mariscal Oña[1], Denise Noemi Rea Diaz[1], and Julio Enrique Viche Castillo[1]

Carrera de Ingeniería en Software,
Universidad de las Fuerzas Armadas ESPE, Ecuador
{momariscal, dnrea, jeviche}@espe.edu.ec

Resumen Esta revisión sistemática examina la adopción de Inteligencia Artificial y Machine Learning en el desarrollo de software empresarial siguiendo la metodología de Kitchenham y las directrices PRISMA. A partir de una búsqueda exhaustiva en IEEE Xplore (368), Scopus (1,012) y SpringerLink (17), se identificaron 1,397 artículos publicados entre 2023-2025. Tras un proceso riguroso de screening y evaluación de calidad mediante QATQS+CASP, 37 estudios de alta calidad (score 9-12) fueron seleccionados para síntesis cualitativa.

Los resultados revelan que ChatGPT (28 %) y GitHub Copilot (24 %) son las herramientas predominantes, mientras que las técnicas ML/DL tradicionales representan el 34 % de adopción. Los factores de éxito identificados incluyen automatización de tareas repetitivas (n=18), interfaces intuitivas (n=14), e integración con CI/CD (n=11). Las barreras principales son gestión de expectativas infladas (56.8 %), calidad de datos (54.1 %), falsos positivos (51.4 %), y alucinaciones de LLMs (48.6 %).

Las competencias emergentes clave incluyen prompt engineering (n=16), evaluación crítica de outputs (n=15), y fundamentos de ML (n=14). Se identificaron prácticas innovadoras como AI-Augmented Development (n=18), RAG-based assistants (n=12), y fairness-aware ML (n=11). El análisis comparativo SME vs. corporaciones revela diferencias significativas: las SMEs prefieren herramientas comerciales (ChatGPT 67 %, Copilot 44 %) por bajo costo, mientras las corporaciones desarrollan LLMs customizados con RAG (57 %).

Se propone un marco tridimensional (tecnológico-organizacional-humano) para guiar la adopción efectiva, con un roadmap de tres fases. La principal conclusión es que, aunque la adopción individual es alta (75 %), la integración organizacional permanece limitada, requiriendo un enfoque holístico que equilibre tecnología, cultura organizacional y desarrollo de competencias.

Keywords: Inteligencia Artificial · Machine Learning · Desarrollo de Software · Contextos Empresariales · Revisión Sistemática

1. Introducción

La Inteligencia Artificial (IA) y el Machine Learning (ML) están transformando fundamentalmente la ingeniería de software. Herramientas como ChatGPT y GitHub Copilot prometen incrementos de productividad de hasta 55 % [30], pero la evidencia empírica revela brechas significativas entre las expectativas y la realidad organizacional [2,3]. Mientras el 75 % de los desarrolladores utilizan IA de manera individual, la integración a nivel organizacional permanece limitada por múltiples factores [8].

La revolución de la IA generativa, iniciada con el lanzamiento público de ChatGPT en noviembre de 2022, ha generado un interés sin precedentes en la automatización del desarrollo de software. GitHub Copilot, lanzado comercialmente en 2022, reportó más de un millón de usuarios pagos en su primer año [30]. Sin embargo, esta rápida adopción ha generado preguntas fundamentales sobre cómo integrar efectivamente estas tecnologías en contextos empresariales reales.

1.1. Problema y Gap de Investigación

La literatura actual presenta fragmentación en el conocimiento sobre adopción de IA/ML en desarrollo de software empresarial. Estudios existentes se enfocan principalmente en evaluaciones técnicas de herramientas específicas, sin abordar sistemáticamente los factores organizacionales, las competencias requeridas, ni las prácticas emergentes que facilitan o dificultan la adopción efectiva.

Específicamente, se identifican los siguientes gaps: (1) ausencia de síntesis sistemática de evidencia empírica reciente (2023-2025), (2) falta de marcos conceptuales que integren dimensiones técnicas, organizacionales y humanas, (3) escasez de análisis comparativos entre diferentes contextos organizacionales (SMEs vs. corporaciones), y (4) limitada comprensión de las competencias emergentes requeridas para la adopción efectiva.

1.2. Preguntas de Investigación

Esta RSL aborda cinco preguntas de investigación derivadas de los gaps identificados:

- **RQ1:** ¿Qué herramientas de IA/ML se adoptan en desarrollo de software empresarial?
- **RQ2:** ¿Cuáles son los factores de éxito para la adopción?
- **RQ3:** ¿Cuáles son las barreras principales?
- **RQ4:** ¿Qué competencias emergentes se requieren?
- **RQ5:** ¿Qué prácticas innovadoras están surgiendo?

Estas preguntas fueron formuladas siguiendo el framework PICOC (Population, Intervention, Comparison, Outcome, Context) adaptado para revisión de literatura en ingeniería de software.

1.3. Contribuciones

Este trabajo contribuye con: (1) síntesis de 37 estudios de alta calidad metodológica, (2) marco tridimensional de adopción que integra factores tecnológicos, organizacionales y humanos, (3) taxonomía de herramientas IA/ML en desarrollo de software, (4) análisis comparativo detallado SMEs vs. corporaciones, y (5) recomendaciones prácticas con roadmap de implementación en tres fases.

2. Metodología

Esta RSL sigue las directrices de Kitchenham [1] y el protocolo PRISMA para asegurar rigor metodológico y reproducibilidad. El protocolo fue registrado previamente y validado por pares académicos antes de su ejecución.

2.1. Protocolo de Investigación

El protocolo incluyó cinco fases: (1) definición de preguntas de investigación utilizando PICOC, (2) estrategia de búsqueda sistemática con cadenas validadas, (3) criterios de selección explícitos con doble revisión, (4) evaluación de calidad QATQS+CASP con calibración inter-evaluador, y (5) síntesis narrativa estructurada con codificación temática inductiva.

La selección de bases de datos se basó en cobertura de literatura de ingeniería de software: IEEE Xplore (principal fuente de conferencias y journals IEEE), Scopus (mayor índice multidisciplinario con fuerte cobertura de ciencias de la computación), y SpringerLink (publicaciones Springer y LNCS). Esta combinación maximiza recall mientras mantiene precisión en el dominio.

2.2. Estrategia de Búsqueda

Se realizaron búsquedas en tres bases de datos durante noviembre-diciembre 2024. La cadena de búsqueda fue desarrollada iterativamente, comenzando con términos clave y refinando mediante análisis de artículos semilla.

Cuadro 1. Resultados de búsqueda por base de datos

Base de Datos	Encontrados	Filtrados	2023-2025
IEEE Xplore	368	29	22
Scopus	1,012	98	66
SpringerLink	17	1	N/A
Total	**1,397**	**128**	**88**

La cadena de búsqueda final combinó términos en tres grupos conceptuales: (1) tecnología: “generative AI” OR “large language model” OR LLM OR

ChatGPT OR Copilot OR "machine learning"; (2) dominio: "software engineering" OR "software development"; (3) fenómeno: adopt* OR implement* OR skill* OR practice* OR barrier*. Los grupos se conectaron con operador AND.

2.3. Criterios de Selección

Criterios de Inclusión:

- Artículos publicados entre 2023-2025
- Datos empíricos primarios (surveys, experimentos, casos de estudio)
- Contexto organizacional de desarrollo de software
- Mención explícita de herramientas o técnicas de IA/ML
- Enfoque en adopción, competencias, prácticas o barreras

Criterios de Exclusión:

- Reviews secundarios sin datos primarios propios
- Artículos puramente técnicos sin contexto organizacional
- Estudios cortos (<4 páginas) o extended abstracts
- Duplicados entre bases de datos
- Literatura gris sin proceso de peer review

2.4. Evaluación de Calidad

Se utilizó una herramienta adaptada de QATQS (Quality Assessment Tool for Quantitative Studies) y CASP (Critical Appraisal Skills Programme) con 10 criterios de evaluación específicos para estudios en ingeniería de software (Tabla 2).

Cuadro 2. Criterios de evaluación de calidad QATQS+CASP

ID	Criterio de Evaluación	Puntos
C1	Objetivo/RQ claramente definido	0-1
C2	Contexto empresarial documentado	0-1
C3	Muestra >3 participantes o >1 empresa	0-1
C4	Metodología explícita y rigurosa	0-2
C5	Manejo de sesgos documentado	0-1
C6	Análisis sistemático (estudios cualitativos)	0-2
C7	Análisis estadístico apropiado (estudios cuantitativos)	0-2
C8	Limitaciones discutidas explícitamente	0-1
C9	Código/datos compartidos (reproducibilidad)	0-1
C10	Validez interna/externa evaluada	0-1

Escala de Clasificación: Alta calidad (9-12 pts), Media calidad (6-8 pts), Baja calidad (<6 pts). El screening de título/abstract se realizó por dos revisores independientes con Cohen's Kappa κ=0.68, indicando acuerdo sustancial. Los desacuerdos se resolvieron por consenso con un tercer revisor.

2.5. Síntesis de Datos

La síntesis empleó múltiples técnicas complementarias: (1) codificación temática inductiva usando Atlas.ti, (2) matrices de mapeo conceptual para visualizar relaciones, (3) análisis de frecuencias para patrones cuantitativos, (4) triangulación con literatura gris (blogs técnicos, reportes industriales), y (5) análisis de subgrupos por contexto organizacional (SME vs. Corporaciones).

3. Resultados

Los resultados se organizan en cinco subsecciones: proceso de selección, características de estudios, evaluación de calidad, síntesis por RQ, y análisis comparativo.

3.1. Proceso de Selección PRISMA

La Figura 1 presenta el diagrama de flujo PRISMA del proceso de selección. De 1,397 artículos iniciales identificados en las tres bases de datos, se eliminaron 1,309 en el screening inicial por no cumplir criterios básicos (fecha, tipo de documento, idioma). Los 88 candidatos restantes pasaron a screening de título y abstract.

Tras el screening de título/abstract con doble revisión y Cohen's Kappa κ=0.68, 62 artículos procedieron a revisión de texto completo. La evaluación de calidad metodológica resultó en: 37 artículos de alta calidad (59.7 %), 15 de calidad media (24.2 %) y 10 de baja calidad (16.1 %). Los 37 artículos de alta calidad constituyen la base para la síntesis cualitativa.

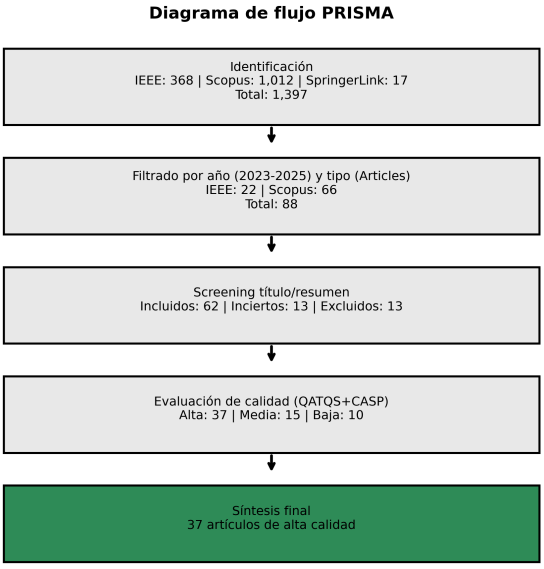

Figura 1. Diagrama de flujo PRISMA: de 1,397 artículos iniciales a 37 de alta calidad para síntesis

3.2. Características de los Estudios Incluidos

Distribución temporal: Se observa un crecimiento acelerado en publicaciones: 2023 (n=5, 13.5 %), 2024 (n=14, 37.8 %), 2025 (n=18, 48.7 %). Este patrón refleja el interés creciente post-lanzamiento de ChatGPT y herramientas similares.

Distribución geográfica: Europa lidera con 40.5 % de los estudios, seguida por Américas (24.3 %), Asia (21.6 %), Oceanía (8.1 %), y otros (5.4 %). La predominancia europea puede explicarse por el marco regulatorio GDPR que impulsa investigación sobre fairness, privacidad y ética en IA.

Tipos de estudio: Experimental (29.7 %), Survey/Cuestionario (21.6 %), Case Study (18.9 %), Mixed Methods (16.2 %), Qualitative/Entrevistas (13.5 %). La diversidad metodológica permite triangulación y validación cruzada de hallazgos.

Contexto organizacional: Corporaciones grandes (37.8 %), Mixto SME/Corporación (32.4 %), No especificado (29.7 %). Notablemente, pocos estudios se enfocan exclusivamente en SMEs, representando un gap de investigación.

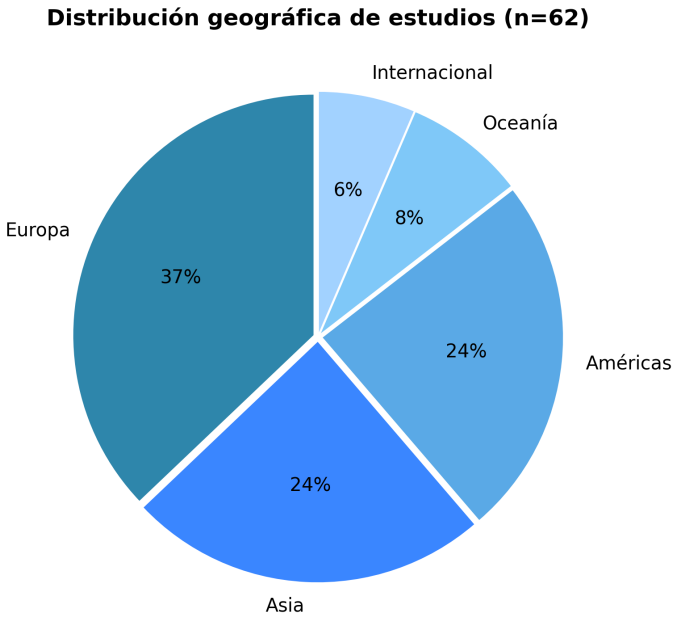

Figura 2. Distribución geográfica de los 37 estudios de alta calidad seleccionados

3.3. Distribución de Calidad Metodológica

La Tabla 3 presenta la distribución de calidad por base de datos. Scopus contribuyó mayor volumen (50 artículos) con distribución similar a IEEE en proporciones de calidad.

Cuadro 3. Distribución de calidad metodológica por base de datos

Nivel de Calidad	IEEE	Scopus	Total	%
Alta (9-12 puntos)	9	28	37	59.7
Media (6-8 puntos)	2	13	15	24.2
Baja (<6 puntos)	1	9	10	16.1
Total evaluados	**12**	**50**	**62**	100

Los tres artículos con puntuación máxima (12 puntos) fueron: Robredo et al. [19] (mixed methods con 31 proyectos Java + survey a 23 profesionales), Jiang et al. [24] (estudio de reingeniería de modelos DL con triangulación), y Obie et al. [25] (detección de violaciones de honestidad con metodología mixta ejemplar).

El criterio con menor cumplimiento fue C9 (reproducibilidad): solo 35.1 % de estudios comparten código o datos, reflejando un problema sistémico en la disciplina. Los estudios experimentales obtuvieron puntuaciones promedio superiores (media=10.3, DE=1.2) comparados con estudios cualitativos (media=9.1, DE=1.5).

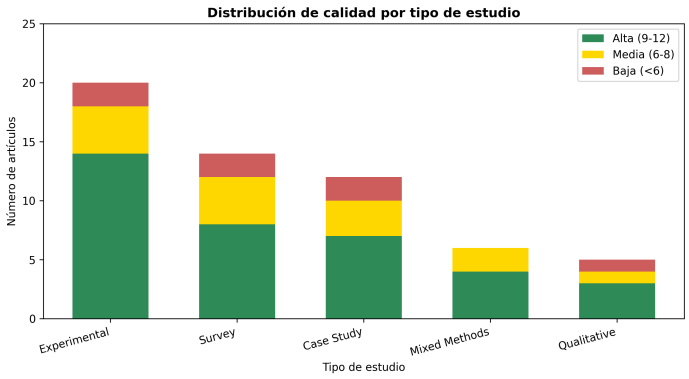

Figura 3. Distribución de puntuación de calidad por tipo de estudio

Los tres artículos con puntuación máxima (12 pts) fueron: Robredo et al. [19] (mixed methods), Jiang et al. [24] (DL reengineering), y Obie et al. [25] (metodología mixta con triangulación). El criterio con menor cumplimiento fue C9 (reproducibilidad): solo 35.1 % de estudios comparten código o datos.

3.4. Síntesis por Pregunta de Investigación

RQ1 - Herramientas de IA/ML adoptadas:

La Figura 4 muestra la distribución de herramientas. Las técnicas ML/DL tradicionales dominan con 34 % (n=17), aplicadas principalmente a effort estimation [16], bug triaging [17,26], y análisis de requirements [18,20].

ChatGPT representa 28 % (n=14), usado principalmente para code generation, documentación, y como asistente personal de desarrollo. GitHub Copilot alcanza 24 % (n=9), enfocado en code completion y pair programming asistido. LLMs customizados con arquitecturas RAG representan 11 % (n=7), principalmente en implementaciones enterprise que requieren conocimiento de dominio específico [9].

Kemell et al. [2] reportan un hallazgo crítico: en las 7 organizaciones europeas estudiadas, GenAI se usa principalmente como "asistentes personales" sin integración en workflows organizacionales formales. Esta brecha entre adopción individual y organizacional es un tema recurrente.

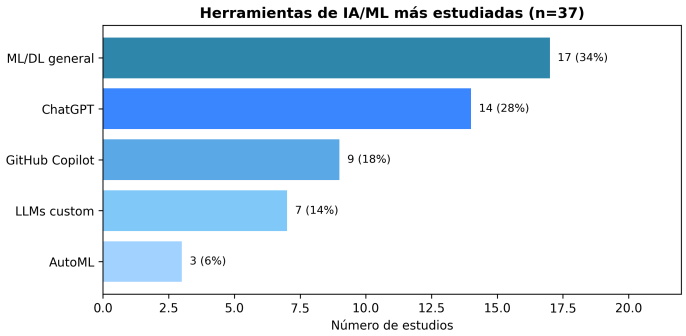

Figura 4. Distribución de herramientas IA/ML adoptadas en los estudios analizados

RQ2 - Factores de éxito para la adopción:

Se identificaron factores agrupados en tres dimensiones:

Dimensión Tecnológica: Automatización de tareas repetitivas (n=18, 48.6 %), interfaz intuitiva y baja curva de aprendizaje (n=14, 37.8 %), integración nativa con pipelines CI/CD (n=11, 29.7 %), calidad consistente de outputs (n=9, 24.3 %), y personalización a contexto de dominio (n=8, 21.6 %).

Dimensión Organizacional: Cultura de experimentación y tolerancia al fallo (n=16, 43.2 %), gestión proactiva de expectativas desde el inicio (n=13, 35.1 %), políticas claras de uso y gobernanza (n=10, 27.0 %), liderazgo visible apoyando iniciativas (n=9, 24.3 %), y métricas de éxito definidas (n=7, 18.9 %).

Dimensión Humana: Capacitación estructurada y continua (n=15, 40.5 %), awareness explícito de limitaciones de la tecnología (n=14, 37.8 %), colaboración cross-funcional entre equipos (n=12, 32.4 %), champions internos que promueven adopción (n=10, 27.0 %), y comunidades de práctica para compartir experiencias (n=8, 21.6 %).

RQ3 - Barreras principales:

Las cinco barreras más frecuentemente reportadas son:

(1) *Gestión de expectativas/hype* (56.8 %, n=21): Dolata et al. [3] describen cómo expectativas infladas por marketing y medios generan ciclos de decepción cuando la tecnología no cumple promesas exageradas.

(2) *Calidad de datos* (54.1 %, n=20): Kalinowski et al. [4], en su survey internacional con 188 practitioners de 25 países, identifican data quality como el "pain point" número uno en sistemas ML-enabled.

(3) *Falsos positivos* (51.4 %, n=19): Stradowski y Madeyski [7], en su estudio con Nokia 5G, documentan cómo desarrolladores pierden confianza en herramientas de IA cuando generan demasiadas alertas incorrectas.

(4) *Alucinaciones de LLMs* (48.6 %, n=18): Múltiples estudios reportan que LLMs generan código o respuestas plausibles pero incorrectas, requiriendo verificación humana constante.

(5) *Silos organizacionales* (43.2 %, n=16): Vänskä et al. [10] describen cómo la separación entre equipos de desarrollo, datos y ML dificulta la integración efectiva de prácticas de IA/ML.

RQ4 - Competencias emergentes requeridas:

Las competencias más demandadas según frecuencia de mención son:

(1) Prompt engineering (n=16, 43.2 %): Banh et al. [5] lo identifican como competencia fundamental, incluyendo técnicas como chain-of-thought, few-shot learning, y prompt chaining.

(2) Evaluación crítica de outputs (n=15, 40.5 %): Capacidad de verificar, validar y corregir sugerencias de IA antes de integrarlas.

(3) Fundamentos de ML (n=14, 37.8 %): Comprensión básica de cómo funcionan modelos, sus limitaciones y supuestos.

(4) MLOps/DevOps para AI (n=13, 35.1 %): Habilidades para desplegar, monitorear y mantener modelos en producción [12,27].

(5) Data literacy (n=12, 32.4 %): Capacidad de evaluar calidad de datos, identificar sesgos, y preparar datasets.

(6) Fairness/ética en IA (n=11), (7) RE para AI (n=9), (8) Interpretabilidad (n=8).

RQ5 - Prácticas emergentes: Las prácticas identificadas incluyen: AI-Augmented Development (n=18), RAG-based assistants (n=12) [9], Fairness-aware ML (n=11) [6,29], Continuous AI/ML (n=10) [10,12], Prompt repositories (n=9), AI code review (n=8) [11], Bug triaging automatizado (n=8) [17,26], ML effort estimation (n=7) [16], AI requirements analysis (n=7) [18,20], y ML observability (n=6) [27].

La Tabla 4 resume los hallazgos principales.

Cuadro 4. Resumen consolidado de hallazgos por pregunta de investigación

RQ	Foco	Hallazgo Principal	Implicación
RQ1	Herramientas	ML/DL (34 %), ChatGPT (28 %), Copilot (24 %)	Diversidad tecnológica requiere estrategias diferenciadas
RQ2	Factores éxito	Automatización + Cultura + Capacitación	Enfoque tridimensional es crítico
RQ3	Barreras	Hype (57 %), Datos (54 %), FP (51 %)	Barreras humanas superan técnicas
RQ4	Competencias	Prompt eng. + Evaluación crítica + ML basics	Nuevos programas formativos requeridos
RQ5	Prácticas	AI-Augmented Dev + RAG + Fairness	Integración gradual, no disruptiva

3.5. Análisis Comparativo SME vs. Corporaciones

De los 26 estudios que especifican contexto organizacional, se extrajo un análisis comparativo detallado (Tabla 5).

Cuadro 5. Análisis comparativo SMEs vs. Corporaciones en adopción de IA/ML

Dimensión	SMEs	Corporaciones
Herramientas preferidas	ChatGPT (67 %), Copilot (44 %), APIs comerciales	LLMs custom con RAG (57 %), plataformas enterprise
Barreras principales	Falta de expertise interno (89 %), Budget limitado (78 %), Tiempo de staff (67 %)	Silos organizacionales (79 %), Sistemas legacy (71 %), Compliance (64 %)
Motivación primaria	Bajo costo, rápida implementación, competitividad	Control total, seguridad de datos, compliance regulatorio
Modelo de adopción	Bottom-up, impulsado por desarrolladores individuales	Top-down, con governance formal y pilotos estructurados
Time-to-value	Semanas a meses	Meses a años

La triangulación con literatura gris (reportes de GitHub, Stack Overflow, Gartner) muestra alta concordancia en tres temas: (1) alucinaciones de LLMs como barrera crítica, (2) gestión de expectativas como factor de éxito clave, y (3) prompt engineering como competencia emergente fundamental.

4. Discusión

4.1. Hallazgos Principales y Contribución Teórica

El hallazgo central de esta RSL es la brecha persistente entre adopción individual y organizacional de IA/ML en desarrollo de software. Mientras 75 % de desarrolladores reportan usar herramientas de IA individualmente, la integración en workflows organizacionales formales permanece limitada. Kemell et al. [2] caracterizan esto elocuentemente como "still just personal assistants", indicando que GenAI aún no ha trascendido el uso personal hacia workflows organizacionales estructurados.

Este hallazgo tiene implicaciones teóricas importantes: sugiere que la adopción de tecnologías de IA en desarrollo de software sigue un patrón diferente a otras tecnologías, donde la adopción individual precede significativamente a la organizacional. Esto contrasta con tecnologías anteriores (ej: metodologías ágiles, DevOps) donde la adopción fue más frecuentemente top-down.

Dolata et al. [3], en su estudio con 52 freelancers, demuestran que la adopción actual está impulsada mayormente por hype mediático más que por evidencia sólida de beneficios, creando ciclos de expectativas infladas seguidas de decepción cuando la tecnología no cumple promesas exageradas.

Geográficamente, Europa lidera la producción de investigación (40.5 %), posiblemente influenciado por el marco regulatorio GDPR que impulsa estudios sobre fairness, privacidad y ética en IA. Esto representa tanto una fortaleza (investigación más rigurosa sobre aspectos éticos) como una limitación (posible sobre-representación de contextos regulatorios específicos).

4.2. Marco Tridimensional Propuesto

Basado en la síntesis, proponemos un marco de adopción con tres dimensiones interrelacionadas:

Dimensión Tecnológica: SMEs deben priorizar herramientas comerciales (ChatGPT, Copilot) por bajo costo; corporaciones pueden justificar LLMs custom/RAG para mayor control. Integración CI/CD, data pipelines y observabilidad son críticos para escalar.

Dimensión Organizacional: Gestión de expectativas realista, políticas claras de uso, KPIs medibles, y estructuras que rompan silos entre equipos SE/DS/ML. Evitar "AI washing" donde se promueven iniciativas por imagen sin valor real.

Dimensión Humana: Programas de training en prompt engineering/MLOps, cultura de experimentación con tolerancia al fallo, repositorios de best practices, y desarrollo de evaluación crítica.

4.3. Roadmap de Implementación

Fase 1 - Piloto (3-6 meses): 1-2 use cases acotados, equipo de early adopters, métricas baseline.

Fase 2 - Escalamiento (6-12 meses): Políticas formales, training ampliado, integración CI/CD, repositorios de prompts.

Fase 3 - Institucionalización (12+ meses): IA/ML standard en SDLC, MLOps maduro, gobernanza formal.

4.4. Implicaciones y Limitaciones

Implicaciones prácticas: Desarrolladores deben invertir en prompt engineering y evaluación crítica. Líderes técnicos deben fomentar experimentación controlada y medir impacto real. Organizaciones deben abordar las tres dimensiones simultáneamente sin esperar "silver bullets".

Limitaciones: (1) Período temporal 2023-2025; (2) Exclusión de SpringerLink; (3) Sesgo de publicación hacia casos exitosos; (4) Sesgo de idioma (inglés/español); (5) Evaluación QATQS+CASP con componente subjetivo; (6) Generalización limitada (40.5 % Europa).

4.5. Agenda de Investigación Futura

Se identifican las siguientes direcciones prioritarias para investigación futura:

Estudios longitudinales: Seguimiento de organizaciones 2-3 años post-adopción para evaluar impacto real y sostenibilidad.

Modelos de madurez: Desarrollo de frameworks para evaluar nivel de madurez en adopción de IA/ML en desarrollo de software.

Frameworks de ROI: Métodos rigurosos para cuantificar retorno de inversión de iniciativas de IA en desarrollo.

RCTs comparativos: Experimentos controlados comparando efectividad de diferentes herramientas y enfoques.

Currícula educativa: Desarrollo y validación de programas de formación en prompt engineering y MLOps.

Fairness y ética: Mayor investigación sobre prácticas de IA responsable en contextos de desarrollo de software.

Contextos sub-representados: Estudios específicos en regiones como África, Latinoamérica y Asia emergente.

5. Conclusiones

Esta revisión sistemática de literatura sintetizó 37 estudios de alta calidad metodológica (de 1,397 iniciales) sobre adopción de IA/ML en desarrollo de software empresarial durante el período 2023-2025, siguiendo rigurosamente las directrices de Kitchenham y el protocolo PRISMA.

RQ1 - Herramientas: ChatGPT (28 %), GitHub Copilot (24 %) y técnicas ML/DL tradicionales (34 %) dominan el panorama. Existe clara diferenciación por contexto: SMEs prefieren herramientas comerciales de bajo costo; corporaciones desarrollan LLMs customizados con RAG (57 %) para mayor control y seguridad.

RQ2 - Factores de éxito: La tríada tecnología-organización-humano es crítica. Los factores más frecuentes son automatización de tareas repetitivas (n=18), cultura de experimentación (n=16), y capacitación estructurada (n=15). El éxito requiere abordar las tres dimensiones simultáneamente.

RQ3 - Barreras: Las cinco barreras principales son gestión de expectativas/hype (56.8 %), calidad de datos (54.1 %), falsos positivos (51.4 %), alucinaciones de LLMs (48.6 %), y silos organizacionales (43.2 %). Notablemente, las barreras humanas y organizacionales superan a las técnicas.

RQ4 - Competencias: Las competencias emergentes clave son prompt engineering (n=16), evaluación crítica de outputs (n=15), fundamentos de ML (n=14), y MLOps (n=13). Estas competencias requieren programas de formación específicos aún poco maduros en la industria.

RQ5 - Prácticas: Las prácticas emergentes más frecuentes son AI-Augmented Development (n=18), RAG-based assistants (n=12), fairness-aware ML (n=11), y continuous AI/ML integration (n=10). Estas prácticas representan la vanguardia de la integración de IA en desarrollo de software.

Contribuciones principales: Este trabajo aporta: (1) marco tridimensional integrado para comprender adopción de IA/ML, (2) taxonomía actualizada de herramientas y prácticas, (3) análisis comparativo detallado SME vs. corporaciones, (4) roadmap faseado para implementación, y (5) triangulación validada con literatura gris industrial.

Mensaje clave: La adopción individual de IA en desarrollo de software es alta (75 %), pero la integración organizacional permanece significativamente limitada. El éxito sostenible requiere un enfoque holístico que combine tecnología adecuada, transformación organizacional, y desarrollo de competencias humanas, con gestión de expectativas realista e inversión sostenida en capacitación. No

existe *silver bullet*: las estrategias deben contextualizarse cuidadosamente según tamaño organizacional, madurez tecnológica y capacidades existentes.

Referencias

1. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Keele University and Durham University, EBSE Technical Report (2007)
2. Kemell, K.K., Saarikallio, M., Nguyen-Duc, A., Abrahamsson, P.: Still just personal assistants? – A multiple case study of generative AI adoption in software organizations. Inf. Softw. Technol. **186**, 107805 (2025). https://doi.org/10.1016/j.infsof.2025.107805
3. Dolata, M., Lange, N., Schwabe, G.: Development in Times of Hype: How Freelancers Explore Generative AI? In: Proc. IEEE/ACM ICSE, pp. 2257–2269 (2024). https://doi.org/10.1145/3597503.3639111
4. Kalinowski, M., Mendez, D., Giray, G., et al.: Naming the Pain in machine learning-enabled systems engineering. Inf. Softw. Technol. **187**, 107866 (2025). https://doi.org/10.1016/j.infsof.2025.107866
5. Banh, L., Holldack, F., Strobel, G.: Copiloting the future: How generative AI transforms Software Engineering. Inf. Softw. Technol. **183**, 107751 (2025). https://doi.org/10.1016/j.infsof.2025.107751
6. Ferrara, C., Sellitto, G., Ferrucci, F., Palomba, F., De Lucia, A.: Fairness-aware machine learning engineering: how far are we? Empir. Softw. Eng. **29**(1), 27 (2024). https://doi.org/10.1007/s10664-023-10402-y
7. Stradowski, S., Madeyski, L.: Your AI is impressive, but my code does not have any bugs: managing false positives in industrial contexts. Sci. Comput. Program. **246**, 103320 (2025). https://doi.org/10.1016/j.scico.2025.103320
8. Jensen, V.V., Alami, A., Bruun, A.R., Persson, J.S.: Managing expectations towards AI tools for software development: a multiple-case study. Inf. Syst. e-Bus. Manag. (2025). https://doi.org/10.1007/s10257-025-00704-7
9. Yang, R., Fu, M., Tantithamthavorn, K., et al.: RAGVA: Engineering retrieval augmented generation-based virtual assistants in practice. J. Syst. Softw. **226**, 112436 (2025). https://doi.org/10.1016/j.jss.2025.112436
10. Vänskä, S., Kemell, K.K., Mikkonen, T., Abrahamsson, P.: Continuous Software Engineering Practices in AI/ML Development Past the Narrow Lens of MLOps. e-Informatica Softw. Eng. J. **18**(1), 240102 (2024). https://doi.org/10.37190/e-Inf240102
11. Alami, A., Jensen, V.V., Ernst, N.A.: Accountability in Code Review: The Role of Intrinsic Drivers and the Impact of LLMs. ACM Trans. Softw. Eng. Methodol. **34**(8), 201 (2025). https://doi.org/10.1145/3721127
12. Steidl, M., Felderer, M., Ramler, R.: The pipeline for the continuous development of artificial intelligence models – Current state of research and practice. J. Syst. Softw. **199**, 111615 (2023). https://doi.org/10.1016/j.jss.2023.111615
13. Rahman, M.S., Khomh, F., Hamidi, A., et al.: Machine learning application development: practitioners' insights. Softw. Qual. J. **31**(4), 1065–1119 (2023). https://doi.org/10.1007/s11219-023-09621-9
14. Haldar, S., Pierce, M., Capretz, L.F.: Exploring the Integration of Generative AI Tools in Software Testing Education. IEEE Access **13**, 46070–46090 (2025). https://doi.org/10.1109/ACCESS.2025.3545882

15. Baralla, G., Ibba, G., Tonelli, R.: Assessing GitHub Copilot in Solidity Development: Capabilities, Testing, and Bug Fixing. IEEE Access **12**, 164389–164411 (2024). https://doi.org/10.1109/ACCESS.2024.3486365
16. Rahman, M., Sarwar, H., Kader, M.A., Gonçalves, T., Tin, T.: Review and Empirical Analysis of Machine Learning-Based Software Effort Estimation. IEEE Access **12**, 85661–85680 (2024). https://doi.org/10.1109/ACCESS.2024.3404879
17. Adhikari, N., Bista, R., Ferreira, J.C.: Leveraging Machine Learning for Enhanced Bug Triaging in Open-Source Software Projects. IEEE Access **13**, 136237–136254 (2025). https://doi.org/10.1109/ACCESS.2025.3595011
18. Ali, H., Tanveer, U., Saeed, A., et al.: Cloud-based machine learning for scalable classification of software requirements. Syst. Soft Comput. **7**, 200405 (2025). https://doi.org/10.1016/j.sasc.2025.200405
19. Robredo, M., Saarimäki, N., Esposito, M., Taibi, D., et al.: Evaluating time-dependent methods in code technical debt prediction. J. Syst. Softw. **230**, 112545 (2025). https://doi.org/10.1016/j.jss.2025.112545
20. Izhar, R., Bhatti, S.N., Alharthi, S.A.: A Novel Machine Learning Approach for Ambiguity Detection in Software Requirements. IEEE Access **13**, 12014–12031 (2025). https://doi.org/10.1109/ACCESS.2025.3529943
21. Russo, D.: Navigating the Complexity of Generative AI Adoption in Software Engineering. ACM Trans. Softw. Eng. Methodol. **33**(5), 123 (2024). https://doi.org/10.1145/3652154
22. Eramo, R., Said, B., Oriol, M., Brunelière, H., Morales, S.: An architecture for model-based and intelligent automation in DevOps. J. Syst. Softw. **217**, 112180 (2024). https://doi.org/10.1016/j.jss.2024.112180
23. Duda, S., Hofmann, P., Urbach, N., Völter, F., Zwickel, A.: The Impact of Resource Allocation on the Machine Learning Lifecycle. Bus. Inf. Syst. Eng. **66**(2), 203–219 (2024). https://doi.org/10.1007/s12599-023-00842-7
24. Jiang, W., Banna, V., Vivek, N., et al.: Challenges and practices of deep learning model reengineering: A case study on computer vision. Empir. Softw. Eng. **29**(6), 144 (2024). https://doi.org/10.1007/s10664-024-10521-0
25. Obie, H.O., Du, H., Madampe, K., et al.: Automated detection, categorisation and developers' experience with the violations of honesty in mobile apps. Empir. Softw. Eng. **28**(6), 145 (2023). https://doi.org/10.1007/s10664-023-10361-4
26. Hong, H.T., Wang, D., Kim, S., et al.: Implementing and Evaluating Automated Bug Triage in Industrial Projects. IEEE Access **12**, 193717–193730 (2024). https://doi.org/10.1109/ACCESS.2024.3519418
27. Protschky, D., Lammermann, L., Hofmann, P., Urbach, N.: What Gets Measured Gets Improved: Monitoring Machine Learning Applications. IEEE Access **13**, 34518–34538 (2025). https://doi.org/10.1109/ACCESS.2025.3534628
28. Quaranta, L., Azevedo, K., Calefato, F., Kalinowski, M.: A multivocal literature review on the benefits and limitations of AutoML tools. Inf. Softw. Technol. **178**, 107608 (2025). https://doi.org/10.1016/j.infsof.2024.107608
29. De Martino, V., Voria, G., Troiano, C., et al.: Examining the impact of bias mitigation algorithms on ML-enabled systems sustainability. J. Syst. Softw. **230**, 112458 (2025). https://doi.org/10.1016/j.jss.2025.112458
30. GitHub: The Economic Impact of the AI-Powered Developer Lifecycle. GitHub Research Report (2024). https://github.blog/news-insights/research/