

FIIADS: A Holistic Framework for Artificial Intelligence Implementation in Software Development

Lic. Marcela Paladino¹ and Dra. Claudia Pons^{1,2,3}

¹ Facultad de Informática; Universidad Nacional de La Plata (UNLP), La Plata, Argentina

² Comisión de Investigaciones Científicas (CIC-PBA), Argentina

³ Universidad Abierta Interamericana (UAI), Buenos Aires, Argentina

marcelpaladino@yahoo.com.ar

Abstract. The integration of Artificial Intelligence (AI) into the Software Development Life Cycle (SDLC) has intensified in recent years, driven by advances in machine learning and, particularly, by the expansion of generative models. Although these technologies offer significant opportunities to improve productivity, automation, and software quality, their effective adoption still faces technical, organizational, and ethical challenges that are often addressed in a fragmented manner in the literature. In this context, this paper proposes FIIADS (Framework for the Implementation of Artificial Intelligence in Software Development), a holistic framework aimed at guiding the incorporation of AI throughout the SDLC in a systematic, responsible, and assessable way. The research was conducted under the Design Science Research (DSR) paradigm, based on a critical review of the state of the art and the design of a conceptual artifact structured around three dimensions: technical, organizational, and ethical. The framework includes processes, metrics, and artifacts intended to support AI implementation across different SDLC phases and organizational contexts. Validation was carried out at a conceptual level through comparison with existing frameworks, application to documented cases in the literature, and internal coherence analysis. The results suggest that FIIADS provides a structured and adaptable foundation for the gradual, responsible, and assessable implementation of AI in software development organizations.

Keywords: artificial intelligence, software development life cycle, framework, software engineering, AI governance.

FIIADS: un framework holístico para la implementación de inteligencia artificial en el desarrollo de software

Resumen. La incorporación de la inteligencia artificial (IA) en el ciclo de vida del desarrollo de software (Software Development Life Cycle, SDLC) se ha intensificado en los últimos años, impulsada por avances en aprendizaje automático y, especialmente, por la expansión de modelos generativos. Aunque estas tecnologías ofrecen

oportunidades para mejorar la productividad, la automatización y la calidad del software, su adopción efectiva continúa enfrentando desafíos técnicos, organizacionales y éticos que suelen abordarse de manera fragmentada en la literatura. En este contexto, este trabajo propone FIIADS (Framework de Implementación de Inteligencia Artificial en el Desarrollo de Software), un framework holístico orientado a guiar la incorporación de IA a lo largo del SDLC de forma sistemática, responsable y evaluable. La investigación se desarrolló bajo el paradigma Design Science Research (DSR), a partir de una revisión crítica del estado del arte y del diseño de un artefacto conceptual articulado en tres dimensiones: técnica, organizacional y ética. El framework incluye procesos, métricas y artefactos que permiten orientar la implementación de IA en distintas fases del SDLC y en diversos contextos organizacionales. La validación se realizó a nivel conceptual mediante la comparación con frameworks existentes, la aplicación a casos documentados en la literatura y el análisis de coherencia interna. Los resultados sugieren que FIIADS ofrece una base estructurada y adaptable para la implementación gradual, responsable y evaluable de IA en organizaciones de desarrollo de software.

Palabras clave: inteligencia artificial, ciclo de vida del desarrollo de software, framework, ingeniería de software, gobernanza de IA.

1 Introducción

La incorporación de IA en la ingeniería de software se ha acelerado de manera visible durante los últimos años, especialmente a partir de la expansión de modelos de aprendizaje automático y modelos generativos capaces de asistir tareas cognitivas complejas. En el SDLC, esta transformación ya no se limita a actividades experimentales: hoy alcanza la estimación de proyectos, el análisis de requerimientos, el diseño, la codificación asistida, la automatización de pruebas y el mantenimiento del software (Amershi et al., 2019; Fan et al., 2023; Zhang et al., 2023).

Las oportunidades son significativas. En ciertas tareas de desarrollo, los asistentes basados en IA pueden mejorar la velocidad de producción, automatizar actividades repetitivas y ofrecer apoyo sobre grandes volúmenes de información (Peng et al., 2023). Del mismo modo, los avances en procesamiento de lenguaje natural amplían las posibilidades de automatización en requerimientos, documentación técnica y trazabilidad (Zhao et al., 2021). Sin embargo, estos beneficios no implican una adopción automática, lineal ni exenta de riesgos.

La literatura reciente muestra que la incorporación efectiva de IA en el desarrollo de software continúa siendo problemática porque **los estudios existentes suelen centrarse en herramientas específicas o en fases aisladas del ciclo de vida**. Como consecuencia, las dimensiones organizacionales y éticas quedan subrepresentadas o tratadas de forma secundaria, pese a que son decisivas para el uso sostenible de estas tecnologías en entornos reales (Dignum, 2019; Russo,

2024). Esta fragmentación se vuelve aún más visible en el contexto de la IA generativa, donde la productividad potencial convive con desafíos de transparencia, supervisión humana, privacidad y trazabilidad (Liao & Wortman Vaughan, 2024; Fu et al., 2025).

En ese marco, el objetivo de este trabajo es presentar **FIIADS, un framework holístico para la implementación de inteligencia artificial en el desarrollo de software**. La propuesta se construye como un artefacto metodológico basado en Design Science Research (DSR) (Wieringa, 2014) y se orienta a guiar decisiones de adopción, integración, evaluación y seguimiento de IA mediante tres dimensiones complementarias: técnica, organizacional y ética. La contribución central de este estudio consiste en destacar el fundamento del framework, su estructura, su articulación con el SDLC y los resultados de su validación conceptual.

El resto del artículo se organiza de la siguiente manera. La Sección 2 presenta el marco teórico sobre la adopción de inteligencia artificial en ingeniería de software y los enfoques conceptuales que fundamentan la propuesta. La Sección 3 describe la metodología de investigación basada en Design Science Research. La Sección 4 presenta el framework FIIADS y sus componentes principales. La Sección 5 expone la estrategia de validación conceptual y sus resultados. La Sección 6 discute los hallazgos, contribuciones y limitaciones del estudio. Finalmente, la Sección 7 presenta las conclusiones y líneas de trabajo futuro.

2 Trabajos relacionados y brecha de conocimiento

La evidencia disponible confirma que la IA ya se aplica en múltiples fases del SDLC. En planificación, los modelos predictivos pueden apoyar la estimación de esfuerzo, duración y costos cuando existen datos históricos adecuados (Ali & Gravino 2019; Pospieszny et al., 2018). En requerimientos, las técnicas de Procesamiento de Lenguaje Natural (NLP) permiten detectar ambigüedades, defectos y relaciones de trazabilidad, mientras que los enfoques generativos abren nuevas posibilidades para reformular especificaciones o generar escenarios de prueba (Zhao et al., 2021; Cheng et al., 2026). En diseño y desarrollo, la IA interviene tanto en el análisis de decisiones arquitectónicas como en la generación asistida de código, con mejoras potenciales en productividad, pero también con nuevos riesgos de calidad, seguridad y mantenibilidad (Peng et al., 2023; Fu et al., 2025). En pruebas y mantenimiento, la literatura reporta avances en la generación automática de casos de prueba, en la priorización de regresión, en la predicción de defectos y en el monitoreo del comportamiento del software en producción (Siddiq et al., 2024; Kamei et al., 2013). Sin embargo, la efectividad de estos enfoques depende de manera crítica de la calidad de los datos, de la estabilidad del entorno y de la existencia de mecanismos de monitoreo continuo y validación sistemática (Sculley et al., 2015; Breck et al., 2017; Kreuzberger et al., 2023).

A partir de esta revisión, se identifican tres grupos de barreras recurrentes. En el plano técnico, persisten problemas de interoperabilidad con arquitecturas existentes, dependencia de datos, baja explicabilidad y deuda técnica asociada a sistemas de IA (Sculley et al., 2015; Breck et al., 2017). En el plano organizacional, la resistencia al cambio, la falta de competencias específicas, la indefinición de roles y la ausencia de mecanismos de gobernanza limitan la institucionalización de estas herramientas (Tornatzky & Fleischer, 1990; Venkatesh & Davis, 2000; Russo, 2024; Paladino &

Pons, 2025). En el plano ético, la preocupación por sesgos, trazabilidad, seguridad, privacidad y supervisión humana introduce exigencias adicionales que no pueden ser tratadas como anexos posteriores a la implementación (Dignum, 2019; IEEE, 2020; UNESCO, 2021).

Si bien existen marcos valiosos, ninguno resuelve de forma integrada el problema planteado. Machine Learning Operations (MLOps) aporta una perspectiva sólida para la operación técnica de modelos, pero su foco principal está en el ciclo de vida del modelo y no en el conjunto del SDLC (Kreuzberger et al., 2023). El AI RMF del NIST fortalece la gestión del riesgo y la gobernanza, aunque no prescribe procesos específicos por fase de desarrollo (NIST, 2023). Los marcos éticos internacionales proporcionan principios imprescindibles, pero suelen carecer de traducción operativa en actividades concretas de ingeniería de software (Floridi et al., 2018; European Commission High-Level Expert Group on Artificial Intelligence, 2019). Esta combinación de evidencias justifica la necesidad de un framework holístico específicamente orientado al desarrollo de software.

3 Metodología

La investigación se desarrolló bajo el paradigma de Design Science Research, dado que este enfoque se centra en el diseño y la evaluación de artefactos orientados a resolver problemas complejos del mundo real (Hevner et al., 2004; Gregor & Hevner, 2013). En este caso, el artefacto central es un framework metodológico para la implementación de IA en el desarrollo de software. DSR resultó apropiado porque la problemática estudiada es socio-técnica, multidimensional y dinámica, y requiere integrar evidencia teórica, prácticas existentes y criterios normativos en una solución comunicable y aplicable (Peffer et al., 2007; Wieringa, 2014).

Como base de conocimiento se realizó una revisión narrativa ampliada de la literatura. La elección de este tipo de revisión respondió a su capacidad para sintetizar conocimiento heterogéneo proveniente de ingeniería de software, adopción tecnológica, gobernanza y ética de IA, sin restringir el estudio a la estimación cuantitativa de efectos (Grant & Booth, 2009). La revisión se orientó a identificar aplicaciones de IA por fase del SDLC, barreras técnicas, organizacionales y éticas, y frameworks existentes relevantes para el diseño del artefacto.

El procedimiento metodológico siguió cuatro etapas principales: identificación del problema y delimitación del dominio; definición de objetivos del artefacto; diseño y desarrollo del framework; y evaluación mediante validación conceptual. Esta última etapa incluyó tres estrategias complementarias: comparación estructurada con frameworks existentes, aplicación del framework a casos empíricos documentados en la literatura y análisis de coherencia interna. En línea con DSR, la validez del artefacto se sustentó en la trazabilidad entre problema, fundamentos teóricos, estructura del framework y mecanismos de evaluación (Hevner et al., 2004; Jabareen, 2009; Whetten, 1989).

4 Propuesta del framework FIIADS

4.1 Principios rectores

FIIADS se diseñó a partir de cuatro principios. El primero es la **integralidad**: la adopción de IA no puede reducirse a una decisión técnica, porque involucra capacidades organizacionales, prácticas de trabajo y criterios éticos. El segundo es la **transversalidad**: la IA puede aportar valor en todas las fases del SDLC y, por lo tanto, el framework debía contemplar actividades, riesgos y controles desde la planificación hasta el mantenimiento. El tercero es la **adaptabilidad**: el framework debería ser utilizable en organizaciones con diferentes niveles de madurez digital, metodologías de desarrollo y dominios de aplicación. El cuarto principio ubica la **ética y la responsabilidad** como núcleo, integrando desde el inicio estándares internacionales de IA responsable, en lugar de tratarlos como una capa posterior de cumplimiento (UNESCO, 2021; IEEE, 2020; European Union, 2024).

La selección de las dimensiones, procesos y principios de FIIADS no se realizó de manera arbitraria, sino a partir de la triangulación entre literatura sobre IA aplicada al SDLC, modelos de adopción tecnológica y marcos de gobernanza y ética de IA. La dimensión técnica deriva de los desafíos de integración, validación, deuda técnica y monitoreo de sistemas de IA; la dimensión organizacional se fundamenta en Technology–Organization–Environment (TOE) y Technology Acceptance Model (TAM), que explican la influencia de factores organizativos, humanos y contextuales en la adopción tecnológica; y la dimensión ética se apoya en marcos internacionales de IA responsable, que enfatizan transparencia, trazabilidad, equidad y supervisión humana. La Tabla 1 sintetiza esta trazabilidad entre fundamentos conceptuales y decisiones de diseño.

Tabla 1. Fundamentos conceptuales y decisiones de diseño de FIIADS.

Elemento de FIIADS	Fundamento conceptual	Decisión de diseño
Dimensión técnica	MLOps, deuda técnica en ML, validación y monitoreo de sistemas de IA	Incorporar procesos de identificación, factibilidad, selección, validación y monitoreo técnico.
Dimensión organizacional	TOE, TAM y literatura sobre adopción de IA en ingeniería de software	Incorporar madurez organizacional, gestión del cambio, capacitación, roles e integración en procesos.
Dimensión ética	UNESCO, IEEE, Trustworthy AI, AI Act y literatura sobre IA responsable	Incorporar identificación de riesgos, evaluación ética, salvaguardas, supervisión humana y monitoreo continuo.
Integralidad	Enfoques sociotécnicos y DSR	Evitar una adopción centrada solo en herramientas o desempeño técnico.
Transversalidad	Literatura sobre IA aplicada a fases del SDLC	Articular el framework con planificación, requerimientos, diseño, desarrollo, pruebas y mantenimiento.

Adaptabilidad	TOE, TAM y DSR	Permitir aplicación gradual según madurez digital, contexto organizacional y dominio.
Ética como núcleo	Marcos internacionales de gobernanza y ética de IA	Integrar criterios éticos desde el inicio y no como control posterior.

4.2 Estructura general

FIIADS organiza la aplicación de IA en tres dimensiones, técnica, organizacional y ética, y las articula con las seis fases del SDLC adoptadas en este estudio: planificación, requerimientos, diseño, desarrollo, pruebas y mantenimiento. Cada dimensión se operacionaliza mediante procesos específicos, prácticas recomendadas, artefactos de soporte, riesgos asociados y métricas de evaluación. Esta doble estructura, por dimensiones y por fases, como se describe en la Tabla 2, permite una aplicación holística y progresiva en distintos contextos organizacionales.

Tabla 2. Dimensiones y procesos del framework FIIADS.

Dimensión	Procesos	Propósito principal
Técnica	T1 Identificación de oportunidades; T2 Factibilidad; T3 Selección y adaptación; T4 Validación y calidad; T5 Monitoreo y operación	Asegurar viabilidad, integración, trazabilidad, calidad y monitoreo continuo de las soluciones de IA.
Organizacional	O1 Madurez organizacional; O2 Gestión del cambio y capacitación; O3 Redefinición de roles; O4 Integración en procesos organizativos	Preparar a la organización, institucionalizar el uso de IA y alinear su adopción con objetivos, capacidades y flujos de trabajo.
Ética	E1 Identificación de riesgos; E2 Evaluación ética; E3 Controles y salvaguardas; E4 Monitoreo ético continuo	Traducir principios de IA responsable en controles operativos, documentación trazable y supervisión humana.

Cada dimensión se operacionaliza mediante procesos específicos, sintetizados en la Tabla 2: cinco técnicos (T1-T5), cuatro organizacionales (O1-O4) y cuatro éticos (E1-E4). Su especificación complete, actividades, prácticas, riesgos y artefactos, se detalla en el material complementario (Paladino & Pons, 2026).

4.3 Articulación con el SDLC

La utilidad de FIIADS depende de su capacidad para orientar decisiones concretas a lo largo del SDLC. La Tabla 3 resume los procesos predominantes en cada fase del SDLC, así como su foco de implementación.

Tabla 3. Procesos predominantes del FIIADS por fase del SDLC.

Fase del SDLC	Procesos predominantes	Foco de implementación
---------------	------------------------	------------------------

Planificación	T1, T2, O1, O2, E1	Seleccionar casos de uso con valor real, evaluar viabilidad y preparar capacidades institucionales.
Requerimientos	T1, T2, T4, O2, E1, E2	Usar NLP o IA generativa con trazabilidad, validación experta y control de sesgos.
Diseño	T1, T3, T4, O3, E2, E3	Integrar herramientas de apoyo arquitectónico sin delegar decisiones críticas.
Desarrollo	T1, T3, T4, O2, O3, E1	Aprovechar asistentes de código con validación humana y criterios de uso responsable.
Pruebas	T1, T3, T4, O2, E3	Automatizar pruebas sin reemplazar la evaluación experta ni la trazabilidad de aceptación.
Mantenimiento	T1, T4, T5, O4, E4	Monitorear desempeño, drift, incidentes y alineación continua con políticas organizacionales.

4.4 Métricas y artefactos de evaluación

Un rasgo distintivo de FIIADS es que no se limita a describir principios o recomendaciones generales: incorpora métricas y artefactos explícitos. Entre los artefactos centrales se encuentran el mapa de oportunidades de IA, el informe de factibilidad técnica, el registro de herramientas y modelos seleccionados, la matriz de validación técnica, el informe de madurez organizacional, el plan de capacitación, el registro de riesgos éticos y el registro de incidentes éticos. Estos insumos permiten documentar decisiones, hacer trazables los supuestos y ofrecer evidencia para la mejora continua.

Del mismo modo, las métricas propuestas trascienden el rendimiento puramente técnico. Además de productividad, cobertura de pruebas, defectos e incidentes post-despliegue, el framework incorpora indicadores de preparación del equipo, nivel de adopción, satisfacción con las herramientas, cobertura de riesgos éticos identificados, decisiones críticas supervisadas por humanos y grado de cumplimiento de estándares éticos y de gobernanza. Esta amplitud evaluativa es central para evitar interpretaciones reduccionistas del impacto de la IA en el desarrollo de software.

Debido a restricciones de espacio, este artículo presenta únicamente una síntesis de los artefactos y métricas propuestos. Una especificación detallada de los procesos, métricas y artefactos de FIIADS está disponible como material complementario (Paladino & Pons, 2026), accesible en <https://www.doi.org/10.5281/zenodo.20977531>

5 Validación conceptual del framework

La validación de FIIADS se realizó a nivel conceptual, estrategia consistente con el alcance de la investigación y con la lógica de DSR para artefactos de tipo framework (Gregor & Hevner, 2013; Wieringa, 2014). En lugar de evaluar la implementación en

una organización específica, el estudio analizó la cobertura del framework, su coherencia interna y su aplicabilidad potencial. Para ello se combinaron tres estrategias: comparación con marcos existentes, aplicación a casos empíricos documentados en la literatura y análisis de coherencia interna, como se describe en las siguientes secciones.

5.1 Comparación con frameworks existentes

La comparación se realizó con tres familias de marcos frecuentemente citadas en la literatura: MLOps, el AI Risk Management Framework del NIST y los marcos éticos internacionales como UNESCO, IEEE y Trustworthy AI de la Unión Europea. El criterio de comparación consideró cobertura del SDLC, presencia de dimensiones técnica, organizacional y ética, grado de definición de procesos, existencia de métricas y artefactos, y capacidad de mejora continua. Los resultados confirman que FIIADS no reemplaza a estos marcos, sino que los integra y operacionaliza en un nivel específicamente orientado al desarrollo de software. La Tabla 4 resume la comparación con dichos marcos.

Tabla 4. Comparación conceptual entre FIIADS y marcos existentes.

Criterio	MLOps	NIST AI RMF	UNESCO/IEEE/UE	FIIADS
Cobertura del SDLC	Parcial	Transversal	Transversal	Completa
Dimensión técnica	Alta	Media	Baja	Alta
Dimensión organizacional	Media	Media	Media	Alta
Dimensión ética	Baja	Media	Alta	Alta y operacionalizada
Procesos definidos	Sí, técnicos	No específicos	No	Sí, T-O-E
Métricas y artefactos	Técnicos	Conceptuales	Normativos	Explícitos
Mejora continua	Alta	Alta	Recomendada	Integrada

En conjunto, la comparación confirma que FIIADS no reemplaza estos marcos sino que los integra y operacionaliza en el nivel específico del SDLC.

5.2 Aplicación a casos empíricos documentados

La segunda estrategia consistió en aplicar FIIADS a casos documentados en la literatura, con el objetivo de evaluar su capacidad explicativa y organizadora, y no su exhaustividad ni su representatividad estadística. Se seleccionaron seis casos ilustrativos mediante muestreo intencional, procurando cubrir las distintas fases del SDLC y representar enfoques diversos de IA: machine learning (ML), procesamiento de lenguaje natural e IA generativa mediante Large Language Models (LLMs). La correspondencia entre cada caso y los procesos del framework se estableció mediante

análisis conceptual, mapeando las actividades, los riesgos y las prácticas reportadas en cada estudio sobre las dimensiones técnica, organizacional y ética.

Tabla 5. Aplicación ilustrativa de FIIADS a casos documentados.

Fase	Caso documentado	Lectura desde FIIADS
Planificación	Pospieszny et al. (2018): estimación de esfuerzo con ML	Activa T1 y T2 para identificar valor predictivo y evaluar calidad/representatividad de datos; requiere O1, O2 y E1-E2.
Requerimientos	Ferrari et al. (2018): defectos en requisitos con NLP	Activa T1, T2 y T4; exige interpretación experta, integración con gestión de requisitos y control de sesgos o falsas detecciones.
Diseño	Silva et al. (2023): predicción de problemas de diseño	Activa T1, T3 y T4; demanda roles claros para la interpretación de recomendaciones y trazabilidad en decisiones arquitectónicas.
Desarrollo	Peng et al. (2023): GitHub Copilot y productividad	Activa T1, T3 y T4; refuerza O2-O3 y E1 por el riesgo de dependencia, errores o fragmentos inseguros.
Pruebas	Siddiq et al. (2024): generación de JUnit con LLMs	Activa T1, T3 y T4; evidencia que cobertura no equivale a calidad y que se requieren criterios humanos de validación.
Mantenimiento	Kamei et al. (2013): just-in-time quality assurance	Activa T1, T4 y T5; integra monitoreo continuo, revisión de desempeño y trazabilidad en la priorización de cambios.

Como muestra la Tabla 5, la aplicación resultó consistente en las seis fases: en cada una, FIIADS permitió ordenar procesos dominantes y complementarios y explicitar riesgos y necesidades de supervisión que no siempre estaban formulados en los estudios originales analizados. Por ejemplo, en codificación y testing destacó T4 como proceso transversal de validación, y en planificación y mantenimiento, el peso de T2 y T5 frente al drift y a los cambios del contexto operativo. La lectura organizacional y ética, además, mostró que incluso las aplicaciones técnicamente exitosas requieren capacidades institucionales, roles definidos y salvaguardas de gobernanza.

5.3 Análisis de coherencia interna

La tercera estrategia de validación examinó la coherencia interna del artefacto. Siguiendo a Whetten (1989), una contribución conceptual sólida debe responder con claridad a qué componentes incluye, cómo se relacionan entre sí y por qué esas relaciones son significativas. Asimismo, Jabareen (2009) destaca que un framework consistente debe exhibir definiciones claras, ausencia de solapamientos improductivos y articulación explícita entre conceptos. Bajo esos criterios, FIIADS mostró consistencia entre principios rectores, dimensiones, procesos, artefactos, métricas y fases del SDLC. En particular, la integralidad se refleja en la incorporación conjunta de dimensiones técnicas, organizacionales y éticas; la transversalidad en la aplicación

de procesos a lo largo de todas las fases del ciclo de vida; la adaptabilidad en la posibilidad de ajustar herramientas, métricas y procedimientos a diferentes contextos organizacionales; y la orientación ética y responsable en la inclusión explícita de procesos destinados a la identificación, evaluación, mitigación y monitoreo de riesgos asociados al uso de IA.

Desde la perspectiva de DSR, el análisis también verificó la trazabilidad entre el problema inicial, la fragmentación de enfoques para adoptar IA en el SDLC, y la solución propuesta. La literatura identificó barreras técnicas, organizacionales y éticas; el framework respondió a ellas con tres dimensiones diferenciadas; y la evaluación conceptual comprobó que la estructura resultante es interpretable, aplicable y suficientemente operacionalizada para orientar futuras implementaciones (Hevner et al., 2004; Gregor & Hevner, 2013; Wieringa, 2014). En síntesis, la validación conceptual no demuestra todavía efectividad empírica, pero sí aporta fundamento teórico, cobertura del dominio y plausibilidad de uso.

6 Discusión

Los resultados de esta investigación muestran que la principal contribución de FIIADS es desplazar la discusión sobre IA en ingeniería de software desde un enfoque centrado en herramientas aisladas hacia una perspectiva socio-técnica y evaluativa. En lugar de preguntar solamente qué puede automatizar la IA, el framework obliga a preguntarse bajo qué condiciones la adopción es técnicamente viable, organizacionalmente sostenible y éticamente aceptable. Esta ampliación es particularmente relevante en el contexto actual de la IA generativa, donde la presión por adoptar herramientas de alto impacto en productividad puede llevar a decisiones apresuradas, dependencias no controladas o deterioro de la calidad del software si no existen procesos explícitos de validación y gobernanza (Russo, 2024; Fu et al., 2025).

Desde el punto de vista metodológico, la investigación también muestra el valor de Design Science Research para abordar problemas emergentes en ingeniería de software. FIIADS fue construido a partir de evidencia científica actualizada, articulado en una estructura conceptual clara y examinado mediante una validación conceptual sistemática. Esto aporta un ejemplo concreto de cómo DSR puede utilizarse para producir conocimiento útil y comunicable cuando los fenómenos estudiados son demasiado recientes o complejos para ser capturados inicialmente por diseños experimentales tradicionales.

No obstante, el trabajo reconoce límites importantes. La validación realizada es conceptual y no incluye todavía la implementación del framework en organizaciones reales. Por ello, no es posible medir empíricamente su impacto sobre productividad, calidad, reducción de riesgos o aceptación organizacional. Asimismo, no se realizaron pruebas de concepto completas ni evaluaciones sistemáticas con expertos, las cuales podrían aportar evidencia adicional sobre la utilidad, aplicabilidad y facilidad de adopción del framework en distintos contextos. Además, el nivel de abstracción deliberado que le da flexibilidad también implica que cada organización deberá adaptar herramientas, métricas y procedimientos a su dominio específico. Si bien

FIIADS fue concebido para ser independiente de la metodología de desarrollo utilizada, el framework no explora de manera explícita su integración con enfoques Development and Operations (DevOps), Business–Development–Operations (BizDevOps) o MLOps, donde la incorporación de IA puede impactar procesos de integración continua, despliegue continuo, monitoreo operacional y automatización de pipelines. La realización de estudios de caso, pruebas de concepto, evaluaciones con expertos y la articulación específica con estos enfoques constituyen líneas prioritarias de trabajo futuro. Finalmente, el ritmo de evolución de la IA, especialmente de los modelos generativos, exige que el framework sea revisado y actualizado periódicamente para mantener su vigencia.

6.1 Implicancias para la práctica profesional

Desde la práctica profesional, FIIADS ofrece una ventaja concreta: permite desplazar la adopción de IA desde decisiones ad hoc, guiadas por la novedad tecnológica o por la presión competitiva, hacia un proceso de implementación progresivo, documentado y evaluable. En lugar de incorporar asistentes de código, herramientas de análisis o modelos generativos como soluciones aisladas, el framework propone comenzar por la identificación de oportunidades de valor, evaluar la factibilidad técnica y organizacional, definir roles claros, implementar salvaguardas y monitorear el desempeño de manera sostenida. Esta secuencia puede ayudar a reducir costos de ensayo y error, a mejorar la gobernanza interna y a generar evidencia útil para decidir cuándo escalar, ajustar o incluso discontinuar una herramienta de IA. Además, la articulación con métricas y artefactos facilita la trazabilidad y fortalece el alineamiento con estándares emergentes de gestión y cumplimiento, como ISO/IEC 42001 o marcos de evaluación de riesgo centrados en confiabilidad y supervisión humana (International Organization for Standardization, 2023; NIST, 2023).

6.2 Agenda de investigación y validación empírica

Desde la investigación académica, el principal desafío es avanzar desde la validez conceptual hacia la validez empírica. Ello supone estudiar cómo responde FIIADS en organizaciones con distintos niveles de madurez digital, dominios regulados, restricciones presupuestarias o equipos distribuidos. También abre la posibilidad de operacionalizar variables comparables para analizar el impacto de la IA no solo en productividad, sino en calidad del software, coordinación del trabajo, aprendizaje organizacional, cumplimiento ético y percepción de utilidad por parte de los desarrolladores. En este sentido, el framework puede funcionar tanto como guía de implementación como estructura analítica para estudios de caso, diseños longitudinales o investigaciones mixtas. Su aporte, por lo tanto, no se limita a proponer una arquitectura conceptual, sino que también ofrece un lenguaje común para futuras evaluaciones comparativas sobre adopción responsable de IA en ingeniería de software.

Asimismo, resulta de interés explorar la integración de FIIADS con entornos DevOps, BizDevOps y MLOps mediante mecanismos automatizados de monitoreo,

trazabilidad y auditoría continua, evaluando cómo estas capacidades pueden fortalecer la gobernanza y el seguimiento operativo de sistemas basados en IA.

7 Conclusiones

Este trabajo presentó FIIADS, un framework holístico para la implementación de inteligencia artificial en el desarrollo de software, concebido como respuesta a una brecha persistente en la literatura: la falta de marcos que articulen de manera explícita y operativa las dimensiones técnica, organizacional y ética a lo largo de todo el ciclo de vida del desarrollo. Bajo el paradigma Design Science Research, FIIADS organiza procesos, métricas y artefactos de soporte para orientar decisiones de adopción, integración, evaluación y seguimiento de IA en las seis fases del SDLC, asumiendo la ética y la supervisión humana como núcleo del diseño y no como una capa posterior de cumplimiento.

La validación conceptual, sustentada en la comparación con marcos existentes, la aplicación a casos documentados y el análisis de coherencia interna, sugiere que FIIADS ofrece una cobertura más amplia y una operacionalización más específica para el SDLC que los enfoques disponibles y que, en los casos analizados, su estructura facilitó la identificación de procesos, riesgos y controles que no estaban explicitados en los estudios originales. No obstante, su alcance es deliberadamente conceptual: aporta fundamento teórico, cobertura del dominio y plausibilidad de uso, pero no demuestra todavía efectividad empírica. Esta distinción delimita con honestidad lo que el framework permite afirmar en su estado actual y, a la vez, define su agenda de consolidación.

El paso siguiente es avanzar desde la validez conceptual hacia la validez empírica mediante estudios de caso, evaluaciones con expertos y diseños longitudinales en organizaciones con distintos niveles de madurez digital y dominios de aplicación, así como profundizar su articulación con estándares emergentes como ISO/IEC 42001 y con entornos DevOps, BizDevOps y MLOps. Más allá de proponer una taxonomía de procesos, el principal aporte de FIIADS reside en ofrecer una lógica de implementación que ayuda a incorporar la IA como una capacidad organizacional gobernada, evaluable y responsable, y no como una herramienta puntual. En ese sentido, el framework busca tender un puente entre la promesa tecnológica de la inteligencia artificial y las exigencias concretas de la ingeniería de software contemporánea.

Referencias

- Ali, A. & Gravino, C. (2019). A systematic literature review of software effort prediction using machine learning methods. *Journal of Software: Evolution and Process*, 31(10), e2211. <https://doi.org/10.1002/smr.2211>
- Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., et al. (2019). Software engineering for machine learning: A case study. *Proceedings of the 41st International Conference on Software Engineering*, 291–300. <https://doi.org/10.1109/ICSE-SEIP.2019.00042>

- Breck, E., Cai, S., Nielsen, E., Salib, M., & Sculley, D. (2017). The ML test score: A rubric for ML production readiness and technical debt reduction. *Proceedings of the IEEE International Conference on Big Data*, 1123–1132. <https://doi.org/10.1109/BigData.2017.8258038>
- Cheng, H., Husen, J. H., Lu, Y., Racharak, T., Yoshioka, N., Ubayashi, N., & Washizaki, H. (2026). Generative AI for Requirements Engineering: A systematic literature review. *Software: Practice and Experience*, 56(2), 141–170. <https://doi.org/10.1002/spe.70029>
- Dignum, V. (2019). *Responsible artificial intelligence: How to develop and use AI in a responsible way*. Springer. <https://doi.org/10.1007/978-3-030-30371-6>
- European Commission High-Level Expert Group on Artificial Intelligence. (2019). *Ethics guidelines for trustworthy AI*. European Commission.
- European Union. (2024). Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act).
- Fan, A., Gokkaya, B., Harman, M., Lyubarskiy, M., Sengupta, S., Yoo, S., & Zhang, J. M. (2023). Large language models for software engineering: Survey and open problems. *arXiv preprint arXiv:2310.03533*. <https://doi.org/10.1109/ICSE-FoSE59343.2023.00008>
- Ferrari, A., Gori, G., Rosadini, B., Trotta, I., Bacherini, S., Fantechi, A., & Gnesi, S. (2018). Detecting requirements defects with NLP patterns: An industrial experience in the railway domain. *Empirical Software Engineering*, 23(6), 3684–3733. <https://doi.org/10.1007/s10664-018-9596-7>
- Floridi, L., Cowls, J., Beltrametti, M., Chatila, R., Chazerand, P., Dignum, V., et al. (2018). AI4People-An ethical framework for a good AI society. *Minds and Machines*, 28(4), 689–707. <https://doi.org/10.1007/s11023-018-9482-5>
- Fu, Y., Liang, P., Tahir, A., Li, Z., Shahin, M., Yu, J., & Chen, J. (2025). Security weaknesses of Copilot-generated code in GitHub projects: An empirical study. *ACM Transactions on Software Engineering and Methodology*. <https://doi.org/10.1145/3716848>
- Grant, M. J., & Booth, A. (2009). A typology of reviews: An analysis of 14 review types and associated methodologies. *Health Information & Libraries Journal*, 26(2), 91–108. <https://doi.org/10.1111/j.1471-1842.2009.00848.x>
- Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly*, 37(2), 337–355. <https://doi.org/10.25300/MISQ/2013/37.2.01>
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105. <https://doi.org/10.2307/25148625>
- IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems. (2020). *Ethically aligned design: A vision for prioritizing human well-being with autonomous and intelligent systems* (2nd ed.). IEEE Standards Association.
- International Organization for Standardization. (2023). *ISO/IEC 42001:2023. Artificial intelligence - Management system*.
- Jabareen, Y. (2009). Building a conceptual framework: Philosophy, definitions, and procedure. *International Journal of Qualitative Methods*, 8(4), 49–62. <https://doi.org/10.1177/160940690900800406>
- Kamei, Y., Shihab, E., Adams, B., Hassan, A. E., Mockus, A., Sinha, A., & Ubayashi, N. (2013). A large-scale empirical study of just-in-time quality assurance. *IEEE Transactions on Software Engineering*, 39(6), 757–773. <https://doi.org/10.1109/TSE.2012.70>
- Kreuzberger, D., Köhl, N., & Hirschl, S. (2023). Machine Learning Operations (MLOps): Overview, definition, and architecture. *IEEE Access*, 11, 31866–31879. <https://doi.org/10.1109/ACCESS.2023.3262138>
- Liao, Q. V., & Wortman Vaughan, J. (2024). AI transparency in the age of LLMs: A human-centered research roadmap. *Harvard Data Science Review*, Special Issue 5. <https://doi.org/10.1162/99608f92.8036d03b>

- National Institute of Standards and Technology. (2023). Artificial intelligence risk management framework (AI RMF 1.0). U.S. Department of Commerce.
- Paladino, M., & Pons, C. (2025). From Obstacles to Strategies: Integrating Artificial Intelligence into the Software Development Life Cycle. 54JAIIO. Jornadas Argentinas de Informática, 11(2), 101–117.
- Paladino, M., & Pons, C. (2026). Material complementario de FIIADS: un framework holístico para la implementación de inteligencia artificial en el desarrollo de software (documento de especificación). (Versión: 1.0 - 25/6/2026). Zenodo.
<https://doi.org/10.5281/zenodo.20977531>
- Peppers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77. <https://doi.org/10.2753/MIS0742-1222240302>
- Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). The impact of AI on developer productivity: Evidence from GitHub Copilot. arXiv:2302.06590.
- Pospieszny, P., Czarnacka-Chrobot, B., & Kobylinski, A. (2018). An effective approach for software project effort and duration estimation with machine learning algorithms. *Journal of Systems and Software*, 137, 184–196. <https://doi.org/10.1016/j.jss.2017.11.066>
- Russo, D. (2024). Navigating the complexity of generative AI adoption in software engineering. *ACM Transactions on Software Engineering and Methodology*.
<https://doi.org/10.1145/3652154>
- Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., et al. (2015). Hidden technical debt in machine learning systems. *Advances in Neural Information Processing Systems*, 28, 2503–2511.
- Siddiq, M. L., Santos, J. C. S., Tanvir, R. H., Ulfat, N., Al Rifat, F., & Lopes, V. C. (2024). Using large language models to generate JUnit tests: An empirical study. *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*, 313–322. <https://doi.org/10.1145/3661167.3661216>
- Silva, R. K., Farias, K., Kunst, R., & Dalzochio, J. (2023). An approach based on machine learning for predicting software design problems. *Proceedings of the XIX Brazilian Symposium on Information Systems*. <https://doi.org/10.1145/3592813.3592888>
- Tornatzky, L., & Fleischer, M. (1990). *The processes of technological innovation*. Lexington Books.
- UNESCO. (2021). *Recommendation on the ethics of artificial intelligence*. United Nations Educational, Scientific and Cultural Organization.
- Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science*, 46(2), 186–204.
<https://doi.org/10.1287/mnsc.46.2.186.11926>
- Whetten, D. A. (1989). What constitutes a theoretical contribution? *Academy of Management Review*, 14(4), 490–495. <https://doi.org/10.2307/258554>
- Wieringa, R. (2014). *Design science methodology for information systems and software engineering*. Springer. <https://doi.org/10.1007/978-3-662-43839-8>
- Zhang, Q., Fang, C., Xie, Y., Zhang, Y., Yang, Y., Sun, W., Yu, S., & Chen, Z. (2023). A survey on large language models for software engineering. arXiv:2312.15223.
- Zhao, L., Alhoshan, W., Ferrari, A., Letsholo, K. J., Ajagbe, M. A., Chioasca, E.-V., & Batista-Navarro, R. T. (2021). Natural language processing for requirements engineering: A systematic mapping study. *ACM Computing Surveys*, 54(3), Article 55.
<https://doi.org/10.1145/3444689>