



Universidad Nacional



Sistemas de Estudios de Posgrado

Maestría en Tecnologías de la Información (MATI)

Énfasis en gestión de proyectos

Proyecto Final de Graduación

**Adopción estratégica de la Inteligencia Artificial en
fábricas de software.**

Autor:

Ricardo Villalobos Gonzalo

Heredia, Costa Rica, 2025

Índice General

Índice General	2
Índice de Tablas	6
Índice de Figuras	7
Índice de anexos	8
Dedicatoria	9
Resumen ejecutivo	11
CAPÍTULO I. El problema y su importancia	13
A) Antecedentes	14
A) Justificación	15
B) Problema	16
C) Objetivos	16
1) Objetivo General	16
2) Objetivos Específicos	17
D) Metas por objetivo	18
CAPÍTULO II. Marco teórico o conceptual	20
A) Gestión de proyectos de software	21
B) Enfoques de gestión de proyectos: tradicional, ágil e híbrido	23
C) Metodología ágil: Scrum y Sprint	25
D) Fábricas de software y desarrollo de software	27
E) Calidad y mejora de procesos: CMMI, DevOps y métricas DORA	29
F) Inteligencia Artificial y aprendizaje automático en el desarrollo de software	33
G) Ética de la Inteligencia Artificial	36
H) Necesidad de un modelo de gestión para la adopción estratégica de IA	38

I)	Definición académica de prompt	41
J)	Definición técnica de <i>prompt</i>	41
CAPÍTULO III. Marco Metodológico		43
A)	Enfoque de la Investigación	44
B)	Sujetos y fuentes de información	45
1)	Sujetos	45
2)	Fuentes de Información	46
C)	Instrumentos y técnicas de recolección	47
1)	Procedimientos	49
CAPÍTULO IV. Diagnóstico y análisis de resultado		58
A)	Abstracts de los Artículos Seleccionados	59
1)	Venkatesh (2021) – Adoption and Use of AI Tools: A Research Agenda Grounded in UTAUT	59
2)	Kauppaymuthoo & Collins (2023) – Artificial Intelligence Adoption in Software Development: A Qualitative Study on Developers' Views	59
3)	Carrillo Zenteno et al. (2024) – The Impact of Artificial Intelligence on Project Management	60
4)	Wu et al., 2022 Sustainable AI: Environmental Implications, Challenges and Opportunities	60
5)	Ligozat et al., 2022 – “Unraveling the Hidden Environmental Impacts of AI Solutions”	61
6)	Nishant, Kennedy & Corbett, 2020 – “Artificial Intelligence for Sustainability: Challenges, Opportunities and a Research Agenda”	61
B)	Resultados de la Revisión de la Literatura	62
1)	Factores críticos de adopción	62
2)	Beneficios tangibles	62
3)	Barreras persistentes	63
4)	Análisis del impacto de la AI en el medio ambiente	63

5)	Contribución al modelo de gestión propuesto.....	63
C)	Resultados del Cuestionario.....	64
1)	Perfil de los participantes.....	64
2)	Sección I - Prácticas actuales de IA	65
3)	Sección II - Barreras y facilitadores.....	68
4)	Sección III – Impacto y métricas	74
5)	Sección IV – IA sostenible / Impacto ambiental	79
D)	Alfa de Cronbach	88
1)	Análisis.....	88
2)	Fórmula del Alfa de Cronbach	88
3)	Interpretación	90
E)	Principales Hallazgos de la Encuestas.....	90
1)	Prácticas actuales de IA.....	90
2)	Barreras y facilitadores	91
3)	Impacto y métricas.....	91
4)	IA sostenible / Impacto ambiental.....	91
	CAPÍTULO V. Solución del Problema	93
A)	Propuesta de solución	94
B)	Desarrollo de la solución	95
	Modelo de gestión para la adopción estratégica de IA en fábricas de software.....	95
	Fase 1: Diagnóstico inicial y línea base	99
	Fase 2: Planificación estratégica de la adopción de IA.....	103
	Fase 3: Pilotos controlados.....	110
	Fase 4: Integración y escalamiento	119
	Fase 5: Operación sostenible y mejora continua.....	127
	Dimensión opcional: Sostenibilidad ambiental (“capa verde”).....	136
C)	Procedimiento de implementación	139
1)	Prototipo.....	139

2) Criterio de expertos.....	142
CAPÍTULO VI. Análisis financiero.....	145
A) Costos	146
B) Parámetros.....	148
C) Flujo de Caja	149
CAPÍTULO VII. Conclusiones y Recomendaciones	151
A) Conclusiones.....	152
B) Recomendaciones	153
CAPÍTULO VIII. Análisis retrospectivo.....	156
Referencias Bibliográficas	158
Anexos	164

Índice de Tablas

Tabla 1. Metas por objetivos	19
Tabla 2. Generación de Generación de Prompts	57
Tabla 3. Tabulación de respuestas	Error! Bookmark not defined.
Tabla 4. Resultados.....	Error! Bookmark not defined.
Tabla 5. Interpretación	Error! Bookmark not defined.
Tabla 6. Comentarios o Sugerencias Gustavo Conejo Madrigal.....	143
Tabla 7. Costo RRHH.....	146
Tabla 8. Costo Operación	147
Tabla 9. Parámetros.....	148
Tabla 10. Flujo de Caja.....	150

Índice de Figuras

Figura 1. ¿Qué herramientas o tecnologías de IA utilizan hoy en el ciclo de vida del software?	65
Figura 2. Frecuencia promedio de uso de IA en el trabajo diario.....	66
Figura 3. Etapas del ciclo de desarrollo donde se integra la IA	67
Figura 4. Obstáculos principales para adoptar IA	69
Figura 5. Factores que más han facilitado la adopción	70
Figura 6. Nivel de apoyo institucional a las iniciativas de IA.....	72
Figura 7. ¿La organización cuenta con políticas o lineamientos sobre uso responsable/ético de IA?.....	73
Figura 8. Beneficios tangibles percibidos tras adoptar IA.....	75
Figura 9. Métricas en las que ha observado mejora gracias a IA.....	76
Figura 10. Impacto global de la IA en el rendimiento del ciclo de desarrollo de software.....	78
Figura 11. ¿Su organización mide o monitorea el impacto ambiental (energía, carbono, agua) derivado del uso de IA?.....	80
Figura 12. ¿Qué acciones concretas han implementado para reducir la huella ambiental de la IA?	81
Figura 13. Al contratar proveedores de servicios de IA, ¿evalúan criterios de sostenibilidad?.....	82
Figura 14. ¿Algún cliente les ha solicitado que las soluciones de IA sean carbono-neutras?	84
Figura 15. Frente a esas solicitudes (o futuras), ¿Cómo respondería su fábrica de software?.....	85
Figura 16. ¿Qué importancia estratégica le otorga su organización a la sostenibilidad ambiental de la IA?	87
Figura 17. Fórmula Alfa de Cronbach.....	88
Figura 18. Tabulación de respuestas.....	89
Figura 19. Resultados	89
Figura 20. Interpretación	90
Figura 21. Mapa de Procesos (nivel 0).....	94
Figura 22. Diagrama del modelo de adopción estratégica de IA en fábricas de software	98

Índice de anexos

A) Anexo 1: Cuestionario: Adopción estratégica de la Inteligencia Artificial en fábricas de software.....	164
B) Anexo 3: Kit de adopción de la solución.....	171
C) Anexo 4: Piloto Modelo de Adopción	171
D) Análisis Financiero	171

Dedicatoria

Agradezco infinitamente a Dios por la vida, por la salud, por la energía, por la sabiduría y por la alegría de poder estudiar y seguir creciendo en cada etapa. Nada de esto tendría sentido sin esa fuerza espiritual que me sostiene y me da esperanza incluso en los días más difíciles.

A mi madre Patricia, gracias por nunca dejar de creer en mí. Esta maestría de la UNA es también un regalo para vos; un sueño compartido que hoy nos une aún más, porque ahora podemos decir con orgullo que somos egresados de la misma universidad. Tu ejemplo, tu fe, tu soporte y tu amor han sido mi base siempre para lograr grandes cosas.

A mi esposa Melissa y a mis hijas Sara y Camila, que son mi motor y mi inspiración diaria. Gracias por su paciencia, por acompañarme en este proceso, por entender los sacrificios y por recordarme cada día por qué vale la pena dar lo mejor de mí. Todo lo que hago también es por el amor que les tengo.

A mi hermana Rebeca, por su apoyo incondicional; a mi sobrina María Celeste y a mi cuñado Malber, por estar siempre presentes, por su cariño y por ser una fuente constante de motivación. Gracias por empujarme con palabras, con gestos y con ese "aquí estamos" que tanto significa.

A mi padre Ricardo, gracias por alegrarte con mis logros y por sentirti orgulloso de mí. Tu orgullo lo llevo como una fuerza silenciosa que me acompaña en cada meta.

A mi hermano de otra madre, Oscar, gracias de corazón. Fuiste quien me impulsó y casi que me obligó a estudiar cuando más lo necesitaba. Siempre estuviste, estás y sé que siempre estarás para celebrar mis victorias como si fueran tuyas. Tu lealtad y tu confianza han sido un regalo enorme en mi vida.

Y finalmente, al profesor Harold, gracias por su sabiduría, por su solidez profesional y por dar cátedra, con su ejemplo, de cómo enfrentarse a la vida en el ámbito profesional. Su guía ha dejado huella y se queda conmigo más allá de este trabajo.

A todos, gracias por ser parte de este camino. Este logro también les pertenece.

Resumen ejecutivo

Este trabajo final de graduación nace de una realidad muy concreta: la inteligencia artificial ya está entrando con fuerza en las fábricas de software, pero en la mayoría de los casos lo hace de forma dispersa, por iniciativas aisladas y sin un camino claro que garantice valor, control y sostenibilidad. Ante ese panorama, el propósito central de la investigación fue desarrollar un modelo de gestión que guíe la adopción estratégica de IA en fábricas de software, integrando buenas prácticas de la literatura y criterios de innovación, calidad y generación de valor, y dejarlo validado como una propuesta viable para el sector.

Para lograrlo, el estudio se apoyó en tres pilares: revisión del estado del arte, diagnóstico de prácticas reales mediante cuestionario a perfiles con experiencia en dirección y entrega de software, y un juicio de expertos para validar la propuesta. Esa combinación permitió no solo entender qué se está usando hoy, sino también identificar barreras persistentes, facilitadores, beneficios percibidos y el nivel de preparación organizacional. Con esa información se diseñó un modelo que responde al problema de fondo: pasar de “probar IA” a adoptarla con disciplina, alineada al negocio y con criterios de éxito verificables.

El resultado principal es un modelo por fases, pensado para que la adopción sea gradual y medible. Arranca con un diagnóstico y línea base para entender madurez, cultura y oportunidades; sigue con una planificación estratégica donde se definen casos de uso, métricas y gobierno; continúa con pilotos controlados que validan valor en pequeño alcance antes de invertir fuerte; luego plantea integración y escalamiento de lo que funciona; y cierra con una operación sostenible basada en mejora continua. La lógica del modelo es simple pero potente: primero gobernanza y criterios de uso responsable,

después herramientas; y todo el tiempo midiendo impacto con métricas técnicas y con indicadores de negocio. De forma opcional, el modelo incorpora una “capa verde” que introduce criterios ambientales desde la planificación, para que la IA también sea sostenible en términos de energía y huella de carbono.

Además de la propuesta metodológica, el trabajo deja claro que el valor de la IA no se sostiene solo con entusiasmo: se sostiene con evidencia. Por eso el plan piloto propuesto describe etapas, roles, recursos y métricas de éxito para que cualquier fábrica pueda probar el modelo en un entorno real, aprender rápido y decidir con base en resultados si escala o ajusta.

En conjunto, la investigación aporta una hoja de ruta práctica para reducir retrabajos, mejorar calidad, acelerar entregas y evitar el riesgo de implementaciones ad hoc que luego no se pueden repetir ni gobernar.

CAPÍTULO I. El problema y su importancia

A) Antecedentes

En los últimos años la inteligencia artificial (IA) ha pasado de ser una curiosidad académica para impulsar cambios sustanciales en la industria del software. Encuestas globales recientes muestran un crecimiento explosivo de herramientas de IA generativa; por ejemplo, en 2023 un tercio de las organizaciones declara utilizar IA generativa regularmente en al menos una función (McKinsey Global Survey, 2023), proporción que se duplicó en 2024 (65% de las empresas) al derivar valor tangible de estas tecnologías (McKinsey Global Survey, 2024). Este auge se refleja en estrategias corporativas: un 40% de las empresas planea aumentar su inversión en IA (McKinsey Global Survey, 2023), y los líderes empresariales atribuyen ya el 20% o más de su EBIT a la adopción de IA avanzada. A nivel de prácticas de ingeniería, la IA potencia la generación automática de código (p. ej. asistentes como GitHub Copilot), pruebas inteligentes y la optimización de pipelines CI/CD mediante analítica predictiva. Un estudio de campo demostró que desarrolladores que emplean Copilot incrementan en ~26% sus entregas de código (InfoQ, 2023), lo que ilustra cómo las herramientas de IA mejoran la productividad en Java, .NET y otros entornos comunes. Además, las tecnologías basadas en IA atienden retos típicos, según una encuesta de Forbes Advisor, el 64% de las empresas creen que la inteligencia artificial desempeñará un papel fundamental en la elevación de su productividad general, y cerca del 25% ya usa IA para compensar la escasez de talento cualificado en áreas de análisis y soporte (Baig, 2023). Sin embargo, estos avances también exponen brechas críticas: menos de la mitad de las organizaciones mitiga riesgos fundamentales de la IA (p. ej. imprecisión) (McKinsey Global Survey, 2023), y muchas aún carecen de procesos maduros para integrar estas herramientas. En síntesis, el panorama muestra que la IA puede acelerar el time-to-market y reducir costes (mediante código asistido y

validación automatizada), pero solo si se aplican buenas prácticas consolidadas. Por ello, es necesario evaluar la madurez organizacional mediante marcos reconocidos (CMMI, métricas DORA) e incorporar recomendaciones específicas (p. ej. infraestructura en AWS/Azure para ML, pruebas automatizadas basadas en IA) para cerrar brechas actuales.

A) Justificación

En el contexto global y temporal 2025-2026, la adopción de IA es ya una prioridad estratégica. Gartner reporta que 87% de los CEOs opinan que los beneficios de la IA superan a sus riesgos, y que la mención de IA como prioridad tecnológica clave saltó del 4% al 24% en 2024 (Gartner, 2024). Estas cifras reflejan que las organizaciones líderes están reconfigurando sus estrategias para integrar IA y buscar un crecimiento sostenido. La literatura indica además beneficios concretos: adopciones tempranas ya reportan reducciones de costo y aumentos de ingreso en áreas donde se desplegó IA (McKinsey Global Survey, 2024). En este escenario, las fábricas de software deben prepararse para no quedarse atrás. Un marco metodológico de adopción de IA aportará impactos positivos operativos y estratégicos: disminuirá retrabajos, mejorará la calidad del software (mediante pruebas inteligentes) y acortará los ciclos de desarrollo (optimización CI/CD). Además, generará valor competitivo al acelerar la innovación de productos de software a través de prácticas avanzadas. En suma, la creciente aceptación de IA en el ámbito empresarial y tecnológico mundial justifica la creación de una guía práctica que facilite su incorporación ordenada en las fábricas de software, maximizando beneficios e impactando positivamente en la industria (McKinsey Global Survey, 2023).

B) Problema

A pesar del potencial de la IA, las fábricas de software actuales carecen de un marco metodológico que se haya popularizado o estandarizado para adoptarla de forma estructurada. Como resultado, las iniciativas de IA suelen ser ad hoc y poco replicables. Esta carencia se refleja en indicadores globales: menos de la mitad de las empresas declara tener mitigado el riesgo principal de la IA (la imprecisión) (McKinsey Global Survey, 2023), lo que indica gobernanza inmadura. La falta de un proceso definido provoca efectos negativos: retrabajo en desarrollo por integración deficiente de IA, extensos tiempos de entrega (time-to-market prolongado) y dependencia de talento externo costoso. Además, las organizaciones prevén recortes de personal y la necesidad de reentrenamiento masivo del equipo técnico (McKinsey Global Survey, 2023), evidencia de brechas en habilidades. En este contexto, sin un marco sistemático de adopción de IA ni criterios de madurez (CMMI/DORA) adaptados, las empresas limitan su innovación interna y corren riesgo de rezagarse ante competidores internacionales (McKinsey Global Survey, 2024). Por tanto, es importante abordar la causa raíz: definir un modelo metodológico que alinee objetivos de negocio con tecnología IA, para así corregir estos efectos adversos.

C) Objetivos

1) Objetivo General

Desarrollar un modelo de gestión para la adopción estratégica de la inteligencia artificial (IA) en fábricas de software, integrando las mejores

prácticas identificadas en la literatura especializada y principios de innovación, calidad y valor agregado, con el fin de guiar a las fábricas de software en la implementación eficaz de soluciones de IA que optimicen sus procesos de desarrollo de software. Meta (Indicador de logro): Modelo de gestión propuesto documentado y validado por expertos del sector, demostrando su viabilidad y eficacia en el contexto de fábricas de software.

2) Objetivos Específicos

- 2.1 Investigar y analizar el estado del arte de la adopción estratégica de la IA en fábricas de software, mediante la revisión de la literatura, con el fin de identificar las mejores prácticas, desafíos y oportunidades clave que sirvan de fundamento para el desarrollo del modelo de gestión propuesto.
- 2.2 Diseñar un modelo de gestión para la adopción estratégica de la IA en fábricas de software, integrando las mejores prácticas y recomendaciones identificadas en la revisión bibliográfica e incorporando principios de innovación, mejora de la calidad y valor agregado, con el propósito de estructurar un enfoque práctico que facilite la implementación eficaz de soluciones de IA en los procesos de desarrollo de software de dichas organizaciones.
- 2.3 Validar el modelo de gestión propuesto mediante juicio de expertos, evaluando su eficacia, pertinencia y aplicabilidad en el contexto real de fábricas de software, con el objetivo de asegurar que el modelo cumpla con los estándares y expectativas de la industria de TI en materia de innovación, calidad y valor agregado.
- 2.4 Validar la viabilidad de un plan piloto para implementar el modelo de gestión previamente diseñado para la adopción estratégica de IA en un entorno real de fábrica de software. Esta propuesta detallará las etapas,

recursos necesarios, responsables y métricas de éxito del piloto, con el fin de evaluar la eficacia y viabilidad del modelo en la práctica.

D) Metas por objetivo

#	Objetivo específico	Meta por cumplir
2.1	Investigar y analizar el estado del arte de la adopción estratégica de la IA en fábricas de software, mediante la revisión de la literatura, con el fin de identificar las mejores prácticas, desafíos y oportunidades clave que sirvan de fundamento para el desarrollo del modelo de gestión propuesto.	Revisión bibliográfica que sintetice las mejores prácticas, desafíos y oportunidades identificados para la adopción de IA en fábricas de software.
2.2	Diseñar un modelo de gestión para la adopción estratégica de la IA en fábricas de software, integrando las mejores prácticas y recomendaciones identificadas en la revisión bibliográfica e incorporando principios de innovación, mejora de la calidad y valor agregado, con el propósito de estructurar un enfoque práctico que facilite la implementación eficaz de soluciones de IA en los procesos de	Modelo de gestión diseñado y documentado en detalle, incluyendo sus componentes clave (fases, actividades, roles, métricas) e integrando los principios de innovación, calidad y valor agregado identificados previamente.

	desarrollo de software de dichas organizaciones	
2.3	Validar el modelo de gestión propuesto mediante juicio de expertos, evaluando su eficacia, pertinencia y aplicabilidad en el contexto real de fábricas de software, con el objetivo de asegurar que el modelo cumpla con los estándares y expectativas de la industria de TI en materia de innovación, calidad y valor agregado.	Evaluación experta realizada con especialistas del sector, obteniendo un consenso favorable y viabilidad del modelo en entornos de fábricas de software.
2.4	Validar la viabilidad de un plan piloto para implementar el modelo de gestión previamente diseñado para la adopción estratégica de IA en un entorno real de fábrica de software. Esta propuesta detallará las etapas, recursos necesarios, responsables y métricas de éxito del piloto, con el fin de evaluar la eficacia y viabilidad del modelo en la práctica.	Propuesta de plan piloto documentada, incluyendo la definición de etapas, recursos, responsables y métricas de éxito, lista para su ejecución en un entorno real de fábrica de software.

Tabla 1. Metas por objetivos

CAPÍTULO II. Marco teórico o conceptual

A) Gestión de proyectos de software

Un proyecto se define clásicamente como “un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único” (Institute, 2021). Esta definición enfatiza que todo proyecto tiene un inicio y fin determinados, un alcance específico y resultados únicos. En el ámbito de desarrollo de software, un proyecto implica coordinar recursos humanos y técnicos durante un tiempo limitado para construir un sistema o aplicación que satisfaga requisitos definidos. La naturaleza temporal y única de los proyectos de software los distingue de las operaciones continuas de una organización.

El éxito de un proyecto tradicionalmente se ha evaluado mediante la “triple restricción” de tiempo, costo y alcance (Pérez, 2024). Es decir, si el software se entrega dentro del plazo previsto, respetando el presupuesto asignado y cumpliendo con todos los requisitos funcionales acordados, el proyecto se considera exitoso bajo esta visión clásica (Pérez, 2024). Sin embargo, la literatura reciente destaca que esta visión es limitada. Organismos como el PMI han ampliado la noción de éxito para incluir la calidad del producto y la satisfacción de los interesados con los beneficios obtenidos (Institute, 2021). En proyectos de software modernos, no basta con terminar “a tiempo y en presupuesto”; es fundamental que el proyecto genere valor para el cliente o la organización, ya sea en forma de mejoras en procesos, ventajas competitivas o alta satisfacción del usuario final (Pérez, 2024). Por lo tanto, la gestión exitosa de proyectos de software equilibra las restricciones de plazo y costo con la entrega de valor y calidad.

Para lograr estos objetivos, la adecuada gestión de los recursos del proyecto es crucial. En gestión de proyectos, recurso se entiende ampliamente como “aquello que es necesario para cumplir una tarea”, abarcando desde

recursos financieros, materiales y tecnológicos hasta el tiempo disponible y el equipo humano (Slack, 2023). En un proyecto de fábrica de software, los recursos incluyen desarrolladores, herramientas de desarrollo, infraestructuras de hardware y software, presupuesto y tiempo de trabajo. Una planificación eficiente asegura que haya recursos suficientes y bien asignados para cada actividad (Cristian Rincón-Guio Ing. Esp. Mg. PhD(c), 2017), evitando cuellos de botella y retrasos. La gestión de recursos efectiva implica administrar estos insumos de manera óptima, asegurando que el personal tenga las habilidades adecuadas, que el equipamiento soporte las necesidades técnicas y que el tiempo se distribuya conforme a las prioridades del proyecto. Esto reduce riesgos de sobrecarga del equipo, controla los costos y aumenta la probabilidad de éxito del proyecto.

Otro concepto fundamental es el del marco metodológico bajo el cual se conduce el proyecto. (Arias, 2006) define el marco metodológico como el “conjunto de pasos, técnicas y procedimientos que se emplean para formular y resolver problemas”. En otras palabras, es la estructura metodológica que guía cómo se planifica, ejecuta y controla un proyecto. En proyectos de software, seleccionar un marco metodológico adecuado significa decidir si se seguirá un enfoque tradicional, ágil u otro, con sus respectivas técnicas. Este marco proporciona disciplina y coherencia al trabajo del equipo, asegurando que todos utilicen prácticas consistentes y probadas. Un buen marco metodológico se adapta al contexto del proyecto: por ejemplo, proyectos con requisitos muy claros y poco cambiantes pueden beneficiarse de metodologías predictivas, mientras que proyectos con alta incertidumbre requieren marcos más flexibles. En síntesis, el marco metodológico escogido determina la forma en que el proyecto avanza desde la concepción hasta la entrega, influyendo en la organización de tareas, la comunicación del equipo y la capacidad de responder a imprevistos.

B) Enfoques de gestión de proyectos: tradicional, ágil e híbrido

En la ingeniería de software han coexistido distintos enfoques para gestionar proyectos, principalmente los enfoques tradicional, ágil e híbrido, cada uno con principios y prácticas propias. Un proyecto tradicional sigue un modelo de gestión predictivo o en cascada, con etapas secuenciales bien definidas (por ejemplo: análisis, diseño, implementación, pruebas, implantación). En la gestión tradicional de proyectos se pone especial énfasis en la planificación exhaustiva al inicio, la documentación detallada y el seguimiento estricto de un plan establecido (Institute, 2021). Los requisitos del software se procuran fijos desde el comienzo y cualquier cambio suele ser rígidamente controlado. Bajo este enfoque, el alcance del proyecto se mantiene estable, mientras que el tiempo y el costo pueden variar para asegurar que se cumplan todos los requisitos (Institute, 2021). Esto proporciona orden y predictibilidad, pero puede llevar a problemas cuando surgen cambios no previstos, ya que la capacidad de adaptación es limitada. En entornos de alta certidumbre (por ejemplo, proyectos gubernamentales con especificaciones reguladas), la metodología tradicional ofrece ventajas como claridad en las etapas y documentación completa, pero en entornos cambiantes puede resultar poco flexible.

En contraste, un proyecto ágil adopta un enfoque adaptativo enfocado en la flexibilidad y la colaboración. El término “Ágil” se popularizó a partir del Manifiesto Ágil de 2001, que estableció valores y principios para desarrollar software de manera más receptiva al cambio. Este manifiesto propone cuatro valores fundamentales: (1) individuos e interacciones sobre procesos y herramientas, (2) software funcionando sobre documentación extensiva, (3) colaboración con el cliente sobre negociación contractual, y (4) respuesta ante el cambio sobre seguir un plan preestablecido (Kent Beck, 2001). En esencia,

la gestión ágil de proyectos privilegia las personas y la comunicación, la entrega temprana y continua de software útil, la involucración activa del cliente durante el desarrollo y la habilidad de adaptar el rumbo del proyecto cuando cambian las circunstancias. En un proyecto ágil de software, el trabajo se organiza en iteraciones cortas con entregas funcionales al final de cada ciclo, lo que permite obtener retroalimentación frecuente. La planificación es incremental: en lugar de un plan cerrado al inicio, el equipo reevalúa prioridades en cada iteración (sprint). Esto conlleva menos énfasis en documentación voluminosa y más en software tangible y en comunicación efectiva. Un ejemplo típico es el uso de Scrum o Kanban para gestionar el desarrollo: se trabaja en equipo de forma autoorganizada, adaptándose rápidamente a cambios en requerimientos o entornos. La agilidad aporta flexibilidad y velocidad de respuesta, reduciendo el riesgo de desarrollar características no deseadas y aumentando la probabilidad de satisfacción del cliente, aunque implica abrazar la incertidumbre y requiere alta disciplina en el equipo para mantener la cadencia de entregas.

Entre ambos extremos surge el enfoque híbrido, cada vez más común en la práctica. Un proyecto híbrido combina técnicas predictivas tradicionales con prácticas ágiles dentro del mismo proyecto, buscando aprovechar lo mejor de ambos (Abad, 2025). La premisa es aplicar enfoques diferentes según las características de cada fase o componente del proyecto. Por ejemplo, en un proyecto híbrido, aquellas partes con requisitos bien conocidos y estables se gestionan con un enfoque tradicional (priorizando planificación detallada y control), mientras que las áreas con alta incertidumbre o propensas al cambio constante se gestionan de forma ágil, permitiendo iteraciones rápidas y retroalimentación continua. Esta dualidad permite mantener control y planificación rigurosa donde es necesaria, a la vez que se introduce flexibilidad y adaptación (Abad, 2025). El resultado es un ciclo de vida mixto: puede iniciarse con una planificación macro al estilo tradicional,

pero la construcción y entrega se realizan en iteraciones ágiles. El enfoque híbrido es útil, por ejemplo, en grandes proyectos corporativos donde ciertos entregables requieren cumplimiento estricto de normativas (ideal para método tradicional) pero al mismo tiempo se desea innovación rápida en otros módulos (aplicando ágil). Al adoptar un modelo híbrido, se maximizan los beneficios de cada enfoque minimizando sus desventajas: se logra mayor rapidez de entrega y capacidad de cambio sin renunciar a cierta predictibilidad y documentación donde es crítica. La gestión de proyectos híbrida brinda una adaptabilidad estratégica, ajustándose a las necesidades específicas de cada proyecto y permitiendo una transición más suave para organizaciones acostumbradas al estilo tradicional que incursionan en la agilidad.

C) Metodología ágil: Scrum y Sprint

Dentro del universo ágil, Scrum destaca como uno de los marcos de trabajo más utilizados para la gestión de proyectos de desarrollo de software. Scrum es un marco de trabajo ágil ligero que ayuda a equipos a generar valor de forma iterativa y adaptativa en contextos complejos (Sutherland, 2020). A diferencia de una metodología prescriptiva con pasos detallados, Scrum provee una estructura básica de roles, eventos y artefactos que los equipos deben complementar con técnicas específicas según su contexto. En Scrum se define un pequeño conjunto de roles clave: Product Owner (responsable de maximizar el valor del producto y gestionar la lista de requisitos o Product Backlog), el Scrum Master (facilitador que asegura que el equipo siga las prácticas Scrum y mejora el proceso) y el Equipo de Desarrollo (multidisciplinario y autoorganizado, encargado de construir el producto). El trabajo se organiza en ciclos cortos llamados Sprints, y en cada Sprint el equipo produce un incremento de producto potencialmente entregable. Scrum

enfatisa principios ágiles fundamentales: transparencia (todos los miembros y stakeholders tienen visibilidad del progreso y los obstáculos), inspección frecuente del producto y del proceso (por medio de reuniones como la Daily Scrum, Sprint Review y Sprint Retrospective), y adaptación continua (ajustar el plan de trabajo en función de la retroalimentación y lo aprendido en cada iteración). Según la Guía Scrum oficial, "Scrum es un marco de trabajo ligero que ayuda a las personas, equipos y organizaciones a generar valor mediante soluciones adaptativas a problemas complejos" (Sutherland, 2020). Esto refleja cómo Scrum se enfoca en dividir problemas grandes en tareas manejables dentro de cada Sprint, permitiendo responder rápidamente a la complejidad y al cambio. Gracias a su sencillez y enfoque empírico, Scrum se ha difundido más allá del software hacia proyectos de diversa índole que requieren equipos altamente colaborativos y flexibles.

El concepto de Sprint es central en Scrum y en muchas metodologías ágiles. Un Sprint es una iteración con duración fija, típicamente de entre una y cuatro semanas, durante la cual el equipo Scrum desarrolla, integra y prueba un subconjunto de funcionalidades del producto. Los Sprints marcan el "latido del corazón" de Scrum, transformando ideas en valor en períodos cortos y regulares (Sutherland, 2020). Cada Sprint comienza con una planificación (Sprint Planning) en la que el equipo selecciona, desde la lista priorizada de requisitos (Product Backlog), las tareas que se compromete a completar en ese Sprint, definiendo un objetivo claro (Sprint Goal). Durante el Sprint, no se permiten cambios que comprometan dicho objetivo, lo que proporciona estabilidad a corto plazo (Sutherland, 2020). El equipo realiza reuniones diarias de sincronización (Daily Scrum) para inspeccionar el progreso y adaptar el plan del Sprint si es necesario. Al finalizar el Sprint, se lleva a cabo una Revisión del Sprint donde el equipo muestra el incremento de software terminado a los interesados para obtener retroalimentación, y una Retrospectiva donde el equipo reflexiona sobre su proceso de trabajo e

implementa mejoras. Según la guía formal, “los Sprints son los latidos de corazón de Scrum, en los cuales las ideas se transforman en valor. Son eventos con duración fija de un mes o menos. Un nuevo Sprint comienza inmediatamente tras la conclusión del Sprint anterior” (Sutherland, 2020). Esto enfatiza la naturaleza cíclica e ininterrumpida del desarrollo ágil: al terminar un Sprint, se planifica el siguiente, asegurando una entrega continua. Trabajar en Sprints cortos mejora la capacidad de predicción (se inspecciona el progreso al menos mensualmente) y controla el riesgo, ya que cualquier desviación o problema puede detectarse pronto y abordarse en el siguiente ciclo (Sutherland, 2020). En resumen, el Sprint proporciona al proyecto un ritmo sostenido de producción y aprendizaje: cada iteración resulta en software funcional y lecciones aprendidas, habilitando la adaptación constante del proyecto hasta alcanzar el resultado final deseado.

D) Fábricas de software y desarrollo de software

El término fábrica de software alude a una modalidad organizativa en la industria del software orientada a la producción sistematizada de aplicaciones para clientes, de manera análoga a cómo una fábrica manufactura bienes. Una fábrica de software es típicamente una empresa (o unidad dentro de una empresa) cuya misión es el desarrollo de software conforme a los requisitos específicos que solicitan sus clientes (Escobar, 2024). Es decir, se dedica a ejecutar proyectos de desarrollo de software de forma repetitiva y estandarizada, convirtiendo las necesidades de distintos clientes en soluciones software a medida. En este modelo, la organización funciona con procesos bien definidos y roles especializados, buscando maximizar eficiencia, calidad y predictibilidad en las entregas. Frecuentemente, las fábricas de software venden proyectos completos u horas de desarrollo como su principal fuente

de ingresos (Escobar, 2024). A diferencia de una consultora que podría abarcar todo el ciclo de vida (desde la consultoría inicial y el diseño funcional hasta la implementación), una fábrica de software suele focalizarse en la construcción y pruebas a partir de especificaciones dadas (Escobar, 2024). Esto implica que una factoría de software opera con entradas bien definidas (requisitos o diseños proporcionados) y produce salidas concretas (sistemas desarrollados), con procedimientos estandarizados en el medio. La estandarización es un rasgo clave: las fábricas de software establecen procesos y metodologías consistentes que se aplican de forma uniforme en todos los proyectos, con el objetivo de lograr reutilización, controlar la calidad y medir la productividad. El concepto de "fábrica" sugiere también un énfasis en la especialización del trabajo (por ejemplo, equipos separados para análisis, programación, pruebas) y en la incorporación de herramientas que automaticen tareas repetitivas del desarrollo. Surgido desde finales de la década de 1960 (Escobar, 2024), el enfoque de fábricas de software ha ido evolucionando para responder a desafíos como asegurar la fiabilidad de los productos, cumplir plazos y presupuestos, y gestionar de forma consistente múltiples proyectos simultáneos (Escobar, 2024). Una fábrica de software representa un entorno de desarrollo altamente organizado, orientado a la producción eficiente de soluciones, lo cual la convierte en terreno propicio para implementar mejoras de procesos y nuevas tecnologías (como Inteligencia Artificial) de manera estratégica.

El desarrollo de software en sí es el proceso ingenieril mediante el cual se concibe, diseña, construye, prueba y mantiene un sistema de software. Implica un conjunto de actividades técnicas y de gestión que transforman una idea o necesidad en un producto software funcional. Estas actividades abarcan típicamente la planificación del proyecto, el análisis de requisitos, el diseño de la arquitectura y componentes, la programación o codificación, las pruebas (unitarias, integrales, de sistema, etc.), la implantación en el ambiente de

producción y las operaciones de mantenimiento posteriores. La literatura describe este proceso a través de modelos de ciclo de vida, como el modelo en cascada, el iterativo incremental, el espiral o el modelo ágil, entre otros (Microsoft, 2025). Independientemente del modelo, el objetivo último del desarrollo de software es producir un producto que satisfaga las necesidades del usuario y cumpla con criterios de calidad (funcionalidad correcta, rendimiento, usabilidad, seguridad, etc.) de forma eficiente y repetible (IBM, 2025). Es importante distinguir desarrollo de software de simples tareas de programación: el desarrollo abarca no solo escribir código, sino también actividades de diseño y gestión necesarias para asegurar que el software construido realmente resuelva el problema correcto. En la práctica de una fábrica de software, el desarrollo se realiza siguiendo procesos definidos (por ejemplo, una metodología de desarrollo seleccionada en el marco metodológico mencionado) para lograr consistencia entre proyectos. Modernamente, muchas organizaciones de desarrollo adoptan enfoques DevOps (integrando desarrollo y operaciones) para acelerar la entrega continua, como se verá más Adelante (IBM, 2025). Así, el desarrollo de software es tanto una disciplina técnica como un proceso de negocio, y su gestión eficaz es esencial para que las fábricas de software puedan escalar su producción sin comprometer la calidad.

E) Calidad y mejora de procesos: CMMI, DevOps y métricas DORA

Para que una fábrica de software adopte exitosamente nuevas tecnologías como la Inteligencia Artificial, es imprescindible contar con marcos de calidad y mejora continua que guíen sus procesos de desarrollo. En este contexto, el modelo CMMI (Capability Maturity Model Integration) ha sido

ampliamente utilizado como referencia de buenas prácticas. CMMI es un modelo de procesos y comportamiento organizacional orientado a la mejora continua, que ayuda a las organizaciones a evaluar la madurez de sus procesos de desarrollo y a implementar prácticas más eficientes y productivas (White, 2021). Desarrollado originalmente por el Software Engineering Institute de la Universidad Carnegie Mellon, CMMI define una serie de niveles de madurez (habitualmente del nivel 1 al 5) que describen la capacidad de una organización para seguir procesos bien controlados y optimizados. Cada nivel de madurez conlleva un conjunto de áreas de proceso y objetivos específicos: desde procesos iniciales e impredecibles (Nivel 1), pasando por procesos gestionados y estandarizados (Niveles 2 y 3), hasta procesos cuantitativamente gestionados y en mejora continua (Niveles 4 y 5). La adopción de CMMI en fábricas de software busca institucionalizar prácticas consistentes como la gestión de requerimientos, la planificación de proyectos, la garantía de calidad y la gestión de riesgos, entre otras. Al seguir las mejores prácticas probadas que propone CMMI, las empresas de desarrollo pueden reducir la variabilidad en sus proyectos, predecir mejor sus resultados y detectar problemas de proceso para corregirlos proactivamente (White, 2021). CMMI funciona como un marco de optimización de procesos que conduce a productos de software de mayor calidad entregados de forma más eficiente. En el contexto de este trabajo, CMMI provee un sustento teórico importante: una fábrica de software con altos niveles de madurez de proceso estará mejor preparada para integrar tecnologías avanzadas (como IA) de manera estructurada y controlada.

Otro concepto clave en las organizaciones de desarrollo actuales es DevOps, que representa una filosofía y conjunto de prácticas destinadas a integrar de manera fluida el desarrollo de software (Dev) con las operaciones de TI (Ops). DevOps nació para eliminar las tradicionales barreras o silos entre los equipos de desarrollo, que construyen el software, y los equipos de

operaciones o infraestructura, que despliegan y administran ese software en producción. En términos formales, “DevOps es un conjunto de prácticas que fusiona el desarrollo de software y las operaciones de TI, con el objetivo de acelerar el ciclo de vida del desarrollo de software y proporcionar una entrega continua de alta calidad” (Microsoft, 2025). Para lograrlo, DevOps promueve una serie de principios: la automatización extensiva de las tareas de construcción, prueba, integración y despliegue; el monitoreo continuo de las aplicaciones y la infraestructura; la colaboración constante entre desarrolladores, testers y personal de operaciones desde las etapas iniciales del proyecto; y una cultura de responsabilidad compartida por el éxito del producto en producción. En la práctica, la implementación de DevOps suele implicar la adopción de herramientas de Integración Continua/Despliegue Continuo (CI/CD), que permiten compilar y probar el código de manera automática con cada cambio y desplegar nuevas versiones de forma frecuente y confiable (IBM, 2025). DevOps complementa las metodologías ágiles: mientras Agile organiza el trabajo de desarrollo de forma iterativa con retroalimentación rápida, DevOps extiende esa agilidad hasta la puesta en producción, asegurando que la entrega de valor al usuario final sea continua. Las fábricas de software que abrazan DevOps tienden a lograr ciclos de desarrollo más cortos y despliegues más frecuentes sin sacrificar la estabilidad, gracias a prácticas como la gestión de la configuración automatizada, la infraestructura como código, pruebas automáticas robustas y una supervisión activa de aplicaciones en tiempo real (Microsoft, 2025). Este enfoque ha demostrado mejorar significativamente los tiempos de entrega y la calidad de los sistemas, alineándose con la necesidad de las empresas de responder rápidamente a los cambios del mercado. Por lo tanto, en una estrategia de adopción de IA, la cultura DevOps serviría para integrar las nuevas soluciones de IA en el flujo de desarrollo y operaciones de manera

eficiente, permitiendo iterar y desplegar modelos de IA o funcionalidades inteligentes con rapidez y control.

Para medir y gestionar el desempeño de los equipos de desarrollo (especialmente en entornos DevOps), se han identificado métricas clave conocidas como métricas DORA. Las métricas DORA constituyen un conjunto de cuatro indicadores clave de rendimiento que evalúan la eficacia del proceso de entrega de software (Blasco, 2024). Estas métricas fueron derivadas de investigaciones científicas sobre DevOps lideradas por (Dra. Nicole Forsgren, 2008). Las cuatro métricas DORA son: (1) Frecuencia de despliegue, que mide con qué frecuencia el equipo pone cambios de software en producción (indicador de velocidad de entregas) (Schuster, 2023); (2) Tiempo de entrega de cambios que cuantifica cuánto tiempo transcurre desde que se comete un cambio en el código hasta que ese cambio está desplegado y disponible para el usuario, reflejando la agilidad del proceso de desarrollo (Schuster, 2023); (3) Tiempo medio de recuperación ante incidentes (MTTR, Mean Time to Recover), que indica la rapidez con la que el equipo puede restaurar un servicio tras una falla o caída en producción (Schuster, 2023); y (4) Tasa de fallos en cambios, que es el porcentaje de despliegues o cambios que provocan incidentes o degradaciones en el servicio (por ejemplo, introducen bugs serios que requieren ajuste) (Schuster, 2023). Estas métricas distinguen a los equipos “elite” de alto rendimiento (capaces de desplegar muy frecuentemente, con tiempos cortos, recuperaciones rápidas y pocas fallas) de aquellos con procesos menos eficientes. Su relevancia teórica radica en que proveen medidas objetivas de la capacidad de entrega continua y confiable de un equipo de desarrollo. En una fábrica de software, monitorizar las métricas DORA permite identificar cuellos de botella (por ejemplo, si el tiempo de entrega es muy largo, podría indicar problemas en el flujo de trabajo o en la fase de pruebas) y oportunidades de mejora (por ejemplo, automatizar despliegues para aumentar la frecuencia). A efectos del presente

estudio, las métricas DORA ofrecen una forma de evaluar el impacto de introducir IA en los procesos: idealmente, la adopción estratégica de IA (por ejemplo, herramientas de IA para asistir en pruebas o en monitoreo) debería reflejarse en mejoras en estas métricas, como despliegues más ágiles, detección y resolución más rápidas de incidencias, etc. En resumen, tanto CMMI, como DevOps y las métricas DORA conforman un andamiaje teórico-práctico para gestionar y mejorar los procesos de desarrollo en fábricas de software, proporcionando la base sobre la cual implementar innovaciones como la Inteligencia Artificial de manera exitosa y sostenible.

F) Inteligencia Artificial y aprendizaje automático en el desarrollo de software

La Inteligencia Artificial (IA) es un campo de la informática que busca crear sistemas o programas capaces de realizar tareas que típicamente requerirían inteligencia humana, tales como aprender, razonar, percibir el entorno o tomar decisiones. La Real Academia Española define la inteligencia artificial como la “disciplina científica que se ocupa de crear programas informáticos que ejecuten operaciones comparables a las que realiza la mente humana” (Azkune, 2024). Esto implica dotar a las máquinas de ciertas capacidades cognitivas, por ejemplo, reconocer patrones, entender lenguaje natural, resolver problemas o adaptarse mediante experiencia, emulando en alguna medida la forma en que lo haría una persona. En la práctica, la IA abarca un amplio espectro de técnicas y enfoques, desde algoritmos simbólicos basados en lógica hasta métodos estadísticos de aprendizaje a partir de datos. Durante décadas, la IA ha evolucionado pasando por diferentes paradigmas, en la actualidad se ha concentrado especialmente en métodos de aprendizaje automático y aprendizaje profundo, que han

demostrado gran eficacia en tareas antes consideradas difíciles para las máquinas (como reconocer imágenes o participar en juegos complejos). Para las fábricas de software, la IA ofrece oportunidades significativas: por un lado, integrar componentes de IA en los productos de software (por ejemplo, agregar funcionalidades “inteligentes” como recomendaciones personalizadas, chatbots de atención, visión artificial, etc.), y, por otro lado, aplicar la IA a sus propios procesos de desarrollo (por ejemplo, estimar esfuerzos de proyecto, detección automática de defectos en código, asistencia en pruebas o despliegues). Ambas facetas requieren una comprensión sólida de qué es y cómo funciona la IA, para adoptarla estratégicamente.

Un subcampo fundamental de la IA es el machine learning o aprendizaje automático, el cual se centra en que las computadoras aprendan por sí solas a mejorar en la realización de una tarea, a partir de datos o de la experiencia previa, en lugar de seguir exclusivamente instrucciones programadas de forma estática. Formalmente, el aprendizaje automático se define como la disciplina que “consiste en la investigación de algoritmos informáticos que mejoran automáticamente a través de la experiencia” (UNESCO, 2024). En lugar de programar explícitamente cada regla de comportamiento, en machine learning se entrena un modelo matemático con conjuntos de datos para que generalice patrones y sea capaz de hacer predicciones o tomar decisiones con datos nuevos. Hay varias formas de aprendizaje automático: el aprendizaje supervisado, donde el modelo aprende a partir de ejemplos etiquetados (por ejemplo, conjuntos de datos de entrada con sus respuestas correctas) y su objetivo es predecir las etiquetas de nuevas entradas; el aprendizaje no supervisado, donde descubre por sí mismo estructuras ocultas en datos sin etiquetar (por ejemplo, agrupando datos similares); y el aprendizaje por refuerzo, donde un agente aprende a través de prueba y error recibiendo recompensas o castigos en un entorno. En el contexto de desarrollo de software, el aprendizaje automático se emplea cada vez más para mejorar

productos y procesos: por ejemplo, se pueden entrenar modelos para predecir cuántos defectos tendrá un módulo de software según métricas de código, o para optimizar la asignación de tareas en un equipo en base a rendimiento histórico. Una aplicación relevante es el análisis predictivo, que usa técnicas de estadística y machine learning para analizar datos históricos y actuales con el fin de predecir eventos futuros (Mucci, 2025). El análisis predictivo en proyectos de software podría, por ejemplo, anticipar riesgos (¿qué probabilidad hay de retraso en determinada funcionalidad?), estimar la demanda de servidores después del lanzamiento de una nueva versión, o pronosticar la evolución de costos. IBM define la IA predictiva precisamente como el uso de análisis estadístico y aprendizaje automático para identificar patrones, anticipar comportamientos y prever próximos eventos (Mucci, 2025). Esta capacidad de predicción aporta un gran valor en la planificación y gestión de proyectos, permitiendo decisiones proactivas basadas en datos (por ejemplo, redistribuir recursos si se predice que una tarea crítica se retrasará).

Otro concepto medular de la IA son las redes neuronales artificiales. Inspiradas en la estructura del cerebro humano, las redes neuronales son modelos computacionales compuestos por múltiples unidades simples interconectadas (neuronas artificiales), organizadas en capas, que cooperan para procesar información y resolver problemas complejos. Una red neuronal artificial es esencialmente un algoritmo de aprendizaje automático que toma decisiones o predicciones imitando de forma simplificada el comportamiento de las neuronas biológicas al activarse en respuesta a estímulos (IBM, 2025). En una red neuronal típica, la capa de entrada recibe datos brutos (por ejemplo, píxeles de una imagen, atributos de un caso), luego hay una o varias capas ocultas donde cada neurona calcula una combinación ponderada de las salidas de la capa anterior y aplica una función de activación no lineal, y finalmente una capa de salida entrega el resultado (por ejemplo, la

clasificación predicha). Las redes neuronales tienen la capacidad de aprender ajustando los pesos de las conexiones entre neuronas durante una fase de entrenamiento, generalmente usando algoritmos como backpropagation o propagación hacia atrás, que minimizan el error en las predicciones del modelo con respecto a ejemplos conocidos. Con suficientes neuronas y capas, las redes neuronales pueden aproximar funciones muy complejas; esto ha dado lugar al campo de deep learning o aprendizaje profundo, que emplea redes neuronales de muchas capas para abordar tareas como reconocimiento de voz, visión por computador, traducción automática, entre otras. Según IBM, “una red neuronal es un programa o modelo de machine learning que toma decisiones de forma similar al cerebro humano, al emplear procesos que imitan la forma en que las neuronas biológicas trabajan juntas para identificar fenómenos, sopesar opciones y llegar a conclusiones” (IBM, 2025). Este paralelismo con el cerebro permite a las redes neuronalizar patrones sutiles en datos masivos. En las fábricas de software, las redes neuronales (y en general las técnicas de IA) pueden integrarse tanto en los productos desarrollados (por ejemplo, un módulo de visión artificial dentro de un sistema mayor) como en herramientas internas (por ejemplo, una red neuronal que analice el historial de commits de código para identificar componentes propensos a defectos). Comprender cómo funcionan las redes neuronales artificiales es parte del sustento teórico para diseñar o adoptar soluciones basadas en IA de manera informada y ética.

G) Ética de la Inteligencia Artificial

La incorporación de la Inteligencia Artificial en cualquier ámbito tecnológico trae aparejado un conjunto de consideraciones éticas que deben ser abordadas rigurosamente. La ética de la IA se refiere a los principios y

valores que orientan el desarrollo y el uso responsable de sistemas de inteligencia artificial, buscando asegurar que estas tecnologías actúen en beneficio de la sociedad y no causen daño. Dado el creciente papel de la IA en nuestra vida cotidiana y en procesos de negocio, “garantizar que la IA se desarrolle, diseñe e implemente de manera ética es fundamental” (The University of Texas at Austin, 2025). Esto implica, por ejemplo, diseñar algoritmos de aprendizaje automático que sean justos (evitando sesgos discriminatorios en sus decisiones), transparentes (que sus criterios de decisión puedan explicarse o auditarse en la medida de lo posible), seguros (resistentes a fallos catastróficos o a usos malintencionados) y responsables (sujetos a rendición de cuentas por parte de sus creadores u operadores). Diversos marcos han propuesto principios éticos para la IA; entre los más citados se encuentran la no maleficencia (que los sistemas de IA no perjudiquen a las personas), la justicia (que sus beneficios y riesgos se distribuyan equitativamente sin sesgos indebidos), la autonomía y consentimiento (especialmente relevante cuando la IA toma decisiones que afectan a individuos, quienes deberían mantener cierto grado de control o conocimiento), la explicabilidad (poder comprender o rastrear las decisiones de la IA) y la privacidad (proteger los datos personales usados por algoritmos) (Iberia, 2024) (The University of Texas at Austin, 2025). En el contexto de las fábricas de software, la ética de la IA cobra doble importancia. Por un lado, si la fábrica desarrolla productos de software con componentes de IA incorporados, debe asegurar que dichos productos cumplan con estándares éticos y normativos (por ejemplo, un sistema de IA integrado en una aplicación financiera no debe perpetuar sesgos contra cierto grupo demográfico). Por otro lado, si la fábrica emplea herramientas de IA para optimizar su proceso interno (por ejemplo, una IA que priorice automáticamente los tickets de soporte o que asigne tareas a desarrolladores), también debe vigilar que esas aplicaciones no introduzcan tratamientos

injustos hacia los miembros del equipo o decisiones opacas sin supervisión humana. La ética de la IA, también llamada a veces “IA responsable”, abarca tanto el proceso de desarrollo (cómo construimos la IA) como el producto en sí (cómo se comporta la IA) (The University of Texas at Austin, 2025). Incorporar consideraciones éticas desde la fase de diseño lo que se conoce como *ethics by design*, se está volviendo una práctica recomendada. Finalmente, aspectos éticos incluyen el cumplimiento de leyes y regulaciones emergentes en este campo (como las directrices de la Unión Europea para una IA confiable), y la construcción de comités o instancias de revisión ética dentro de las organizaciones para evaluar proyectos de IA. Ignorar la ética en la IA no solo puede conllevar riesgos reputacionales y legales, sino que puede contradecir el fin último de cualquier tecnología: mejorar la condición humana. Por ello, una adopción estratégica de IA en fábricas de software debe ir acompañada de un sólido marco ético que guíe qué tipos de aplicaciones de IA desarrollar, con qué datos entrenarlas, cómo evaluar sus impactos y cómo desplegarlas de forma segura y justa.

H) Necesidad de un modelo de gestión para la adopción estratégica de IA

La revisión de los conceptos anteriores pone de manifiesto que la adopción estratégica de Inteligencia Artificial en fábricas de software requiere un enfoque integral de gestión que articule elementos de gestión de proyectos, metodologías de desarrollo, mejora de procesos y consideraciones éticas. En primer lugar, integrar IA en una fábrica de software típicamente se materializa a través de proyectos específicos (por ejemplo, un proyecto para incorporar un motor de recomendación inteligente en un producto existente, u otro para implementar una herramienta interna de estimación automatizada mediante

IA). Gestionar exitosamente estos proyectos demanda entender su naturaleza: pueden presentar alta incertidumbre técnica (propia de la exploración de IA), lo que sugiere la conveniencia de enfoques ágiles o híbridos que toleren cambios y descubrimientos a lo largo del camino. Al mismo tiempo, deben alinearse con objetivos claros de negocio para generar valor, tal como enfatiza la gestión tradicional orientada a éxito en términos de alcance, tiempo, costo y beneficios. Esto implica que un modelo de gestión adecuado deberá combinar disciplina y flexibilidad, planificando hitos y recursos con rigor, pero permitiendo iteraciones y aprendizaje continuo, como un híbrido entre lo predictivo y lo adaptativo.

En segundo lugar, adoptar IA no es solo una cuestión técnica sino también organizacional. Las fábricas de software más maduras en sus procesos (por ejemplo, con altos niveles en CMMI o con prácticas DevOps avanzadas) estarán en mejor posición para integrar componentes de IA, ya que cuentan con procesos estandarizados, automatización y cultura de mejora continua. Sin embargo, la incorporación de IA podría requerir nuevas competencias en el equipo (científicos de datos, especialistas en aprendizaje automático) y la adaptación de procesos existentes (por ejemplo, integrando el entrenamiento de modelos en el ciclo de desarrollo o ajustando las pruebas para sistemas no determinísticos). Un modelo de gestión para la adopción de IA debería proporcionar lineamientos para gestionar estos recursos especializados y posiblemente escasos, facilitando su colaboración con los equipos de desarrollo tradicionales. Asimismo, debería incluir métricas adecuadas (inspiradas en las métricas DORA u otras) para evaluar el impacto de la IA en el desempeño: por ejemplo, ¿los despliegues son más frecuentes y confiables gracias a la automatización inteligente?, ¿se reduce el tiempo de resolución de incidencias usando algoritmos predictivos de detección de fallos? Cuantificar estos efectos ayudará a refinar la estrategia de adopción.

Por último, el modelo de gestión debe incorporar directrices éticas claras. La ética de la IA no puede ser un apéndice tratado al final, sino un componente intrínseco desde la planificación del proyecto hasta la entrega y operación del producto. Esto implica que en la evaluación inicial de viabilidad de una implementación de IA se consideren riesgos éticos (¿hay sesgos en los datos de entrenamiento?) que durante el desarrollo se incluyan actividades para mitigar esos riesgos (por ejemplo, auditorías de equidad del modelo, consultas con el equipo sobre aceptabilidad), y que en la fase de despliegue y mantenimiento se monitoricen los comportamientos del sistema para detectar efectos no deseados. Un enfoque de gestión integral deberá establecer roles o comités encargados de la gobernanza de IA, asegurar el cumplimiento de regulaciones vigentes y fomentar una cultura organizacional donde los desarrolladores y responsables de proyecto estén sensibilizados y capacitados en ética de la tecnología.

Para concluir, la necesidad de un modelo de gestión específico para la adopción estratégica de IA en fábricas de software surge de la complejidad multidimensional de este desafío: se trata de proyectos técnicamente complejos y de alta incertidumbre, que requieren métodos ágiles de gestión; se desarrollan en entornos organizados que deben mantener o mejorar su nivel de calidad de proceso al introducir nuevas prácticas; y abarcan implicaciones éticas y humanas de amplio alcance. Un modelo de gestión sólido actuará como un puente entre el mundo de la ingeniería de software tradicional (con sus proyectos, metodologías y métricas consolidadas) y el mundo de la IA (con su innovación, su necesidad de experimentación y sus consideraciones éticas), garantizando que la incorporación de la IA sea estratégica, es decir, alineada con los objetivos de la organización, sostenida por prácticas de proyecto efectivas, y conducente a beneficios tangibles con riesgos controlados. La elaboración de tal modelo se sustentará en los conceptos teóricos desarrollados en este capítulo, sirviendo como base para

proponer en capítulos posteriores una guía metodológica que oriente a las fábricas de software en su camino hacia la implementación de la inteligencia artificial en sus procesos.

I) Definición académica de prompt

En la literatura académica sobre IA generativa, el término prompt suele definirse de forma concisa. Por ejemplo, (Jenny T Liang, 2024) definen un prompt simplemente como “una consulta en lenguaje natural dirigida a un modelo fundacional” (Jenny T Liang, 2024) (foundation model, es decir, un modelo de IA pre-entrenado de gran escala). Esta definición destaca que el prompt típicamente se formula en lenguaje natural y es lo que dispara la generación de respuesta por parte del modelo. De manera similar, otros trabajos de investigación describen los prompts como instrucciones explícitas que le piden al modelo realizar una tarea y producir una salida (Yasar, 2023). Desde la perspectiva académica un prompt equivale a la pregunta o instrucción inicial con la que se “programa” o condiciona al modelo de lenguaje para que realice una cierta acción.

J) Definición técnica de *prompt*

La documentación técnica de organizaciones líderes en IA también ofrece definiciones claras de prompt. OpenAI, la empresa creadora de GPT-3/4 y ChatGPT, define un prompt como un texto de entrada que inicia una conversación o desencadena una respuesta del modelo (OpenAI, 2024). En otras palabras, es la entrada proporcionada al modelo para indicarle qué respuesta esperamos. Microsoft describe de forma similar que en los modelos

basados en prompts, “el usuario interactúa ingresando un prompt de texto, al cual el modelo responde con una completación de texto” (Microsoft, 2025). Importante hay que señalar que, aunque lo más común es usar texto, un prompt no siempre tiene que ser texto plano: en ciertos sistemas generativos multimodales, una imagen o un audio también puede actuar como prompt para iniciar la generación de contenido (por ejemplo, describir una imagen para que el modelo la continúe o genere variaciones) (OpenAI, 2024). Técnicamente un prompt es la entrada proporcionada al modelo de IA (ya sea texto u otro medio) que configura su comportamiento y da lugar a una salida generada.

CAPÍTULO III. Marco Metodológico

En este capítulo se detalla el enfoque metodológico adoptado para el estudio "Adopción estratégica de la Inteligencia Artificial en fábricas de software". Al respecto, (Balestrini, 2006) define "el marco metodológico como la instancia referida a los métodos, las diversas reglas, registros, técnicas y protocolos con los cuales una teoría y su método calculan las magnitudes de lo real". Según (Camacho, 2008), el marco metodológico está referida al "cómo se realizará la investigación, muestra el tipo y diseño de la investigación, población, muestra, técnicas e instrumentos para la recolección de datos, validez y confiabilidad y las técnicas para el análisis de datos".

A) Enfoque de la Investigación

La presente investigación adopta un enfoque cualitativo, dado que busca comprender en profundidad un fenómeno sociotécnico complejo dentro de su contexto real. La investigación cualitativa se orienta a interpretar significados y perspectivas en contextos específicos, privilegiando la comprensión sobre la cuantificación (Martínez-Salgado, 2012). Se busca explorar las percepciones y experiencias de los actores clave involucrados, reconociendo que el conocimiento válido es contextual y emergente. El estudio se plantea con un alcance exploratorio-descriptivo. Según (Sampieri, 2020) define la investigación exploratoria como "los estudios exploratorios que sirven para familiarizarnos con fenómenos relativamente desconocidos" por otra parte (Sampieri, 2020) menciona, que la investigación descriptiva recoge datos de manera independiente sobre las variables analizadas sin establecer relaciones entre ellas. (Beck, 2012) señala que la investigación exploratoria ilumina la naturaleza de fenómenos novedales o poco estudiados, y da espacio a que los participantes contribuyan con conocimiento nuevo. A la vez, se incorpora un componente descriptivo cualitativo, enfocado en documentar y retratar con

precisión “lo que está ocurriendo” con la adopción de IA en las organizaciones, desde la perspectiva de sus protagonistas. Según (Sandelowski, 1993) los estudios descriptivos buscan dilucidar el “quién, qué y dónde” de los eventos o experiencias, mientras que la faceta exploratoria amplía este panorama hacia el “cómo” y el “por qué” subyacentes.

B) Sujetos y fuentes de información

1) Sujetos

La población objetivo del estudio está compuesta por profesionales con roles de liderazgo o gestión en fábricas de software, quienes estén involucrados en decisiones o proyectos. En particular, se considera como perfil de participantes a Delivery Managers, IT Managers, CIOs (Chief Information Officers), Team Leads y Project Managers (PMs) de dichas organizaciones. Estas personas ocupan posiciones estratégicas desde donde pueden proporcionar información rica sobre prácticas actuales de IA, decisiones estratégicas, obstáculos y facilitadores en la adopción de estas tecnologías dentro del proceso de desarrollo de software. La selección de estos perfiles responde a la necesidad de involucrar informantes clave que posean visión panorámica del tema en sus empresas, por ejemplo, un CIO podrá hablar de la estrategia global de IA, mientras que un Team Lead aportará la experiencia operacional del equipo de desarrollo. En la investigación cualitativa, la elección de participantes no sigue criterios estadísticos sino de relevancia y profundidad de la información (Martínez-Salgado, El muestreo en investigación cualitativa. Principios básicos y algunas controversias, 2012) . Por ello, se emplea un muestreo intencional-teórico (también llamado muestreo por juicio o muestreo deliberado), en el cual los sujetos se

seleccionan de forma deliberada en función de criterios teóricos y prácticos predefinidos. Este muestreo intencional asegura que cada participante tenga altas probabilidades de brindar datos profundos y detallados sobre el fenómeno de interés.

2) Fuentes de Información

La fuente principal de datos de la investigación serán las entrevistas semiestructuradas realizadas a los sujetos antes descritos, las cuales proporcionarán testimonios directos y detallados.

Para mantener la consonancia con el enfoque de estudio de casos, se recurrirá también a fuentes secundarias para complementar y triangular la información. Entre las fuentes secundarias consideradas se encuentran documentos y materiales organizacionales de las fábricas de software estudiadas (por ejemplo, políticas internas de adopción de nuevas tecnologías, guías o manuales técnicos sobre IA si existieran, reportes de proyectos piloto con IA, indicadores de desempeño históricos vinculados a DevOps, etc.), así como información pública relevante (sitios web corporativos, comunicados de prensa sobre iniciativas de IA, informes industriales o gubernamentales sobre transformación digital en software). El uso de documentos y registros como datos adicionales es común en estudios de caso, pues aportan contexto y permiten verificar ciertos hechos o prácticas reportadas en las entrevistas (Shenton, 2003). Por ejemplo, la revisión de documentación podría revelar qué tipos de herramientas de IA ha implementado oficialmente la empresa, o cuáles son los lineamientos estratégicos publicados, lo que se puede contrastar con lo dicho por los entrevistados. Asimismo, se apoyará la investigación en la literatura científica existente sobre el tema relacionadas con la adopción de IA en la industria del software, innovación tecnológica y

métricas de desempeño en DevOps. Estas fuentes bibliográficas secundarias que incluyen artículos, tesis y reportes especializados sirven para contextualizar los hallazgos y sustentar conceptualmente el análisis. Por ejemplo, estudios cualitativos previos han explorado la incorporación de asistentes de IA en el desarrollo de software (Kauppaymuthoo, 2023), y sus hallazgos sobre percepciones de los desarrolladores pueden ofrecer un marco de comparación y referencia.

C) Instrumentos y técnicas de recolección

Entrevistas semiestructuradas: El principal instrumento de recolección de datos será la entrevista semiestructurada, seleccionada por su eficacia para indagar temas complejos. Este tipo de entrevista se conducirá con base en un guion previo de preguntas abiertas, organizadas a ciertos ejes temáticos clave, pero permite al investigador adaptar el orden y hacer preguntas de seguimiento según las respuestas del entrevistado (Sybing, 2025). La semiestructura ofrece así “un equilibrio entre una interacción rígida que produce datos organizados y una conversación fluida que explora aspectos inesperados pero relevantes del fenómeno estudiado” (Sybing, 2025).

En el contexto de esta investigación, se diseñó un protocolo de entrevista dividido en tres ejes temáticos principales:

- Prácticas de IA en la fábrica de software (por ejemplo, uso actual de herramientas de Inteligencia Artificial o aprendizaje automático en el ciclo de desarrollo, casos de uso implementados, nivel de automatización inteligente en procesos de desarrollo, testing, deployment, etc.)
- Barreras y facilitadores para la adopción de IA (incluyendo desafíos técnicos, culturales, organizacionales, éticos o de formación que hayan

enfrentado, así como factores que han impulsado o acelerado la incorporación de IA)

- Métricas DORA y desempeño del desarrollo (indagando si y cómo la adopción de IA se refleja en métricas de rendimiento de DevOps, concretamente las cuatro métricas clave definidas por DORA: Frecuencia de Despliegue, Tiempo de ciclo de cambios (Lead Time), Tasa de fallos en cambios (Change Failure Rate) y Tiempo medio de recuperación.

Las entrevistas serán grabadas en audio/vídeo (previo consentimiento), para luego ser transcritas palabra por palabra. La duración aproximada de cada entrevista oscila entre 15 y 30 minutos, dependiendo de la disponibilidad y nivel de detalle provisto por el participante.

Observación in situ y bitácora de campo: Como complemento a las entrevistas, cuando las condiciones lo permitan, se realizó observación no participante en las organizaciones (o en entornos de trabajo virtual) con el objetivo de registrar de primera mano cómo se integran (o no) las herramientas de IA en el flujo de desarrollo diario.

Dado que algunas fábricas de software ofrecen la posibilidad de visitas o sesiones de observación de equipo, se coordinarán sesiones breves de equipo, demostraciones de herramientas o revisiones del entorno de desarrollo para ver ejemplos tangibles del uso de IA. Este componente observacional se complementará de manera oportunista (es decir, se aprovechará cuando algún participante extienda la invitación o facilite el acceso). La observación directa es uno de los seis tipos de evidencia sugeridos por (Yin, 2009) en estudios de caso, y aportará contexto adicional a lo expresado en entrevistas, permitiendo anotar comportamientos, reacciones o dinámicas que quizás los participantes no verbalizan explícitamente.

Toda la información observada se registrará en una bitácora de campo (notas de observación), describiendo las situaciones, interacciones relevantes,

ambiente tecnológico, etc, siempre respetando la confidencialidad (no se incluirán identificadores reales de personas o código fuente de los propietarios). Estas notas de campo posteriormente se utilizarán para triangular con los datos de entrevista.

1) Procedimientos

1.1 Reclutamiento de participantes

Los sujetos de estudio serán contactados mediante estrategias de muestreo intencional apoyadas en redes profesionales. En primera instancia, se utilizará la plataforma LinkedIn para identificar y contactar a potenciales participantes que reunieran el perfil deseado filtrando por ubicación geográfica en países de Latinoamérica. Se les enviará un mensaje personalizado explicando de manera breve el objetivo del estudio y solicitando su participación voluntaria. Paralelamente, se solicitará la colaboración con el capítulo local del Project Management Institute (PMI) de Costa Rica, que servirá como puerta de entrada a una comunidad de gerentes de proyecto y líderes de TI. También se aprovecharán contactos académicos y referencias personales: a los primeros participantes reclutados se les pedirá referir con colegas de perfil similar. Se procura la diversidad en la selección: por ejemplo, participantes de diferentes países (incluir representantes de al menos cuatro países de LATAM).

1.2 Recolección de datos

Dada la dispersión geográfica de los casos y las restricciones de tiempo y movilidad, la mayor parte de las entrevistas se realizará de forma remota,

mediante videollamadas (utilizando herramientas como Zoom, Microsoft Teams o Google Meet, según la preferencia del entrevistado).

Al inicio de cada sesión, se presentará nuevamente el propósito del estudio y se asegurará el consentimiento informado verbal (en adición al consentimiento escrito/digital previamente enviado y firmado electrónicamente). Con autorización expresa, las entrevistas serán grabadas en audio y video para su posterior análisis.

La administración del cuestionario se realizará de forma remota mediante Google Forms: a cada potencial participante que no pudiera agendar una entrevista sincrónica se le envía un enlace único al formulario, invitándolo a responder. Esta estrategia complementaria permite mantener la cobertura de la muestra del estudio sin introducir sesgos significativos, ya que las preguntas reflejan el contenido de la guía de entrevista original. Las respuestas recopiladas por esta vía serán analizadas con técnicas cualitativas de forma similar a las transcripciones de entrevistas, y las preguntas cerradas proporcionarán datos adicionales para detectar patrones generales en la adopción de IA.

1.2.1 Apoyo tecnológico mediante herramientas de IA

Ingeniería de prompts con ChatGPT

Para optimizar la formulación de las preguntas de la entrevista y del cuestionario mixto se utilizó ChatGPT (OpenAI GPT-4) como asistente de prompt engineering. El equipo investigador iteró sobre borradores iniciales de las preguntas, refinándolos con la ayuda del modelo hasta lograr enunciados claros, neutros y culturalmente adecuados. El uso de ChatGPT permitió:

- Reescritura y prueba rápida de variantes de cada ítem, simulando respuestas de un perfil objetivo para verificar comprensibilidad.
- Detección de ambigüedades o sesgos en el lenguaje utilizado.
- Ajuste de tono según el rol (gerencial / técnico) del participante.

La literatura aconseja esta práctica: diseñar un buen prompt equivale a “programar” el modelo para obtener salidas coherentes con la intención del investigador, por lo que la ingeniería de prompts se ha convertido en una habilidad metodológica clave en estudios asistidos por IA (OpenAI, 2025).

1.2.1.2) Revisión bibliográfica asistida con NotebookLM y Consensus

Para complementar la fase de búsqueda y síntesis de literatura, se emplearon dos herramientas especializadas basadas en modelos de lenguaje:

- NotebookLM (Google AI) – actúa como cuaderno de investigación que resume y conecta documentos cargados por el usuario (PDF, HTML, notas). Facilita la extracción de ideas clave, la generación de preguntas y la elaboración de resúmenes temáticos, acelerando la construcción del estado del arte (NotebookLM, 2025) (Johnson, 2024).
- Consensus – motor de búsqueda académico con IA que interroga más de 200 millones de artículos revisados por pares y devuelve síntesis con citas trazables, reduciendo el tiempo necesario para identificar evidencias empíricas relevantes (Consensus, 2024).

Ambas herramientas se integraron al flujo de trabajo de revisión sistemática: los términos de búsqueda se filtraron inicialmente en Consensus; los artículos preseleccionados se exportaron a NotebookLM, donde se

generaron resúmenes comparativos y mapas de temas. Este procedimiento permitió garantizar cobertura reciente y trazabilidad de fuentes.

TFG: Adopción estratégica de la Inteligencia Artificial en fábricas de software

Empresa	Herramienta de IA	Modelo	Prompt	Resultado
OpenAI	ChatGPT	o3	<i>Haz el rol de un ingeniero de prompt y mejora el siguiente prompt para notebooklm: haz el rol de un experto investigador con experiencia en IA y en gestión de proyectos y búscame referencias de artículos con al menos 5 años de antigüedad relacionados con al tema de uso y adopción de IA en Fabricas de Software</i> <i>Haz el rol de un ingeniero de prompt y mejora el siguiente prompt para Consensus: haz el rol de un experto investigador con experiencia en IA y en gestión de proyectos y búscame referencias de artículos con al menos 5 años de antigüedad relacionados con al tema de uso y adopción de IA en Fabricas de Software</i> <i>Haz el rol de un ingeniero de prompt y mejora el siguiente prompt para utilizarlo en las plataformas de NobookLM y en la de Consensus: Haz el rol de un experto investigador con experiencia en IA y en gestión de proyectos y búscame referencias de artículos con al menos 5 años de antigüedad</i>	Actúa como un investigador senior especializado en inteligencia artificial y gestión de proyectos. Elabora una revisión bibliográfica sobre el uso y la adopción de la IA en fábricas de software (software factories). 1. Alcance temporal: incluye solo artículos académicos publicados hace $\geq$ 5 años. 2. Tipo de fuente: revistas arbitradas, conferencias indexadas y libros académicos. 3. Resultado esperado: entrega una tabla con: • Título completo • Autores • Año • Fuente (revista/conferencia/libro) • DOI o URL • Resumen de 2-3 líneas sobre los hallazgos clave. 4. Criterios de calidad: prioriza estudios empíricos o revisiones sistemáticas con alto número de citas. 5. Formato de salida: Markdown compatible con NotebookLM. Rol: Investigador senior especializado en inteligencia artificial y gestión de proyectos. Tarea: Identificar literatura académica sobre la adopción y el uso de IA en "fábricas de software". Criterios de búsqueda Antigüedad mínima: Publicaciones de ($\geq$ 5 años). Tipo de fuente: Revistas revisadas por

			<i>relacionados con al tema de impacto del uso de AI y su huella de carbono</i>	pares, actas de conferencias indexadas y capítulos de libros académicos. Enfoque temático: Estudios que aborden estrategias de adopción, factores críticos de éxito, métricas de desempeño o casos empíricos de IA aplicada al ciclo de vida de desarrollo en fábricas de software. Formato de salida Lista en Markdown con: Título – Autores – Año – Fuente – DOI/URL – Resumen breve (2-3 líneas). Ordena por relevancia (número de citas e impacto en la disciplina). Exclusiones: Descartar artículos que se centren solo en aspectos teóricos de IA sin relación con procesos industriales de desarrollo de software. Rol: Investigador senior especializado en Inteligencia Artificial y gestión de proyectos. Tarea: Localizar literatura académica revisada por pares publicada en 2019 o antes (≥ 5 años) que analice la huella de carbono / impacto ambiental de la IA (p. ej., consumo energético, emisiones de CO₂, uso de agua para refrigeración, estrategias de mitigación). Entrega solicitada: 1. Al menos 10 referencias (artículos de revista o congresos). 2. Formato APA 7 con DOI o URL.
--	--	--	--	--

TFG: Adopción estratégica de la Inteligencia Artificial en fábricas de software

				3. Una frase-síntesis (máx. 25 palabras) sobre el hallazgo principal de cada estudio. 4. Ordenar por relevancia o factor de impacto.
--	--	--	--	--

Google	NotebookLM	Gemini	<i>Actúa como un investigador senior especializado en inteligencia artificial y gestión de proyectos.</i> <i>Elabora una revisión bibliográfica sobre el uso y la adopción de la IA en fábricas de software (software factories).</i> <i>1. Alcance temporal: incluye solo artículos académicos publicados hace ≥ 5 años.</i> <i>2. Tipo de fuente: revistas arbitradas, conferencias indexadas y libros académicos.</i> <i>3. Resultado esperado: entrega una tabla con:</i> • Título completo • Autores • Año • Fuente (revista/conferencia/libro) • DOI o URL • Resumen de 2-3 líneas sobre los hallazgos clave. <i>4. Criterios de calidad: prioriza estudios empíricos o revisiones sistemáticas con alto número de citas.</i> <i>5. Formato de Salida: Markdown compatible con NotebookLM.</i>	• The Impact of Artificial Intelligence on Project Management. • Artificial Intelligence Adoption in Software Development, A Qualitative Study on Developers' Views
--------	------------	--------	--	--

Consensus	NotebookLM	Consensus o3	<i>Actúa como un investigador senior especializado en inteligencia artificial y gestión de proyectos.</i> <i>Elabora una revisión bibliográfica sobre el uso y la adopción de la IA en fábricas de software (software factories).</i> <i>1. Alcance temporal: incluye solo artículos académicos publicados hace ≥ 5 años.</i> <i>2. Tipo de fuente: revistas arbitradas, conferencias indexadas y libros académicos.</i> <i>3. Resultado esperado: entrega una tabla con:</i> <i>• Título completo</i> <i>• Autores</i> <i>• Año</i> <i>• Fuente (revista/conferencia/libro)</i> <i>• DOI o URL</i> <i>• Resumen de 2-3 líneas sobre los hallazgos clave.</i> <i>4. Criterios de calidad: prioriza estudios empíricos o revisiones sistemáticas con alto número de citas.</i> <i>5. Formato de salida: Markdown compatible con NotebookLM.</i>	• Adopción y uso de herramientas de IA: una agenda de investigación basada en UTAUT. • Requisitos previos para la adopción de tecnologías de IA en la manufactura: evidencia de una muestra mundial de empresas manufactureras. • Inteligencia artificial para la industria 4.0: revisión sistemática de aplicaciones, desafíos y oportunidades • Inteligencia artificial y productividad a nivel de empresa. • Adopción de IA en Estados Unidos: quién, qué y dónde • Estudio del Uso de Herramientas de IA para la Gestión de Proyectos de Software, como Instrumento para Mejorar la Seguridad de las Aplicaciones. Corrección: Exploración de los factores que impulsan la adopción de IA en la producción: una revisión sistemática de la literatura y una agenda de investigación futura
-----------	------------	-----------------	--	---

Tabla 2. Generación de Generación de Prompts

CAPÍTULO IV. Diagnóstico y análisis de resultado

A) Abstracts de los Artículos Seleccionados

1) Venkatesh (2021) – Adoption and Use of AI Tools: A Research Agenda Grounded in UTAUT

Este artículo plantea que, pese a la rápida disponibilidad de herramientas de IA, su adopción efectiva todavía es incomprendida. Basándose en la teoría UTAUT (Teoría Unificada de la Aceptación y Uso de la Tecnología), el autor clasifica los determinantes de adopción en características individuales, tecnológicas y ambientales, e identifica intervenciones organizacionales (alfabetización, liderazgo visible, políticas de datos) para impulsar la aceptación. Propone 26 líneas de investigación que conectan constructos UTAUT clásicos con variables emergentes como confianza algorítmica y ética percibida, ofreciendo un mapa conceptual para estudios futuros y para diagnósticos organizacionales (Venkatesh, 2021) .

2) Kauppaymuthoo & Collins (2023) – Artificial Intelligence Adoption in Software Development: A Qualitative Study on Developers' Views

Mediante entrevistas temáticas a 17 desarrolladores, este trabajo descubre cinco núcleos discursivos sobre IA: (a) confianza condicional (verifican todo el código generado), (b) empoderamiento y productividad (incrementos de ~25 % en velocidad), (c) impacto laboral (redefinición de tareas, no sustitución inmediata), (d) preocupaciones éticas (plagio, licencias y sesgos) y (e) barreras técnicas (lenguajes minoritarios, datos internos). Los autores concluyen que la aceptación depende de combinar guías corporativas claras con formación continua para elevar la comprensión y reducir la ansiedad

(Kauppaymuthoo, Artificial Intelligence Adoption in Software Development, A Qualitative Study on Developers' Views, 2023).

3) Carrillo Zenteno et al. (2024) – The Impact of Artificial Intelligence on Project Management

A partir de una encuesta mixta a 145 profesionales ecuatorianos, el estudio constata que el 72 % percibe reducción de tiempos y mejora en la precisión de estimaciones gracias a IA; existe correlación positiva ($r = 0,63$) entre uso de IA y exactitud de costos. No obstante, identifica barreras de presupuesto (37 %), capacidad tecnológica (29 %) y apoyo directivo (24 %). El 83 % espera un aumento “significativo” de adopción en cinco años, aunque advierte sobre la necesidad de estrategias integrales de cambio y capacitación (José Antonio Carrillo Zenteno, 2024).

4) Wu et al., 2022 Sustainable AI: Environmental Implications, Challenges and Opportunities

Este artículo disecciona la huella de carbono de la IA a lo largo de todo su ciclo de vida: datos, algoritmos y hardware. El equipo cuantifica tanto las emisiones operativas durante el entrenamiento e inferencia como las derivadas de fabricar y sustituir servidores y GPUs. Propone tres líneas de acción: co-diseño hardware-software para frenar la curva de consumo energético, migración dinámica de cargas a centros de datos abastecidos con renovables y métricas transparentes que permitan comparar modelos por “gramos CO₂e por inferencia”. El artículo concluye que, si no se controla el

“efecto rebote” —modelos cada vez más grandes—, las ganancias de eficiencia se diluirán (Carole-Jean Wu, 2022).

5) Ligozat et al., 2022 — “Unraveling the Hidden Environmental Impacts of AI Solutions”

Los autores sostienen que medir solo la energía de entrenamiento es quedarse en la superficie. Emplean Análisis de Ciclo de Vida (ACV) para capturar impactos directos e indirectos de una solución de IA, adquisición de datos, transporte, almacenamiento, hardware y fin de vida, y demuestran, con casos reales, que entrenar en regiones con electricidad limpia puede reducir cien veces las emisiones frente a hacerlo en zonas intensivas en carbón. También alertan sobre los efectos rebote: si cada predicción es más barata, el volumen total de inferencias puede dispararse y anular los beneficios. El artículo cierra con una guía práctica para aplicar ACV en servicios de IA corporativos.

6) Nishant, Kennedy & Corbett, 2020 — “Artificial Intelligence for Sustainability: Challenges, Opportunities and a Research Agenda”

Este ensayo revisa cómo la IA podría recortar el uso de energía, agua y materiales en la economía, pero señala cinco peligros que toda empresa debe vigilar: dependencia de datos históricos que perpetúan sesgos, respuestas humanas impredecibles a las intervenciones, riesgos de ciberseguridad, impactos colaterales no deseados y la dificultad de medir resultados medioambientales en el largo plazo. Propone un programa de investigación

que combine perspectivas multiescala, dinámicas de sistemas y diseño centrado en las personas para asegurar que las soluciones de IA entreguen valor ambiental neto.

B) Resultados de la Revisión de la Literatura

1) Factores críticos de adopción

Los tres primeros artículos coinciden en que la confianza en la IA y la transparencia algorítmica son requisitos para que los usuarios la incorporen en su flujo de trabajo. Esto se alinea con la encuesta global de (McKinsey Global Survey, 2024), donde menos de la mitad de las empresas afirma mitigar riesgos de imprecisión, subrayando la importancia de políticas de gobernanza y auditoría ética.

2) Beneficios tangibles

La evidencia empírica muestra incrementos de productividad de 25 %-30 % (desarrolladores) y mejoras en precisión de estimaciones y reducción de retrabajo (gestión de proyectos). Estos hallazgos son coherentes con el State of DevOps Report 2024, que asocia un 25 % de adopción de IA con mejoras en satisfacción laboral y caída del burnout. (Google, 2025) (DORA, 2024) Además, un sondeo de (Insider, 2025) expone que el 62 % de ingenieros declara un aumento ≥ 25 % en velocidad gracias a Copilot y herramientas afines.

3) Barreras persistentes

Las tres primeras investigaciones remarcan obstáculos de capacidad tecnológica, presupuesto y skills. Un informe de (BCG, 2024) confirma que el 74 % de las empresas lucha por escalar valor debido, en 70 % de los casos, a problemas de procesos y personas, no de tecnología. Este hallazgo refrenda la necesidad de un plan de gestión del cambio y de capacitación continúa incluidos en los Objetivos 2.3 y 2.4 de la investigación.

4) Análisis del impacto de la AI en el medio ambiente

La revisión confirma que la huella ambiental debe sentarse en la misma sala donde se discuten costo, tiempo y alcance. Si nuestro modelo de adopción estratégica de IA integra ACV (Análisis de Ciclo de Vida), presupuestos de carbono, métricas verdes junto a las de DevOps y un programa de capacitación sostenible, las fábricas de software podrán seguir innovando con IA y, al mismo tiempo, reforzar su compromiso de dejar una huella positiva en el planeta.

5) Contribución al modelo de gestión propuesto

- Diagnóstico inicial: El marco UTAUT-IA de Venkatesh provee dimensiones y constructos medibles para evaluar preparación organizacional, directamente útiles para la meta 2.1.
- Gestión del cambio: Las percepciones de desarrolladores identificadas por Kauppaymuthoo orientan acciones de comunicación, formación y

diseño de guías éticas, fortaleciendo la fase de implementación del modelo (meta 2.2).

- Indicadores de negocio: Las métricas de impacto en costo, tiempo y calidad de Carrillo ofrecen referencias cuantitativas para el tablero de control del plan piloto, vinculándolo con las métricas DORA de DevOps.
- Alineamiento con buenas prácticas: La convergencia con reportes globales (McKinsey, DORA, BCG) valida la relevancia internacional del marco y justifica su aplicación en fábricas de software latinoamericanas.
- Métricas verdes junto a métricas de negocio: Al colocar gramos de CO₂e al lado de lead-time o MTTR (Tiempo Medio de Reparación), la fábrica de software ve en el mismo panel si un sprint fue rápido y responsable, o solo rápido.

C) Resultados del Cuestionario

1) Perfil de los participantes

Se obtuvo respuesta de 25 participantes de los 50 invitados a colaborar, la mayoría con puestos de nivel medio a alto en sus organizaciones: directores (CEO, CTO, CIO), gerentes de TI/desarrollo, líderes de equipo (Team Leads) y directores de proyecto. Esta composición de la muestra asegura que los hallazgos reflejen perspectivas estratégicas e informadas sobre la adopción de IA, dado que casi todos los encuestados ocupan posiciones de toma de decisión en sus empresas. En cuanto a la distribución geográfica, si bien el cuestionario se difundió principalmente en Latinoamérica (vía LinkedIn y capítulos del PMI latinoamericanos), los resultados se analizaron de forma agregada sin segmentar por país, enfocándose en las tendencias generales del sector. A continuación, se presentan los hallazgos del diagnóstico,

organizados en cuatro ejes temáticos: prácticas actuales de IA, barreras y facilitadores, impacto y métricas, y sostenibilidad de la IA. Cada sección resume las respuestas clave del cuestionario e interpreta su significado en relación con el problema de investigación.

2) Sección I - Prácticas actuales de IA

2.1 ¿Qué herramientas o tecnologías de IA utilizan hoy en el ciclo de vida del software?

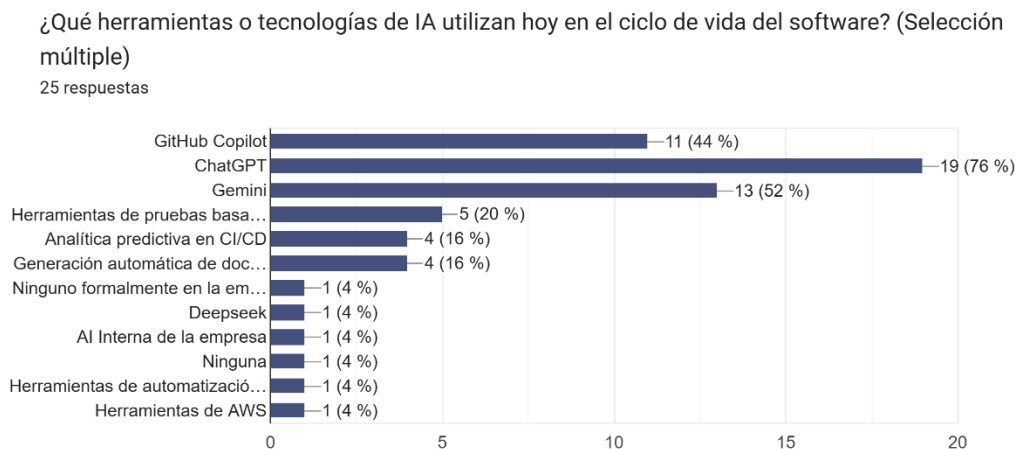


Figura 1. ¿Qué herramientas o tecnologías de IA utilizan hoy en el ciclo de vida del software?

La gran mayoría de los participantes emplea herramientas de IA generativa, siendo ChatGPT la más mencionada (usada por 76% de los encuestados), en segundo lugar, Gemini (52%), seguida por asistentes de desarrollo como GitHub Copilot (44%). También se citaron otras tecnologías en menor medida, como soluciones de analítica predictiva en pipelines CI/CD. Solo un caso aislado indicó no usar ninguna herramienta de IA.

Este hallazgo implica que las fábricas de software ya están experimentando activamente con IA, principalmente mediante asistentes conversacionales y de código, lo cual sienta una base tecnológica sobre la cual construir una adopción más estratégica. La omnipresencia de ChatGPT sugiere que las soluciones de IA de propósito general son un punto de entrada popular, probablemente por su facilidad de acceso y amplio rango de aplicaciones (desde generación de código hasta soporte en documentación).

2.2 Frecuencia promedio de uso de IA en el trabajo diario

Frecuencia promedio de uso de IA en el trabajo diario (Selección única)
25 respuestas

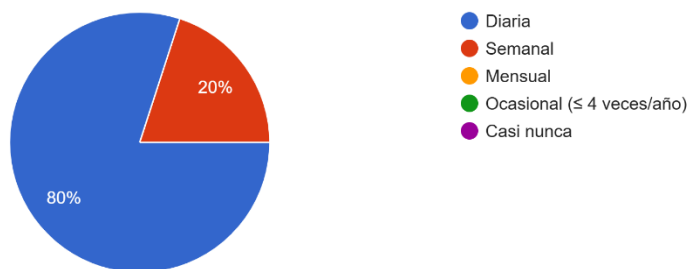


Figura 2. Frecuencia promedio de uso de IA en el trabajo diario

Se refleja que el uso de IA está integrado al día a día el 80% de encuestados reportan utilizar herramientas de IA a diario en sus tareas laborales, y el restante 20% lo hace al menos cada semana. Ningún participante se ubicó en frecuencias mensuales o “raras veces”. Como hallazgo este dato clave denota que, entre quienes han adoptado IA, hay una alta dependencia cotidiana de estas herramientas, lo que implica que la IA ya está engranada en los flujos de trabajo normales. Desde la perspectiva del entrevistado, esta frecuencia se refleja en que “todos en el equipo tienen

abierta alguna herramienta de IA durante la jornada”, indicando normalización de la IA como asistente constante. Para la investigación, esto sugiere que cualquier modelo de gestión propuesto debe asumir que las herramientas IA formarán parte rutinaria del proceso productivo, y por tanto deberá asegurar su uso eficiente y consistente.

2.3 Etapas del ciclo de desarrollo donde se integra la IA

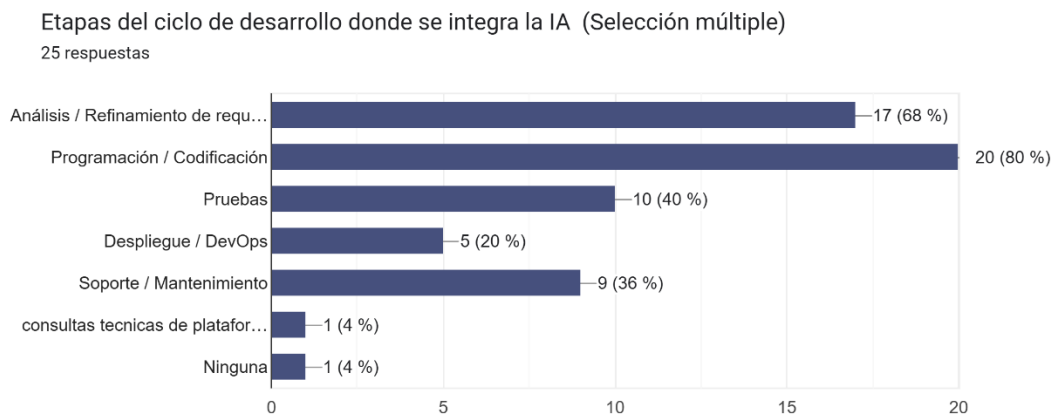


Figura 3. Etapas del ciclo de desarrollo donde se integra la IA

Al consultar en qué fases concretas se aplica la IA, los encuestados señalan principalmente la programación/codificación un 80% y el análisis o refinamiento de requisitos en un 68% como las etapas donde más integran estas tecnologías. También un 40% indicó usar IA en pruebas de software, y una proporción similar un 36% en soporte/mantenimiento. El 20% integran Despliegue/DevOps, reflejando que áreas como la optimización de pipelines o monitoreo inteligente aún son incipientes. Solo un participante respondió no emplear IA en ninguna etapa del ciclo. Como hallazgo este patrón indica que la IA se está aprovechando sobre todo en las fases creativas y de construcción,

mientras que las fases posteriores (implantación, operaciones) presentan una oportunidad de mejora para la adopción futura. El entrevistado corroboró esta distribución: en su fábrica han visto beneficios inmediatos al usar IA durante la codificación y la comprensión de requisitos, pero reconoció que “todavía no hemos aplicado IA para automatizar despliegues o en producción; ahí tenemos campo por explorar”. Se puede evidenciar que las fábricas de software han comenzado por las tareas de desarrollo donde es más fácil integrarla (asistencia en coding, generación de algoritmos, etc.), dejando para etapas posteriores su expansión hacia DevOps y mantenimiento. Esto implica que el modelo de adopción estratégica deberá considerar fases graduales, empezando por áreas con resultados rápidos para luego extender la IA a procesos más avanzados conforme madure la confianza y experiencia en estas herramientas.

3) Sección II - Barreras y facilitadores

Los datos revelan un panorama dual: por un lado, existen obstáculos claros (principalmente brechas de conocimiento y preocupaciones de seguridad) que dificultan la implementación de IA; por otro, hay factores internos (cultura innovadora y liderazgo) que están allanando el camino para dicha adopción. El apoyo institucional es percibido como alto en la mayoría de los casos, aunque las políticas formales de IA responsable aún brillan por su ausencia en muchas organizaciones. El directivo entrevistado ilustró esta dicotomía: si bien su empresa cuenta con respaldo de la alta gerencia para proyectos de IA, han enfrentado resistencias iniciales y están “aprendiendo sobre la marcha, sin un manual corporativo todavía”.

3.1 Obstáculos principales para adoptar IA

Obstáculos principales para adoptar IA (Selección múltiple)

25 respuestas

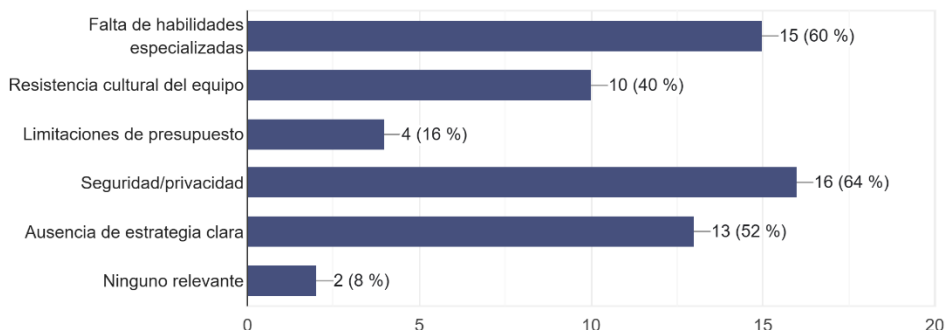


Figura 4. Obstáculos principales para adoptar IA

Las dos barreras más citadas por los encuestados son la falta de habilidades especializadas en IA (60%) y las preocupaciones de seguridad/privacidad de los datos (64%) al usar estas tecnologías. También más de la mitad percibe una ausencia de estrategia clara para IA en su organización (52%). Asimismo, un 40% destaca la resistencia cultural del equipo (por ejemplo: desconfianza o temor al cambio por parte de desarrolladores) como obstáculo. Las limitaciones de presupuesto resultaron ser menos frecuentes (solo 16% las mencionan) y un pequeño 8% opina que “no hay barreras relevantes”. Como hallazgo los obstáculos sugieren que los impedimentos para adoptar IA en fábricas de software no son tanto financieros, sino principalmente humanos y organizativos. En otras palabras, el conocimiento y la confianza son la brecha crítica: las empresas temen riesgos (por ejemplo: exponer datos sensibles a herramientas externas) y carecen de personal formado o directrices para mitigar dichos riesgos. El entrevistado confirmó esta visión: en su caso, “lo primero fue vencer el recelo del equipo y admitir que necesitábamos capacitación; también definimos protocolos para no alimentar la IA con código confidencial”. En la investigación

podemos entender que un modelo de gestión efectivo deberá incluir componentes de capacitación y gestión del cambio, así como lineamientos de gobernanza de datos, para superar estas barreras fundamentales de adopción. La ausencia de una estrategia señala que muchas empresas están abordando la IA de forma ad hoc; por ello, estructurar un plan estratégico (objetivo general de este proyecto) atendería directamente uno de los vacíos más señalados.

3.2) Factores que más han facilitado la adopción

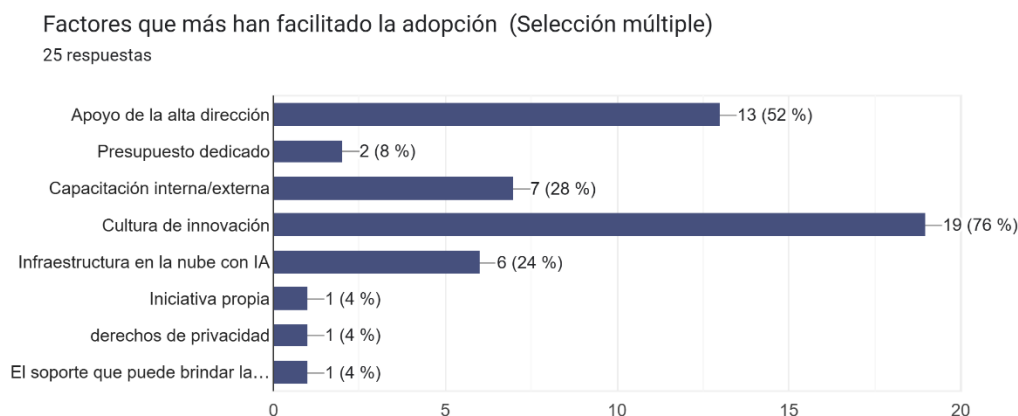


Figura 5. Factores que más han facilitado la adopción

Los principales facilitadores internos reportados giran en torno a la cultura y el liderazgo. El 76% de los participantes atribuye la adopción de IA a poseer una cultura de innovación fuerte en la organización y un 52% señala el apoyo de la alta dirección como clave. En menor medida, un 28% reconoce que la capacitación interna/externa específica en IA ha sido un habilitador, y un 24% menciona la existencia de infraestructura en la nube con capacidades de IA (por ejemplo: plataformas listas para usar servicios cognitivos) como

factor facilitador. Solo unos pocos identifican haber tenido presupuesto dedicado para IA (8%). Los principales hallazgos implican que, para lograr adoptar IA estratégicamente, el entorno organizacional debe ser fértil en actitud innovadora y contar con patrocinadores en “la cúpula”. La entrevista respaldó esto: “la directiva nos dio vía libre para experimentar con un prototipo de IA” y que la empresa promueve internamente la creatividad tecnológica, lo cual facilitó sus primeros proyectos.

El liderazgo y la cultura organizacional resultan ser palancas críticas: un modelo de adopción deberá recomendar acciones como campañas de sensibilización desde la alta gerencia y creación de entornos de experimentación controlada, para cultivar estos facilitadores donde no existan. La relativamente baja mención de presupuesto dedicado también sugiere que muchas fábricas de software están iniciando con proyectos piloto de bajo costo (frecuentemente usando herramientas gratuitas o versiones trial), por lo que formalizar inversión podría ser el siguiente paso conforme madure la iniciativa.

3.3) Nivel de apoyo institucional a las iniciativas de IA

Nivel de apoyo institucional a las iniciativas de IA (Selección única)

25 respuestas

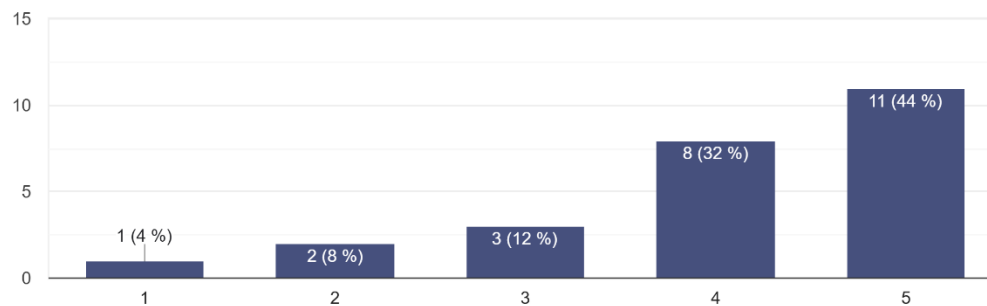


Figura 6. Nivel de apoyo institucional a las iniciativas de IA

La mayoría califica el apoyo de sus empresas a la IA de alto o muy alto. Específicamente, el 76% otorgaron una puntuación de 4 o 5 (en una escala de 1 a 5) al nivel de respaldo institucional hacia iniciativas de IA. Dentro de este grupo, un 44% indicó el máximo (nivel 5, apoyo “muy fuerte”). Tan solo un 12% percibe apoyo bajo (niveles 1–2). El hallazgo refuerza que, al menos entre los participantes, existe un consenso positivo sobre la voluntad institucional de adoptar IA, lo cual es un punto de partida alentador. Implica que en la mayoría de los casos no hay impedimento desde la dirección para avanzar con IA; por el contrario, probablemente hay interés estratégico. Sin embargo, cabe cruzar este dato con el de “ausencia de estrategia clara” mencionado en obstáculos: es posible que, aunque las altas gerencias estén dispuestas (buen apoyo), aún no hayan articulado una hoja de ruta definida. El entrevistado, por ejemplo, afirmó que sus directivos están entusiasmados con la IA “pero nos han dejado a nosotros (TI) proponer cómo usarla; nos apoyan, aunque ellos mismos no tienen el plan detallado”. Esto sugiere que el apoyo debe canalizarse en estructura; el modelo de gestión propuesto puede ayudar proporcionando esa directriz que los líderes están dispuestos a respaldar. Existe capital político a favor de la IA en estas organizaciones, y el

desafío es aprovecharlo eficazmente antes de que se diluya o se use en esfuerzos sin coordinación.

3.4) ¿La organización cuenta con políticas o lineamientos sobre uso responsable/ético de IA?

¿La organización cuenta con políticas o lineamientos sobre uso responsable/ético de IA?
(Selección única)
25 respuestas

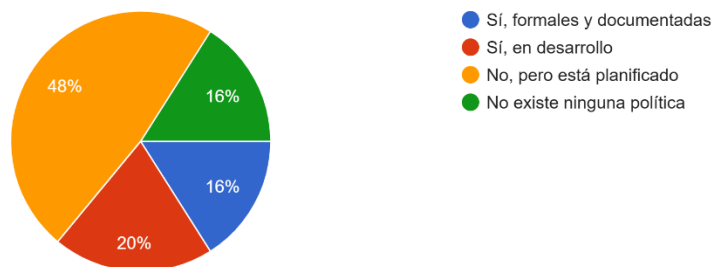


Figura 7. ¿La organización cuenta con políticas o lineamientos sobre uso responsable/ético de IA?

Casi la mitad de las organizaciones no cuenta aún con políticas de IA ética, aunque están en proceso de desarrollarlas (48%), y un 16% adicional no tiene ninguna iniciativa ni lineamiento al respecto. Solo aproximadamente un tercio dispone de algo concreto: 20% indicó que tienen directrices “en desarrollo” (borradores o guías informales) y únicamente 16% posee políticas formales y documentadas vigentes sobre uso responsable de IA. Este hallazgo evidencia una brecha de gobernanza: la adopción tecnológica ha superado a la creación de reglas y controles internos, en otras palabras, muchas fábricas de software están usando IA sin un marco normativo claro que aborde aspectos como privacidad de datos, propiedad intelectual de los outputs generados, uso aceptable de asistentes de código, prevención de sesgos o

riesgos éticos. Desde la entrevista surgió una situación coincidente: el entrevistado admitió que en su empresa “hemos implementado IA antes de escribir las políticas; recién ahora estamos redactando guías porque nos dimos cuenta de que las necesitamos”. Para la investigación, esto significa que cualquier modelo de adopción estratégica deberá integrar explícitamente la dimensión de ética y políticas de IA como un componente prioritario. La falta de lineamientos no solo crea riesgo (uso inadecuado, potenciales problemas legales o reputacionales), sino que también puede alimentar algunos de los obstáculos ya mencionados (por ejemplo, la resistencia del equipo o las dudas de seguridad podrían mitigarse con políticas claras y capacitación asociada). Se identifica la necesidad de desarrollar e institucionalizar políticas de IA responsable en paralelo con la implementación técnica, de modo que la expansión de estas herramientas venga acompañada de la debida orientación y control.

4) Sección III – Impacto y métricas

Uno de los objetivos específicos de esta investigación es comprobar qué beneficios tangibles y mejoras en métricas se asocian a la introducción de IA en el proceso de desarrollo, así como entender la percepción general de su impacto. Los resultados de la encuesta muestran una percepción mayoritariamente positiva: la IA está rindiendo frutos en productividad y calidad según los participantes, aunque es notable que no todas las organizaciones están midiendo rigurosamente ese impacto (muchos no utilizan métricas formales como DORA). Durante la entrevista, el entrevistado corroboró que “hemos visto resultados favorables, como entregas más rápidas y menos errores humanos, aunque honestamente no llevamos números exactos; lo notamos de forma cualitativa”. Esta mezcla de entusiasmo por los beneficios y falta de medición sistemática es consistente con hallazgos de

otros estudios (como la encuesta global de McKinsey, donde muchas empresas reportan beneficios, pero menos de la mitad ha establecido métricas claras de éxito).

4.1) Beneficios tangibles percibidos tras adoptar IA

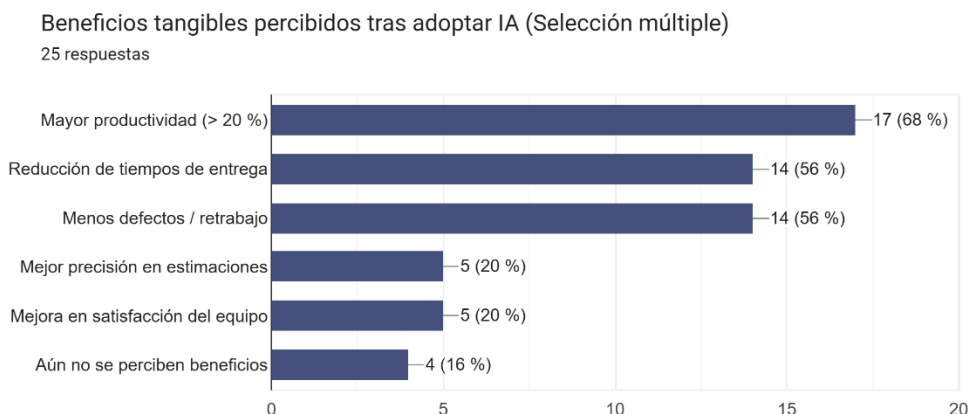


Figura 8. Beneficios tangibles percibidos tras adoptar IA

Un 68% afirma que la productividad de sus equipos aumentó significativamente (más de un 20% de mejora) tras incorporar herramientas de IA. Asimismo, más de la mitad nota reducción de defectos o retrabajo (56%) y acortamiento de los tiempos de entrega (56%) en sus proyectos, lo cual indica mejoras tanto en calidad como en velocidad. Adicionalmente, un 20% destaca mejoras en la satisfacción del equipo y otro 20% reporta mayor precisión en las estimaciones de esfuerzo/costo gracias a la IA. Solo un 16% manifiesta no haber percibido aún beneficios tangibles. Este hallazgo nos da un conjunto de resultados donde implica que la IA aporta valor real en las fábricas de software, sobre todo en eficiencia y calidad del desarrollo. El hecho de que productividad y menos errores encabecen la lista concuerda con

estudios recientes, por ejemplo, los líderes encuestados por (McKinsey Global Survey, 2024) atribuyen ya parte de sus ganancias operativas al uso de IA en ingeniería. El entrevistado ilustró estos beneficios señalando que en su último proyecto “logramos terminar dos sprints adelantados al calendario, en parte porque Copilot ayudó con fragmentos de código repetitivo, y además el código salió con menos bugs gracias a las sugerencias automáticas; el equipo estaba muy motivado”. Para la investigación, este hallazgo reafirma el caso de negocio de la IA: cuando se adopta estratégicamente, se pueden esperar mejoras medibles en KPIs clásicos de proyectos (tiempo, calidad, productividad).

4.2) Métricas en las que ha observado mejora gracias a IA

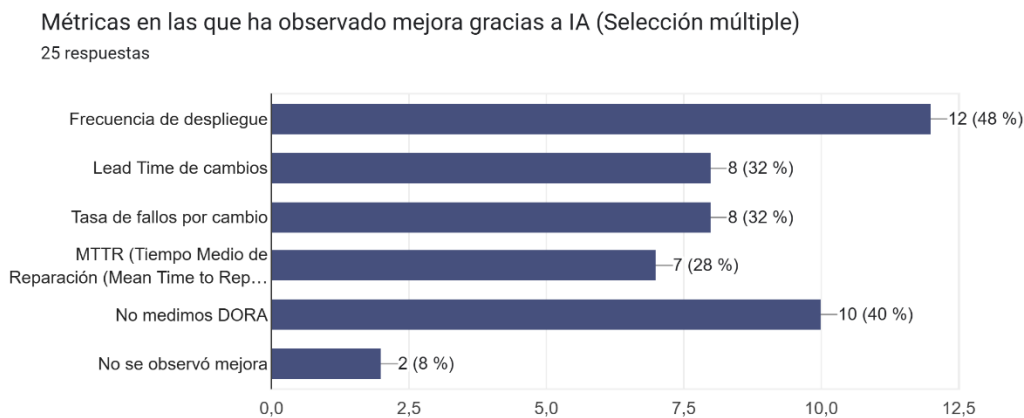


Figura 9. Métricas en las que ha observado mejora gracias a IA

Casi la mitad de los encuestados identificó mejoras en métricas operativas concretas gracias a la IA. El 48% observó que aumentó la frecuencia de despliegues. Alrededor de un tercio reporta mejoras en indicadores de calidad de entrega, como menor tasa de fallos por cambio

(32%), y de velocidad de ciclo, como reducción del Lead Time de cambios (32%) o del Tiempo medio de recuperación ante incidentes – MTTR (28%). Sin embargo, cabe resaltar que un 40% de las empresas admite no medir formalmente las métricas DORA, y además un pequeño 8% indicó que no han visto mejora en ninguna métrica hasta ahora. Este hallazgo revela una situación mixta: por un lado, allí donde se tienen métricas, se detectan señales positivas del impacto de la IA; pero, por otro lado, existe una falta de medición extensiva que deja potencialmente “a ciegas” a muchas organizaciones sobre cuánto mejora (o no) la IA sus procesos. De hecho, varios líderes marcan “No medimos DORA”, lo cual implica que no están evaluando objetivamente la eficacia de la IA en términos de entrega continua. El entrevistado encaja en esta realidad: “Sabemos que vamos más rápido, pero no hemos sacado los números. Deberíamos medir, quizá usando las métricas DevOps, para cuantificar la mejora”. Para la iniciativa de adopción estratégica, esto sugiere la necesidad de implantar prácticas de medición y seguimiento de KPIs asociados a la IA. Incluir en el modelo de gestión un componente de valoración cuantitativa permitiría justificar la inversión en IA con datos duros y afinar dónde está dando resultados (por ejemplo: si se comprueba que la IA mejora mucho la frecuencia de releases pero no tanto el MTTR, se podrían enfocar esfuerzos en robustecer la etapa de operación). La mejora en métricas como frecuencia de despliegue valida el potencial de la IA para acelerar ciclos de desarrollo, pero la falta de medición generalizada representa una oportunidad de mejora en sí misma para gestionar mejor la adopción.

4.2) Impacto global de la IA en el rendimiento del ciclo de desarrollo de software

Impacto global de la IA en el rendimiento del ciclo de desarrollo de software (Selección única)
25 respuestas

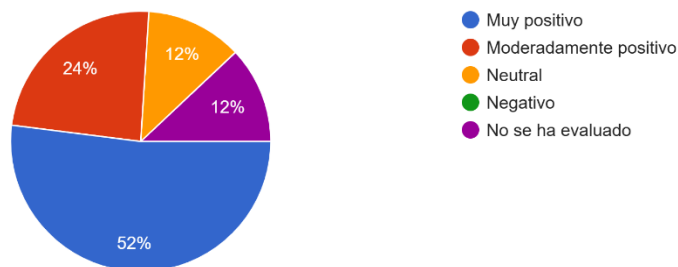


Figura 10. Impacto global de la IA en el rendimiento del ciclo de desarrollo de software

El 52% lo considera “muy positivo”, mientras un 24% lo ve “moderadamente positivo”. El resto se reparte entre un 12% que lo siente neutral (ni mejora ni empeora significativamente) y un 12% que indicó no haber evaluado aún el impacto de forma global.

Este hallazgo sugiere que, en su experiencia hasta ahora, las iniciativas de IA han contribuido favorablemente al desempeño de sus fábricas de software, al menos en percepción de quienes las lideran. El cero por ciento de valoraciones negativas implica que, si ha habido inconvenientes, no han sido lo suficientemente significativos como para opacar los beneficios (o que quienes tuvieron malas experiencias no están representados en la muestra). El entrevistado, por ejemplo, recalcó que “incluso con algunas salvedades, diría que la IA nos ha dado más de lo que nos ha quitado; volver atrás no está en nuestros planes”. Para la investigación, esta positividad general refuerza la hipótesis de que vale la pena adoptar IA estratégicamente, ya que los pioneros internos ya perciben mejoras. No obstante, el hecho de que un grupo admitiera no haber evaluado el impacto sugiere nuevamente falta de métricas sistemáticas o de revisiones post-implementación. Es importante implementar evaluaciones formales (por ejemplo: análisis de ROI de proyectos de IA,

encuestas de satisfacción posteriores, etc.), para tener un sustento objetivo que consolide esa confianza y permita comunicarla a todos los stakeholders.

El impacto percibido muy favorable proporciona un entorno propicio para la adopción estratégica (es más fácil abogar por la IA cuando ya muchos la ven con buenos ojos), pero también subraya la necesidad de institucionalizar ese aprendizaje positivo con métricas y retrospectivas, consolidando así la mejora continua basada en evidencia.

5) Sección IV – IA sostenible / Impacto ambiental

Se exploró el grado en que las fábricas de software están atendiendo la sostenibilidad ambiental en su incursión en la IA, un tema emergente que abarca el consumo energético de los modelos, las emisiones de carbono asociadas y las prácticas “verdes” en el ciclo de vida de la IA. Aquí los hallazgos son contundentes: la mayoría de las organizaciones aún no incorpora la sostenibilidad como parte de su adopción de IA, ya sea porque no miden ese impacto, no han tomado acciones al respecto, o porque ni clientes ni la propia estrategia interna lo han exigido hasta ahora. El entrevistado confirmó esta realidad, reconociendo que “hemos estado más enfocados en lograr que la IA funcione que en su huella de carbono; por ahora nuestros clientes tampoco preguntan, así que no ha sido prioridad”. No obstante, también emergen señales de cambio incipiente, con algunos casos aislados de medición y un puñado de empresas dándole alta importancia estratégica al tema ambiental.

5.1) ¿Su organización mide o monitorea el impacto ambiental (energía, carbono, agua) derivado del uso de IA?

¿Su organización mide o monitorea el impacto ambiental (energía, carbono, agua) derivado del uso de IA? (Selección única)

25 respuestas

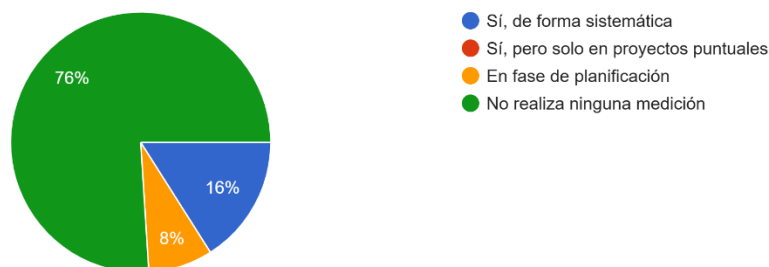


Figura 11. ¿Su organización mide o monitorea el impacto ambiental (energía, carbono, agua) derivado del uso de IA?

El 76% respondió que no realiza ninguna medición actualmente. Solo un 16% indicó que sí lo hace de forma sistemática, y el restante 8% señala que están en fase de planificar dicho monitoreo.

Este hallazgo muestra que tres de cada cuatro fábricas de software están “a oscuras” respecto al impacto ecológico de sus actividades de IA, lo que implica que la sostenibilidad aún no está integrada en sus procesos de gestión de la IA. En términos prácticos, la ausencia de datos dificulta que consideren mejoras o compensaciones ambientales, pues “lo que no se mide, no se gestiona”. El hecho de que 1 de cada 6 sí mida sistemáticamente sugiere que existen algunos pioneros posiblemente motivados por políticas corporativas verdes o por operar en regiones con regulaciones más estrictas. Para la mayoría, la prioridad ha sido adoptar IA por eficacia sin añadir por ahora la “capa verde” de seguimiento. Desde la perspectiva investigativa, esto implica que el modelo de adopción estratégica debería incluir la sostenibilidad como criterio desde etapas tempranas, para cerrar esta brecha, pero de una forma opcional. La experiencia de la industria (y literatura revisada) advierte que la huella de carbono de la IA puede crecer exponencialmente con modelos

más complejos, de modo que ignorarla podría acarrear costos futuros o choques reputacionales.

5.2) ¿Qué acciones concretas han implementado para reducir la huella ambiental de la IA?

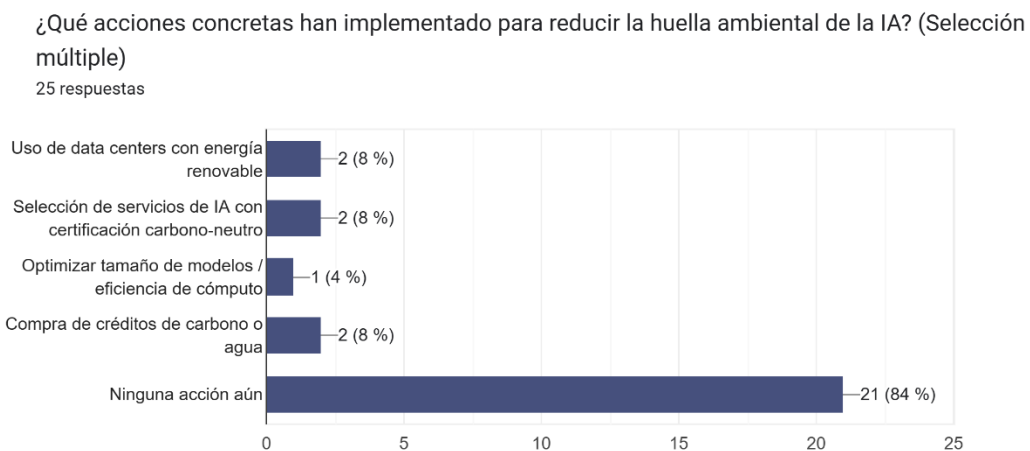


Figura 12. ¿Qué acciones concretas han implementado para reducir la huella ambiental de la IA?

El 84% no ha tomado aún ninguna acción concreta para mitigar el impacto ambiental de su uso de IA. Solo un 16% en total refiere alguna medida aislada, entre las cuales las más mencionadas (cada una por 8% de encuestados) fueron: usar data centers alimentados con energía renovable, seleccionar proveedores de IA con certificación carbono-neutral, y adquirir créditos de carbono/agua para compensar emisiones. Apenas un caso (4%) mencionó optimizar el tamaño de los modelos o la eficiencia computacional como acción.

Este hallazgo implica que la sostenibilidad de la IA está en “pañales” dentro de las fábricas de software: pocas han pasado del discurso a la práctica ambiental. El grueso de las organizaciones no ha modificado sus operaciones ni inversiones para reducir la huella, probablemente porque, ni lo miden ni se los exigen. Aquellas pocas acciones reportadas parecen más iniciativas piloto o de cumplimiento básico (usar energía renovable si el proveedor la ofrece) que un plan integral. El entrevistado señaló en este sentido que “no tenemos políticas de eficiencia de modelos aún”. Para el modelo de gestión propuesto, este resultado sugiere incorporar un capítulo de “IA sustentable” (opcional para las empresas que lo quieran adoptar), con buenas prácticas como las mencionadas. De lo contrario, las fábricas de software podrían encontrarse rezagadas cuando la sostenibilidad pase de ser un factor deseable a un requisito obligatorio en licitaciones o auditorías de clientes. El hecho de que un pequeño grupo sí actúa indica que ya hay referencia de medidas viables que se pueden compartir y estandarizar en la industria.

5.3) Al contratar proveedores de servicios de IA, ¿evalúan criterios de sostenibilidad?

Al contratar proveedores de servicios de IA, ¿evalúan criterios de sostenibilidad? (Selección única)
25 respuestas

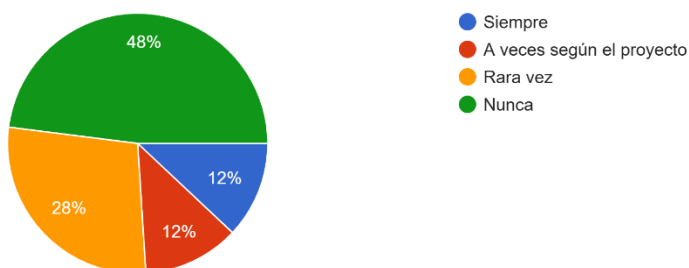


Figura 13. Al contratar proveedores de servicios de IA, ¿evalúan criterios de sostenibilidad?

El 48% “nunca” evalúa criterios de sostenibilidad en sus proveedores de IA, y otro 28% solo “rara vez” lo hace. Un 12% respondió que “siempre” considera esos criterios, mientras el 12% restante lo hace ocasionalmente (“a veces, según el proyecto”). En otras palabras, cerca de 3 de cada 4 organizaciones no incluyen sistemáticamente la huella ecológica dentro de los factores de selección de tecnologías IA.

El hallazgo en esta práctica (o falta de) concuerda con los datos anteriores: si internamente no están midiendo ni actuando, difícilmente exigirán a terceros algo distinto. El nivel de conciencia ecológica en la cadena de suministro de IA es bajo actualmente. La implicación es que las fábricas de software podrían estar indirectamente fomentando el uso de infraestructuras o modelos poco eficientes energéticamente al no evaluarlos; por ejemplo, escogiendo un proveedor únicamente por precisión del modelo sin preguntar por el consumo energético de sus data centers. Aquí hay un hallazgo de complacencia ambiental: refleja que la sostenibilidad no ha permeado en los criterios de gobernanza o compras de tecnología de la mayoría. El entrevistado admitió que en licitaciones o selecciones “pesan más la seguridad y el costo; honestamente, la huella de carbono del proveedor no es algo que preguntemos, a menos que el cliente lo ponga sobre la mesa”.

5.4) ¿Algún cliente les ha solicitado que las soluciones de IA sean carbono-neutras?

¿Algún cliente les ha solicitado que las soluciones de IA sean carbono-neutras? (Selección única)
25 respuestas

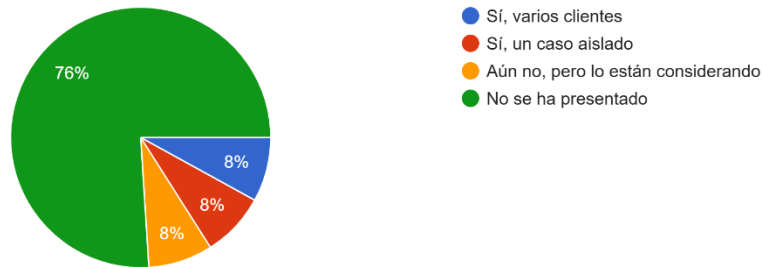


Figura 14. ¿Algún cliente les ha solicitado que las soluciones de IA sean carbono-neutras?

Un 76% de los encuestados afirmó que ningún cliente les ha solicitado hasta ahora que las soluciones de IA que desarrollan sean carbono-neutrales. Solo en un 8% de los casos varios clientes han hecho este tipo de requerimientos, y otro 8% lo vio en un caso aislado. El restante 8% indicó que “aún no, pero los clientes lo están considerando”, lo cual sugiere que empiezan a surgir conversaciones al respecto, aunque sin formalizarse exigencias.

Este hallazgo muestra que el mercado latinoamericano (donde se ubica la mayoría de estos participantes) aún no ha empujado fuertemente la agenda verde en proyectos de IA, a diferencia de lo que se observa en regiones pioneras donde ya hay cláusulas de sostenibilidad en contratos tecnológicos.

La implicación es que muchas fábricas de software están reaccionando a las demandas del cliente, y si estas demandas no incluyen aún neutralidad de carbono, el incentivo para proactividad interna es bajo. Sin embargo, el hecho de que 1 de cada 6 mencione haberlo visto (o preverlo pronto) indica que la tendencia puede estar cambiando gradualmente, por ejemplo, corporaciones multinacionales o ciertos sectores podrían empezar a exigir criterios verdes a sus proveedores de software.

El entrevistado confirmó no haber recibido solicitudes ambientales explícitas de sus clientes por el momento, aunque reflexionó que “en licitaciones de empresas grandes es posible que pronto pidan algo de esto; tenemos que anticiparnos”. Para el modelo de gestión, este resultado sugiere incluir la sostenibilidad como propuesta de valor diferenciadora pero opcional: es decir, aunque hoy no sea un requisito extendido, una fábrica de software que se adelante a ofrecer IA sostenible podría ganar ventaja competitiva o al menos estar preparada cuando se vuelva mandatorio. La ausencia de presión del cliente no debería llevar a la inacción, sino aprovecharse como tiempo para prepararse sin la premura de una imposición externa.

5.5) Frente a esas solicitudes (o futuras), ¿Cómo respondería su fábrica de software?

Frente a esas solicitudes (o futuras), ¿Cómo respondería su fábrica de software? (Selección múltiple)

25 respuestas

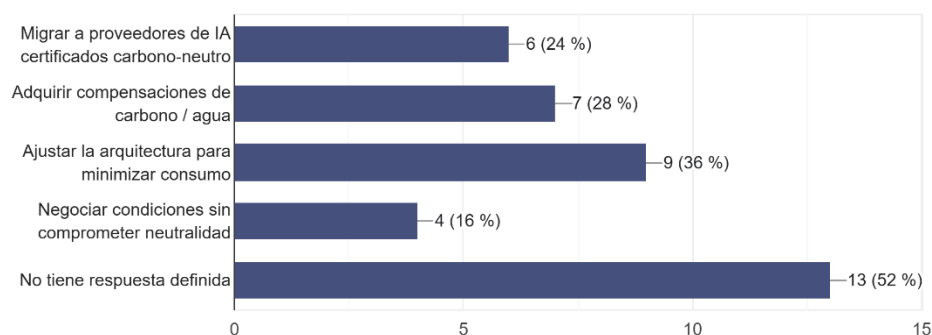


Figura 15. Frente a esas solicitudes (o futuras), ¿Cómo respondería su fábrica de software?

El 52% de los encuestados no tiene aún una respuesta definida para afrontar tal solicitud. Entre quienes sí contemplan acciones, la opción más escogida sería “ajustar la arquitectura para minimizar el consumo energético”

de la solución (36% la mencionan). Otras posibles respuestas incluyen adquirir compensaciones de carbono/agua (28%) para neutralizar las emisiones que no se puedan eliminar, y migrar hacia proveedores de IA con certificaciones carbono-neutral (24%) para asegurarse de que la infraestructura subyacente sea más limpia. Un 16% indicó que negociaría condiciones con el cliente sin comprometer la neutralidad (quizá ofreciendo alternativas o explicitando los costos y limitaciones de lograrla).

Este conjunto de hallazgos implica que, ante un eventual requerimiento ambiental, las fábricas de software mayoritariamente aún improvisarían su respuesta, recurriendo probablemente a soluciones técnicas y compensatorias relativamente inmediatas. Es positivo notar que la respuesta más común sería mejorar la eficiencia arquitectónica, pues aborda el problema de raíz reduciendo consumo (a diferencia de solo comprar créditos de carbono). Sin embargo, la alta tasa de “no tener respuesta” refleja preparación insuficiente: las empresas no han incorporado protocolos o estrategias ambientales en sus planes de contingencia. El entrevistado reconoció esto con franqueza: “Si mañana nos exigen carbono-neutral, tendríamos que correr a averiguar cómo; quizá optimizar código y pagar compensadores. No tenemos un plan formal”. Para la investigación, esto subraya la necesidad de incluir lineamientos de sostenibilidad en el modelo de gestión desde el principio, para que las fábricas de software desarrollen la capacidad de responder a estas exigencias sin improvisación. Incorporar desde ya consideraciones de arquitectura sostenible, proveedores verdes y mecanismos de compensación en la planificación de la IA garantizará que, cuando llegue la demanda, la respuesta no solo exista, sino que esté integrada como parte natural del proyecto.

5.6) ¿Qué importancia estratégica le otorga su organización a la sostenibilidad ambiental de la IA?

¿Qué importancia estratégica le otorga su organización a la sostenibilidad ambiental de la IA?
(Selección única)
25 respuestas

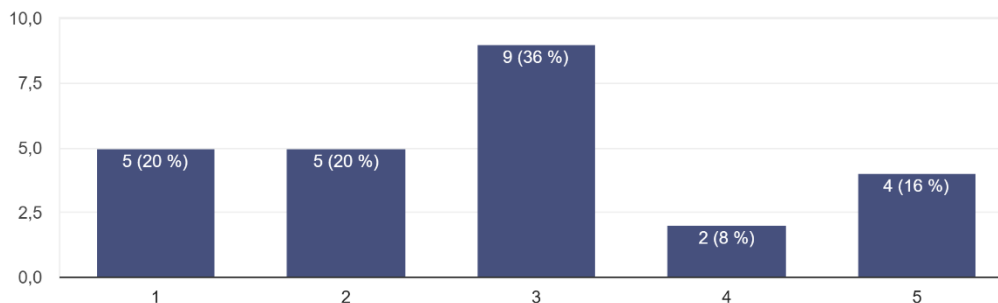


Figura 16. ¿Qué importancia estratégica le otorga su organización a la sostenibilidad ambiental de la IA?

Únicamente el 16% lo visualiza como una importancia alta, la mayoría la ubica en un plano medio o bajo: un 36% le da una importancia moderada (nivel 3), y un 40% la considera de baja o muy baja prioridad (niveles 2 o 1).

Este hallazgo pone en contexto todos los anteriores: ayuda a explicar por qué no se mide ni actúa mayoritariamente simplemente, aún no se le ha dado suficiente peso en la estrategia organizacional. La implicación es que, mientras la sostenibilidad de la IA no escale en prioridad, es poco probable que las fábricas de software dediquen recursos o esfuerzos serios a abordar su huella ecológica. No obstante, también ofrece espacio de mejora: las percepciones pueden cambiar.

Se puede medir que estamos ante una fase temprana de conciencia, donde algunas organizaciones visionarias (el 16% que ya lo prioriza mucho) están marcando la pauta, pero la mayoría espera un disparador externo para actuar. Para el modelo de gestión propuesto, este resultado sugiere incluir la sostenibilidad no solo como apartado técnico, sino también promoverla a nivel

de visión y cultura corporativa, de forma que pase de ser un tema periférico por integrarse en la definición de éxito de la adopción de IA. Estrategias como evangelizar internamente sobre "IA responsable", demostrar casos donde la eficiencia ambiental coincide con eficiencias de costo, o alinearse con objetivos de RSE (Responsabilidad Social Empresarial) de la compañía, pueden ayudar a elevar la importancia estratégica de la IA sostenible.

D) Alfa de Cronbach

1) Análisis

Con el fin de evaluar la consistencia interna de las preguntas de selección única que exploran actitudes y prácticas organizativas sobre la adopción de IA, se aplicó el coeficiente Alfa de Cronbach a los ocho ítems ordinales que comparten una escala homogénea.

2) Fórmula del Alfa de Cronbach

El coeficiente Alfa de Cronbach se calcula mediante la siguiente fórmula:

$$\alpha = \frac{K}{K-1} \left(\frac{\sum_{i=1}^K \sigma_{Y_i}^2}{\sigma_X^2} \right)$$

Figura 17. Fórmula Alfa de Cronbach

	Items								
Encuestados	3	7	8	11	12	14	15	17	Suma
E1	5	1	5	5	5	5	5	5	36
E2	5	4	5	3	1	2	1	2	23
E3	4	5	3	5	1	1	1	3	23
E4	4	5	5	5	5	5	5	5	39
E5	5	3	3	3	1	1	1	5	22
E6	5	5	4	5	1	4	4	3	31
E7	5	4	5	4	3	4	3	3	31
E8	5	5	3	5	1	1	1	3	24
E9	5	5	3	4	1	2	1	2	23
E10	4	5	1	3	1	1	1	1	17
E11	5	3	4	4	1	1	1	1	20
E12	4	4	4	3	5	5	4	4	33
E13	5	4	3	4	1	2	1	2	22
E14	5	3	3	5	1	1	1	3	22
E15	5	4	3	5	1	2	1	3	24
E16	5	4	3	3	1	1	1	3	21
E17	5	5	1	5	1	1	1	1	20
E18	5	2	3	4	1	1	1	2	19
E19	5	4	1	4	1	1	1	1	18
E20	5	5	4	5	5	2	1	1	28
E21	5	2	3	3	1	1	1	2	18
E22	4	5	1	5	1	2	1	3	22
E23	5	5	4	5	3	4	3	5	34
E24	5	4	3	5	1	1	1	3	23
E25	5	5	3	5	1	2	1	4	26
Varianza	0.16	1.2384	1.44	0.6816	2.24	2.0256	1.8016	1.68	
Sumatoria de Varianzas	11.2672								
Varianza de la suma de los ítems	34.4224								

Figura 18. Tabulación de respuestas

3) Resultados

α	Coeficiente de confiabilidad del cuestionario	0.768775
K	Número de ítems del instrumento	8
$\sum_{i=1}^K \sigma_{Y_i}^2$	Sumatoria de las varianzas de los ítems	11.2672
σ_X^2	Varianza total del instrumento	34.4224

Figura 19. Resultados

3) Interpretación

Rango α	Interpretación
≥ 0.90	Excelente (muy alta fiabilidad)
0.80 – 0.89	Buena (alta fiabilidad)
0.70 – 0.79	Aceptable (fiabilidad adecuada)
0.60 – 0.69	Cuestionable (revisar ítems)
0.50 – 0.59	Pobre (escala poco consistente)
< 0.50	Inaceptable (descartar o rediseñar escala)

Figura 20. Interpretación

El valor 0,768 indica una consistencia interna aceptable; los ocho ítems correlacionan lo suficiente como para integrarse en un índice global de “madurez/actitud organizacional hacia la IA”.

E) Principales Hallazgos de la Encuestas

1) Prácticas actuales de IA

En la mayoría de las fábricas de software encuestadas, la inteligencia artificial ya forma parte del día a día. Herramientas como ChatGPT, Gemini y GitHub Copilot son ampliamente utilizadas, sobre todo en la etapa de programación y análisis de requisitos. Su uso es frecuente: ocho de cada diez profesionales la emplean a diario. Las fases de pruebas y mantenimiento también comienzan a beneficiarse, aunque en despliegue y operaciones el uso todavía es limitado.

2) Barreras y facilitadores

Las principales dificultades para adoptar IA no están tanto en el presupuesto, sino en las personas y en la organización: falta de habilidades especializadas, preocupaciones por seguridad y privacidad de datos, ausencia de estrategias claras y cierta resistencia cultural. Por otro lado, una cultura interna de innovación, el respaldo de la dirección y la disponibilidad de infraestructura han impulsado su integración. Sin embargo, casi la mitad de las organizaciones no cuenta con políticas claras sobre uso ético y responsable, lo que supone un riesgo y una oportunidad de mejora.

3) Impacto y métricas

La percepción general es positiva: la mayoría reporta mejoras en productividad, reducción de errores y tiempos de entrega. También se han observado avances en métricas operativas como la frecuencia de despliegues y el tiempo de recuperación ante incidentes, aunque muchas empresas aún no miden de forma sistemática estos indicadores. Esto limita la posibilidad de evaluar con precisión el verdadero impacto de la IA.

4) IA sostenible / Impacto ambiental

La sostenibilidad todavía es un tema pendiente. La mayoría no mide el consumo energético ni la huella de carbono asociada al uso de IA, y son pocas las que han implementado acciones para reducirla. Tampoco es habitual que clientes o proveedores exijan criterios de sostenibilidad. La prioridad

estratégica que se da al tema es baja, por lo que su integración en las decisiones de adopción tecnológica aún es incipiente.

CAPÍTULO V. Solución del Problema

A) Propuesta de solución

Mapa de Procesos (nivel 0)

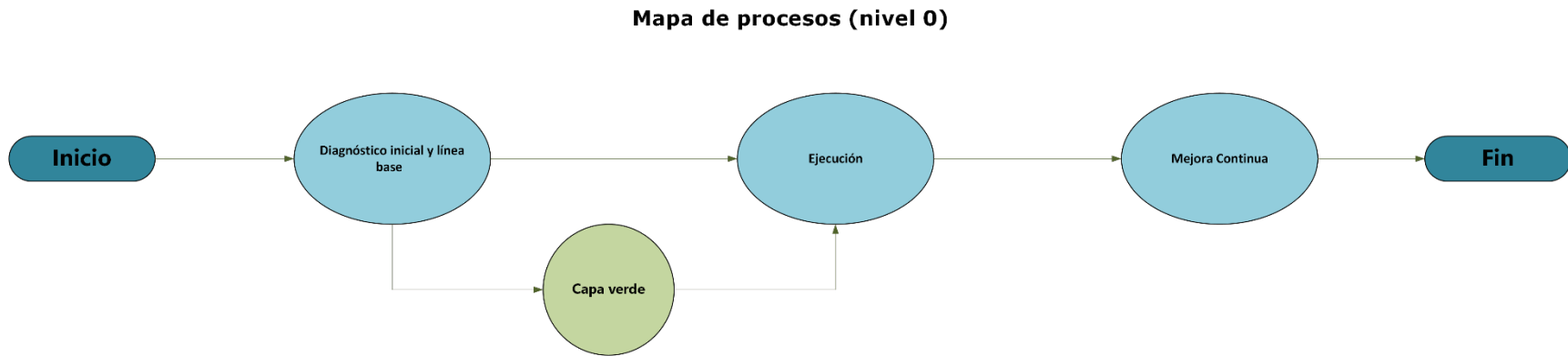


Figura 21. Mapa de Procesos (nivel 0)

B) Desarrollo de la solución

Modelo de gestión para la adopción estratégica de IA en fábricas de software

Las encuestas y el análisis de literatura muestran que la IA ya está en la jornada laboral de la mayoría: herramientas como ChatGPT, Gemini o Copilot se usan sobre todo en programación y análisis de requisitos; pruebas y mantenimiento empiezan a beneficiarse y el tramo de despliegue/operaciones aún tiene margen de mejora. Esto explica el énfasis en “quick wins” iniciales en desarrollo y testing: es donde las barreras de entrada son más bajas y el retorno, más nítido. Al mismo tiempo, casi la mitad de las organizaciones no tiene políticas claras de uso responsable, un hueco que eleva riesgos y que la solución aborda desde la fase de planificación con lineamientos éticos, privacidad y propiedad intelectual.

En percepción de impacto, la mayoría reporta más productividad, menos errores y entregas más rápidas; también hay mejoras en métricas operativas como frecuencia de despliegues y tiempo de recuperación, aunque persiste un déficit: muchas empresas no miden de forma sistemática. Por eso, el modelo ancla la validación en objetivos cuantificados y en el seguimiento de DORA (Investigación y evaluación de DevOps) y métricas de negocio, con la instrucción explícita de revisar si las metas se cumplen y ajustar cuando no. Esta disciplina evita la “niebla del entusiasmo” y deja evidencia del retorno de la inversión.

Otro hallazgo clave es cultural: la adopción sostenida depende de combinar formación continua, liderazgo visible y reglas claras. La literatura revisada, apoyada en UTAUT (Teoría Unificada de Aceptación y Uso de la

Tecnología), sugiere que la aceptación crece cuando se trabaja la confianza algorítmica, se explicitan beneficios y se reducen ansiedades con acompañamiento y práctica guiada. Por eso el diagnóstico inicial releva percepciones y condiciones facilitadoras, y la evaluación final vuelve a medir para confirmar que la IA ya no es “invitada” sino parte natural del flujo de trabajo.

Transición: ¿hacia dónde evoluciona la solución?

La solución planteada traza un recorrido claro que inicia con un diagnóstico procesos, cultura y habilidades, sigue con una planificación estratégica que ordena prioridades, valida con pilotos controlados y, cuando algo funciona, lo integra y escala hasta convertirlo en la forma normal de trabajar. En esa trayectoria, la sostenibilidad no es un adorno: la “capa verde” puede activarse desde temprano para que los criterios ambientales acompañen todo el camino y no se queden para “más adelante”. El resultado buscado es pasar de “experimentar con IA” a operar con IA integrada en el día a día, con métricas que confirmen valor y con la flexibilidad suficiente para adaptarse a nuevas generaciones de herramientas sin rehacerlo todo desde cero.

Esta propuesta se reconoce porque “la IA tiene propósito, métricas y dueño”: cada iniciativa nace con un para qué de negocio, una métrica que la valida y un responsable que la impulsa. No se promete magia: se pasa “del piloto al valor en producción” con criterios de éxito claros y una ruta de integración definida antes de escribir la primera línea del prototipo. La consigna cultural es “gobernanza primero, herramienta después”: políticas de uso, seguridad y ética dan el marco para que la adopción sea sostenible, no sólo entusiasta. La brújula operativa es DORA y los indicadores de negocio: si

no mejoran la frecuencia de despliegue, el lead time (plazo de entrega), la calidad o la satisfacción del cliente, se corrige o se detiene. Y, si la organización decide activarla, la “capa verde” acompaña como criterio transversal, integrándose en la planificación y en los pilotos con objetivos ambientales concretos y un checklist de proveedores y elegir conscientemente dónde y cómo correr la IA.

La sostenibilidad aparece como tarea pendiente y oportunidad estratégica. La mayoría no mide consumo energético ni huella de carbono de la IA y pocos clientes/proveedores lo exigen hoy; integrar la “capa verde” desde la planificación, con objetivos ambientales, responsables designados y checklist de selección de proveedores, prepara a la organización antes de que el mercado o la regulación lo impongan. En términos prácticos, significa elegir regiones y servicios con mejor perfil ambiental, medir energía en entrenamientos o cargas intensivas y, cuando corresponda, compensar. Así, la adopción aporta valor de negocio y credenciales ambientales que diferencian.

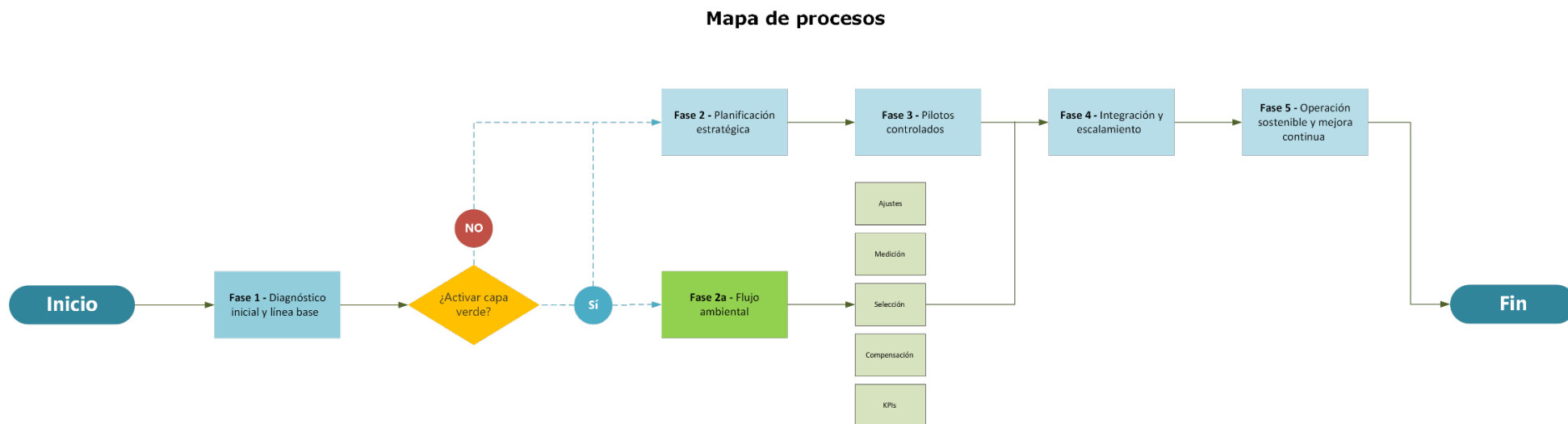


Figura 22. Diagrama del modelo de adopción estratégica de IA en fábricas de software

Diagrama del modelo de adopción estratégica de IA en fábricas de software abarca desde el diagnóstico inicial hasta la operación sostenible (incluye la decisión opcional de activar la "capa verde" de sostenibilidad).

La figura ilustra las fases secuenciales del modelo propuesto. Primeramente, con un Diagnóstico inicial y línea base, seguido por la Planificación estratégica de la adopción. Luego se ejecutan Pilotos controlados para validar en pequeño alcance, tras lo cual viene la Integración y escalamiento de las soluciones IA exitosas al resto de la organización. Finalmente, la Operación sostenible y mejora continua consolida la adopción a largo plazo. Adicionalmente, se observa una dimensión de sostenibilidad ambiental (capa verde) que puede activarse opcionalmente tras el diagnóstico para incorporar criterios ecológicos en todo el proceso.

Fase 1: Diagnóstico inicial y línea base

1) Actividades

1.1) Evaluación de la preparación organizacional

Realizar un diagnóstico del estado actual de la fábrica de software en cuanto a procesos, cultura y habilidades para adoptar IA. Aplicar el marco de referencia UTAUT-IA (una adaptación del modelo de aceptación tecnológica UTAUT para IA) que provee dimensiones medibles para evaluar la preparación organizacional. Esto permite sondear aspectos como expectativa de desempeño con IA, esfuerzo percibido, apoyo social y condiciones facilitadoras en la empresa.

1.2) Identificación de usos actuales de IA y oportunidades

Investigar en qué áreas del ciclo de vida de desarrollo ya se está usando IA (aunque sea de forma informal) y dónde podría aportar valor. Documentar qué herramientas de IA utilizan los equipos (ChatGPT, Copilot, etc.) y con qué frecuencia proporciona una línea base tecnológica y cultural importante.

1.3) Detección de barreras y facilitadores existentes

Mediante entrevistas y cuestionarios internos, relevar los principales obstáculos percibidos para adoptar IA y los factores a favor. Comúnmente, las barreras incluyen falta de habilidades especializadas en IA, preocupaciones por seguridad y privacidad de datos, ausencia de una estrategia clara y resistencia cultural al cambio. Por otro lado, los facilitadores pueden ser una cultura interna de innovación, apoyo de la alta dirección, infraestructura disponible o experiencias previas positivas. Identificar estos puntos desde el diagnóstico permite enfocar la estrategia en cerrar brechas (por ejemplo, plan de capacitación) y apalancar fortalezas (por ejemplo, patrocinio ejecutivo) en fases posteriores.

2) Herramientas específicas recomendadas

2.1) Evaluación de métricas DORA

Aprovechar plataformas de repositorios y gestión de código que permitan evaluar el rendimiento del equipo en las cuatro métricas DORA y brindar recomendaciones. Alternativamente, se pueden usar características nativas de la plataforma de desarrollo actual; por ejemplo, GitLab incluye de

forma nativa el tracking de métricas DORA en sus pipelines y dashboards. Otras opciones como CircleCI (que genera reportes de eficiencia de pipeline) también pueden servir. Lo importante es establecer un mecanismo para capturar datos reales de despliegues, tiempos de ciclo y fallos, ya sea mediante estas herramientas o scripts personalizados, para consolidar la línea base numérica.

2.2) Encuestas y entrevistas estructuradas (UTAUT-IA)

Utilizar el cuestionario del modelo de adopción como sugerencias para estar alineado con los constructos de UTAUT adaptado a IA, cubriendo expectativas, actitud hacia la tecnología, intención de uso, etc. Donde se aplican escalas Likert para cuantificar la percepción de la organización. Adicionalmente, realizar entrevistas a profundidad con líderes (CTO, gerentes de desarrollo, líderes de equipo) para obtener contexto cualitativo. Se sugiere involucrar a tomadores de decisión ofrece perspectivas estratégicas e informadas, por lo que es útil en esta fase recabar su visión.

2.3) Herramientas de monitoreo e infraestructura DevOps

Desde el inicio, se recomienda habilitar sistemas de monitoreo continuo como Grafana o Kibana vinculados a CI/CD actual. Se sugiere configurar dashboards que muestren las métricas base obtenidas (tiempos de build, frecuencia de despliegues, tasas de error) facilitará ver tendencias una vez que se integre IA. También preparar la infraestructura de control de versiones (Git) y la plataforma de CI para registrar información adicional de la IA en el futuro (por ejemplo, etiquetas en commits que indiquen uso de una herramienta IA, para seguimiento). De ser posible, habilitar un entorno de

sandbox seguro donde se puedan probar herramientas de IA durante el diagnóstico sin afectar los proyectos críticos.

3) Recomendaciones prácticas y criterios de validación

3.1) Participación multidisciplinaria

Involucre a diferentes perfiles en el diagnóstico (desarrolladores, ingenieros DevOps, QA, gerentes de proyecto, etc.). Esto además de enriquecer la recolección de datos, también genera aceptación y compromiso desde temprano. Una práctica útil es realizar sesiones de trabajo (workshops) donde equipos evalúan conjuntamente su nivel de madurez. El resultado es más aceptado cuando proviene de una autoevaluación colaborativa.

3.2) Criterios de salida del diagnóstico

Al culminar esta fase, debería producirse un informe de diagnóstico documentado. En él se validan hallazgos como: (a) Índice de Preparación UTAUT-IA, (b) métricas DevOps base (por ejemplo, despliegues mensuales = X, lead time promedio = Y días, MTTR = Z horas, etc.), (c) listado de casos de uso actuales de IA en la empresa y brechas identificadas, y (d) principales obstáculos y facilitadores internos. Un criterio de éxito es que la dirección apruebe este diagnóstico y lo reconozca como reflejo fiel de la realidad actual; eso sentará las bases para la planificación.

3.3) Enfoque en quick wins

Priorizar la detección de áreas donde la IA podría dar resultados rápidos y visibles. La evidencia muestra que las fábricas de software suelen empezar por las fases creativas (codificación, diseño) antes de pasar a operaciones. Siguiendo esto, el diagnóstico debe resaltar oportunidades de “victorias tempranas” (por ejemplo, uso de asistentes generativos en desarrollo, automatización de pruebas unitarias con IA) que puedan pilotarse pronto. A la vez, identifique riesgos críticos que deben atenderse primero, por ejemplo, si hay resistencia cultural, conviene planificar una estrategia de cambio antes de forzar cualquier adopción tecnológica.

Fase 2: Planificación estratégica de la adopción de IA

1) Actividades clave

1.1) Definición de visión y objetivos estratégicos

Con el diagnóstico en mano, la alta dirección y el equipo de proyecto establecen una visión clara de hacia dónde se quiere llevar a la organización con IA. Esto implica articular objetivos SMART alineados al negocio: por ejemplo, “Aumentar la productividad de desarrollo en un 20% en los próximos 12 meses”, “Reducir el tiempo de entrega al cliente en 30% mediante automatizaciones inteligentes” o “Mejorar la calidad (disminuir defectos postproducción en 40%)”. También se definen indicadores clave para monitorear estos objetivos. Muchos de ellos derivarán de las métricas base DORA y métricas de negocio identificadas (tiempo, costo, calidad). Es importante incluir no solo metas técnicas, sino también de valor agregado:

por ejemplo, mejorar la satisfacción del cliente o habilitar nuevos servicios gracias a IA.

1.2) Priorización de casos de uso y roadmap por fases

Dado que no todo se puede abordar a la vez, se elabora un plan por etapas. Una buena práctica es comenzar con áreas de impacto rápido (“quick wins”) identificadas en el diagnóstico, para luego escalar gradualmente. Por ejemplo, el roadmap podría establecer: Fase piloto enfocada en asistencia a desarrolladores (alto beneficio inmediato), siguiente fase para automatizar testing, luego IA en monitoreo de operaciones. Cada etapa con un alcance definido. Se calendariza un piloto inicial (Fase 3) con tiempos acotados, y se prevé cómo se integrará si es exitoso (Fase 4). También se contemplan hitos intermedios y decisiones de si se avanza o no tras cada piloto.

1.3) Gobernanza y estructura del proyecto

En esta fase se establecen los mecanismos de gestión y control para la iniciativa de IA. Se asignan roles y responsabilidades claros: por ejemplo, definir un responsable de IA (un champion o líder de proyecto de IA) encargado de coordinar las fases; identificar dueños para cada piloto (quizá líderes técnicos de cada área involucrada); incluir representantes de Seguridad, DevOps, y otras áreas clave en un comité de seguimiento. Adicionalmente, se formalizan políticas y lineamientos: crear o actualizar políticas de empresa sobre uso de IA (lineamientos éticos, cumplimiento legal, políticas de privacidad de datos al usar herramientas IA, propiedad intelectual del código generado, etc.). La ética de la IA debe incorporarse desde la planificación y diseño, no como una idea tardía. Por tanto, se debe establecer

un comité de ética o al menos criterios éticos que todo proyecto con IA deberá respetar. Esta gobernanza garantizará que la adopción siga siendo responsable y alineada con valores corporativos.

1.4) Gestión de talento

Considerando que las personas son un factor crítico, la planificación estratégica debe incluir acciones concretas de capacitación. Esto podría tomar la forma de un programa de formación escalonado: inicia con talleres introductorios sobre IA para todo el personal (por ejemplo, "IA 101", conceptos básicos), luego capacitaciones específicas para los equipos piloto en las herramientas que usarán, y más adelante formación avanzada o certificaciones externas para roles especializados (científicos de datos, ingenieros de ML, etc., si se necesitarán).

1.5) Selección de herramientas y arquitectura tecnológica

Con objetivos claros, se evalúan qué herramientas de IA específicas y qué infraestructura se requerirá. En la planificación se investigan las opciones disponibles: por ejemplo, si el objetivo es asistencia en codificación, decidir entre GitHub Copilot, Tabnine u otro asistente, comparando costos y compatibilidad; si es IA para pruebas, ver herramientas de generación de tests unitarios. También definir la arquitectura de integración: cómo encajarán esas herramientas en el pipeline DevOps existente. ¿Requerirán acceso a la nube? ¿Afectan el proceso de CI/CD? ¿Se necesitan nuevos entornos (por ejemplo, un servidor para entrenar modelos)? Es importante en este punto planificar la inversión: elaborar un business case donde se estime el costo de licencias, hardware, horas de implementación, versus el beneficio esperado (puede

incluir el costo de no hacer nada). Importante iniciar con pilotos de bajo costo (usando versiones gratuitas), pero en planificación hay que prever el escalamiento: presupuestar la eventual compra de suscripciones o incremento de infraestructura si los pilotos prosperan.

1.6) Consideración de la sostenibilidad (capa verde)

Si la organización decide incorporar la dimensión ambiental, este es el momento de integrarla en la estrategia (más detalle en la sección de sostenibilidad al final). Básicamente, en la planificación se incluye también un objetivo de IA sostenible, con compromisos medibles (por ejemplo, “utilizar proveedores carbono neutrales”). Se determinan responsables para velar por esto, y se planifica la medición (implementando mediciones de energía en los pilotos). Incluir esta capa desde la visión estratégica asegura que la sostenibilidad sea parte de los criterios de éxito global y no un agregado posterior.

2) Herramientas específicas recomendadas

2.1) Frameworks y plantillas de planificación estratégica

Se solicita aplicar marcos adaptados al contexto de IA. Por ejemplo, aplicar un Balanced Scorecard tecnológico, donde se agreguen objetivos de IA en las perspectivas de proceso interno y aprendizaje. O utilizar OKRs (Objectives and Key Results): definir un objetivo (“Implementar IA en el ciclo DevOps aumentando eficiencia”) con KR mensurables (“KR1: mejorar

frecuencia de despliegue de 1 por semana a 3 por semana para Q4”, “KR2: reducir retrabajo en un 15% en 6 meses”, etc.). Estas herramientas obligan a aterrizar la estrategia en indicadores concretos y pueden difundirse a todos los niveles para alinear esfuerzos.

2.2) Matriz de riesgos y plan de mitigación

Apoyarse en herramientas de gestión de riesgos (PMI o CMMI). Utilizar la matriz de riesgos específica de la adopción de IA: donde se pueden enlistar riesgos (por ejemplo, “posible fuga de datos sensibles a herramientas de IA externas”, “falta de adopción por parte del equipo”, “sobrecostos en infraestructura cloud”, “impacto ambiental negativo”), estimar probabilidad e impacto, y planificar respuestas. Por ejemplo, para el riesgo de fuga de datos: acción de mitigación = políticas estrictas + sandbox controlado; para falta de adopción: acción = programa intensivo de formación y comunicación. Esta matriz debe ser revisada y aprobada en la planificación, de modo que todos estén conscientes de los desafíos por venir y cómo se abordarán.

2.3) Herramientas de comunicación y colaboración

Es clave gestionar bien la comunicación durante la transformación. Se recomienda habilitar un espacio central (portal o canal) donde se publique toda la información relevante: la visión, las etapas, los roles, FAQs, documentos de política, etc. Confluence, Notion, Teams o SharePoint pueden servir para crear esta “wiki” del proyecto de IA. Asimismo, definir canales de comunicación regulares: boletines internos sobre avances, un canal de Teams/Slack dedicado a IA para dudas y compartir logros, reuniones periódicas con liderazgo para reforzar el compromiso. Estas herramientas

blandas aseguran que la estrategia esté visible y entendida por todos, minimizando rumores o incertidumbre.

2.4) Plataformas DevOps alineadas con la estrategia

Verificar que las herramientas de ALM (Application Lifecycle Management) actuales puedan adaptarse al plan. Por ejemplo, utilizar Jira o Azure Boards para crear épicas y tareas relacionadas con la adopción de IA (como “Desarrollar piloto X”, “Actualizar política Y”), integrando el seguimiento en el backlog general de la organización. Esto visibiliza que el trabajo de adopción es parte de los proyectos y no algo aislado. También planificar cómo incorporaremos el seguimiento de métricas: configurar el tablero de métricas (en Grafana u otro) para que en la fase de operación muestre los KRs definidos. Preparar esas herramientas ahora ahorra tiempo luego.

3) Recomendaciones prácticas y criterios de validación

3.1) Compromiso ejecutivo y patrocinio

Asegúrese de que la estrategia de adopción de IA esté respaldada al más alto nivel. Vale recalcar que: sin un champion ejecutivo, las iniciativas suelen diluirse. Un indicador de éxito de esta fase es tener una declaración o documento de estrategia de IA aprobado por la dirección (por ejemplo. el CTO) y difundido internamente, de modo que todos sepan que “va en serio”. Si es posible, incluir la adopción de IA en la planificación estratégica corporativa anual o como parte del plan de innovación de la empresa.

3.2) Realismo y foco

Evitar que la planificación se vuelva demasiado ambiciosa o difusa. Es preferible unos pocos objetivos bien definidos que intentar cubrir todas las posibilidades de IA de golpe. La tentación de prometer revolución total con IA puede ser alta, pero una estrategia creíble es incremental. Un criterio de validación es que el plan pueda explicarse en una página: fases, objetivos por fase, resultados esperados. Si es muy complejo de comunicar, tal vez haya que simplificarlo.

3.3) Criterios de éxito intermedios

Establecer hitos medibles para evaluar la ejecución de la estrategia. Por ejemplo, “Al finalizar el Q2, tener 2 pilotos finalizados con mejora demostrada en X”, “Para el Q3, 50% de desarrolladores formados en herramienta Y”. Estos checkpoints servirán en gestión de proyecto para ver si se está en camino. La estrategia debe incluir mecanismos de revisión: quizás reuniones trimestrales de comité para ajustar el rumbo según resultados parciales.

3.4) Alineación con políticas y regulaciones

En esta fase es de importancia validar que la estrategia no contradiga normativas internas o externas. Por ejemplo, si la empresa tiene políticas estrictas de seguridad, asegurar que cualquier herramienta IA a usar cumpla con ellas (¿está aprobada por seguridad? ¿dónde residen los datos?). El cumplimiento también aplica a estándares de calidad: si la empresa está certificada en ISO 9001 o sigue CMMI, asegurarse de que la adopción de IA

no romperá procesos, sino que se documentará y controlará según esos marcos.

3.5) Obtención de “quick wins” tempranos

Aunque los pilotos ocurran en la siguiente fase, la estrategia debería contemplar ganar impulso rápido. Se puede planificar, por ejemplo, un MVP (Producto Mínimo Viable) de IA en un área chica que en pocos meses muestre valor. Tener una pequeña victoria en las primeras etapas ayuda a validar la estrategia y mantener el compromiso. El criterio de validación final de esta fase es la aprobación del plan y la disposición de los recursos necesarios (presupuesto, personal asignado, herramientas preparadas) para arrancar los pilotos. Sin recursos comprometidos, el plan se quedaría en papel.

Fase 3: Pilotos controlados

1) Actividades clave

1.1) Implementación de proyectos piloto en entornos controlados

En esta fase, se pasa del papel a la acción con pequeños proyectos piloto. Cada piloto debe tener un alcance bien definido y acotado para probar una hipótesis de valor de IA. Por ejemplo, un piloto podría ser “Usar un asistente de código (IA) en un módulo crítico para ver si acelera el desarrollo sin pérdida de calidad”, u otro: “Aplicar una herramienta de IA para generar casos de prueba unitarios en uno de nuestros servicios y medir cobertura y defectos descubiertos”. Es importante aislar estos pilotos en entornos controlados: preferiblemente en proyectos no críticos o en ramas separadas

del código, para no poner en riesgo la operación normal. Se establecen equipos pequeños dedicados a cada piloto (20% del equipo) con la misión de ejecutar el experimento y documentar resultados.

1.2) Desarrollo incremental y experimento práctico

Durante el piloto se adopta un enfoque iterativo ágil. No se trata de un proyecto grande, sino de probar rápidamente la herramienta o técnica de IA elegida, ajustarla y aprender. Por ejemplo, si el piloto es introducir ChatGPT para refinar requisitos, se probará en un par de historias de usuario primero, se verá cómo ayuda en claridad o detección de ambigüedades, se recopilará feedback y luego se ampliará gradualmente. Cada piloto debería durar pocas semanas (4-8 semanas típicamente) para mantener enfoque y urgencia. Durante su ejecución, se realizan reuniones frecuentes (diarias o semanales tipo Scrum) para monitorear el avance y remover obstáculos. La idea es experimentar de forma controlada pero rápida: probar la integración técnica de la IA en el proceso, y también observar la reacción del equipo a esa integración.

1.3) Monitoreo y medición de resultados

Desde el arranque de un piloto se definen KPIs específicos que determinarán éxito o fracaso del experimento. Estos deben alinearse con los objetivos definidos en la planificación. Por ejemplo, si el piloto es asistencia de código, KPIs podrían ser: "tiempo promedio para desarrollar una funcionalidad X" comparado contra histórico sin IA, "líneas de código escritas por desarrollador". Si es un piloto de pruebas, KPI: "cobertura de tests lograda (%)" o "número de bugs críticos encontrados antes de deploy". También se

pueden usar métricas DORA a nivel del equipo piloto: ¿mejoró su frecuencia de commits despliegues? ¿Se redujo su tiempo de ciclo? Estas métricas ofrecen evidencia objetiva. Junto a esto, medir la experiencia: encuestas de satisfacción del equipo piloto con la herramienta, observaciones cualitativas (¿la IA sugirió código útil o generó ruido?). Todo resultado debe registrarse. Al final de cada piloto, se comparan los KPIs con la línea base inicial: por ejemplo, “Copilot redujo el tiempo de desarrollo de un módulo en 15% respecto a sprints anteriores sin Copilot”, ese tipo de dato es oro para justificar la adopción.

1.4) Ajustes sobre la marcha y gestión del cambio local

Los pilotos también son un ejercicio de gestión del cambio a pequeña escala. Es normal que surjan imprevistos o resistencias en el equipo piloto. La recomendación es tener una actitud flexible: si la herramienta IA no funciona como se esperaba, iterar (ajustar parámetros, probar otro enfoque) dentro del piloto mismo. Asimismo, acompañar al equipo con coaching: quizá al inicio algunos desarrolladores ignoren la herramienta por hábito; el líder del piloto debe incentivarlos a usarla, mostrarles funcionalidades, etc. Se deben recopilar también las preocupaciones o mejoras sugeridas por el equipo piloto, ya que ellos serán los evangelizadores luego. Una táctica útil es establecer reuniones de retrospectiva semanales donde el equipo discuta qué va bien y mal con la IA, y con base a eso se afinan guías de uso o se decide pivotar de estrategia si algo no da resultado. El piloto es un espacio seguro para aprender y ajustar sin las consecuencias de un fallo a gran escala.

1.5) Documentación de lecciones aprendidas

Al concluir (e incluso durante) cada piloto, es fundamental documentar todo hallazgo y decisión. Se debe elaborar un informe de piloto que incluya: objetivo del piloto, participantes, duración, herramientas utilizadas, resultados cuantitativos (KPIs medidos), resultados cualitativos (feedback del equipo, anécdotas), problemas encontrados, cómo se resolvieron, y una recomendación final (continuar/ajustar/descartar la iniciativa). Este documento servirá para la siguiente fase como guía. Además, cualquier activo generado en el piloto (código fuente, scripts, configuración de herramientas, conjuntos de datos usados) debe almacenarse ordenadamente para reutilizar. Por ejemplo, si se crearon nuevos casos de prueba con IA, dejarlos en el repositorio de tests para el proyecto. La gestión del conocimiento aquí evita reinventar la rueda en cada equipo cuando se escale.

1.6) Validación y decisión

Finalmente, cada piloto termina con una evaluación formal frente a los criterios de éxito definidos. Si cumple o excede las expectativas, se clasifica como “éxito” y se recomienda su adopción más amplia (pasa a fase 4). Si el resultado es mixto o negativo, se analiza si merece una iteración más (ajustar algo y repetir piloto) o si se abandona la idea. Es válido descartar pilotos que no probaron valor, de hecho, es preferible descartar a tiempo en esta fase que insistir en algo inservible. Esta toma de decisión debe involucrar al comité definido en planificación, para que haya objetividad. Por ejemplo, si el KPI esperado era reducir 20% el tiempo de pruebas y solo se logró 5%, quizás el comité decida que no vale la pena escalar esa herramienta en particular. En cambio, si un piloto fue exitoso, debe obtener luz verde y recursos para integrarse más ampliamente.

2) Herramientas específicas recomendadas

2.1) Infraestructura de CI/CD para pruebas controladas

Aprovechar la infraestructura existente, pero aislando los pilotos. Por ejemplo, si se cuenta con Jenkins o GitLab CI, crear pipelines específicos para el piloto. Un caso: en un piloto de testing automático con IA, podríamos configurar un job en Jenkins que ejecuta la herramienta generativa de tests y corre esos tests en un entorno temporal. Así medimos su efectividad sin afectar el pipeline principal. GitLab permite crear ramas de prueba con su propio pipeline; podría usarse una rama “pilot-IA” con etapas adicionales de IA integradas. Esto da flexibilidad para experimentar.

2.2) Sandboxes y entornos de QA

Si la empresa tiene entornos de QA, úselos para los pilotos en lugar de producción. Incluso se pueden levantar entornos efímeros en la nube para ciertas pruebas (por ejemplo, un entorno Kubernetes temporal para desplegar un componente resultante de IA y testearlo). Docker y la virtualización ayudan a encapsular pilotos de forma reproducible. Herramientas como Kind (Kubernetes en Docker) pueden crear clusters locales de k8s para probar, por ejemplo, un operador de IA en un entorno controlado.

2.3) Herramientas de medición y tracking

Para evaluar los pilotos, se pueden emplear herramientas analíticas especializadas. Existen plataformas de Engineering Analytics como Plandek o LinearB que integran con repos y pipelines para ofrecer métricas de

productividad y flujo. Estas podrían habilitarse al menos temporalmente durante el piloto para capturar datos detallados (commits, tiempos de ciclo, etc.) y compararlos con el rendimiento. Alternativamente, si no se quiere adquirir nuevas herramientas, usar lo que esté a mano: por ejemplo, extraer logs de builds y calcular manualmente tiempos, o usar scripts que consulten las APIs de la herramienta (GitLab API para obtener duración de pipelines, cobertura, etc.). Para medir percepción del equipo, usar encuestas breves (tipo Google Forms) pre y post piloto.

2.4) Herramientas IA específicas para cada piloto

Aquí la selección es amplia y depende del caso de uso. Algunos ejemplos:

- Asistentes de código: GitHub Copilot, Tabnine, Amazon CodeWhisperer. Copilot es muy popular (basado en GPT) y tiene extensiones en VSCode, JetBrains, etc. Para un piloto, se puede usar la prueba gratuita o licencias individuales.
- Generación de pruebas: herramientas como Diffblue Cover (para Java, genera tests unitarios), OpenAI Codex con prompts personalizados para crear tests a partir de descripciones, o Testim (plataforma de AI testing para front-end).
- Analítica de requerimientos: usar ChatGPT o Claude de Anthropic para analizar especificaciones en lenguaje natural. Por ejemplo, suministrar documentos de requisitos y pedirle a la IA extraer posibles casos de uso o detectar inconsistencias. Esto puede hacerse vía API con un script Python, por ejemplo.

- IA en DevOps/Monitoreo: probar herramientas de AIOps, como Dynatrace Davis (que usa AI para alertas) o el módulo de Machine Learning de Elastic Stack para detectar anomalías en logs durante despliegues.

Lo importante es usar versiones trial o entornos gratuitos en pilotos para minimizar costos. Muchas de estas herramientas ofrecen trial de 14-30 días, suficientes para un piloto. También considerar herramientas open source (por ejemplo, SonarQube tiene plugins con análisis inteligentes de código).

2.5) Control de versión para pilotos

Se recomienda aislar los cambios generados por IA en una rama separada. Por ejemplo, rama experimento-copilot. Esto facilita comparar con la rama principal las diferencias producidas por la IA. Además, en caso de problemas, se descarta la rama sin afectar nada. También se puede usar feature flags si aplica: desplegar una función producida con ayuda de IA tras un flag que solo se habilita para pruebas internas, y apagarla rápido si algo falla.

2.6) Herramientas de colaboración y revisión

Durante el piloto, es útil intensificar revisiones de código o pares de programación. Por ejemplo, usar GitHub Pull Requests como mecanismo para que otros revisen el código generado por IA, asegurando que cumpla estándares. Incluso el propio concepto de pair programming puede adaptarse: un desarrollador codifica y otro monitorea la sugerencia del asistente de IA, para aprender conjuntamente. Algunas organizaciones usan la técnica “prompt programming”: un miembro del equipo se especializa en escribir buenos prompts para obtener el mejor resultado de la herramienta, mientras

otro integra esas salidas al código. Esto puede formalizarse en el piloto como rol experimental.

3) Recomendaciones prácticas y criterios de validación

3.1) Acotar el alcance para demostrar valor

Un error común es querer que el piloto resuelva algo demasiado amplio, diluyendo los resultados. Es preferible un piloto pequeño que entregue un resultado claro. Por ejemplo, en vez de “usar IA en todo el proceso de testing”, enfocar en “generar pruebas unitarias para el módulo X con IA”. Así se sabrá con certeza si funcionó o no. El criterio de validación fundamental es: ¿El piloto probó la hipótesis? Si la hipótesis era “Copilot aumenta productividad”, ¿se observó aumento? Si era “ChatGPT mejora la calidad de requisitos”, ¿se encontraron menos ambigüedades? Cada piloto debe tener su hipótesis validada o refutada con evidencia.

3.2) Éxito o fracaso, ambos son útiles

Transmitir al equipo que fracasar en un piloto no es algo negativo, siempre y cuando deje lecciones. Esto anima a experimentar honestamente. Un criterio de éxito no es solo “el piloto dio mejoras”, sino también “el equipo aprendió qué no hacer”. Documentar por qué algo falló (por ejemplo, la IA generó código inseguro, o tardaba demasiado) es muy valioso. De hecho, un piloto podría ser declarado exitoso en términos de aprendizaje, aunque los KPIs no mejoren, si por ejemplo revela limitaciones que ahorrarán inversión posterior.

3.3) Implicar a usuarios finales si corresponde

Si el resultado del piloto toca algo que el cliente o usuario final ve (por ejemplo, un chatbot de soporte al usuario), es importante recoger feedback externo. En un piloto controlado, quizás se lo presente a un pequeño grupo de usuarios beta para que lo prueben y opinen. Su aceptación debe ser un criterio de validación si la solución es de cara al cliente. No serviría introducir IA que los clientes rechacen o no encuentren útil.

3.4) Tiempo y recursos dedicados

Asegúrese de que el equipo piloto esté liberado lo suficiente para enfocarse. Un piloto no rinde si sus encargados lo hacen “en sus ratos libres”. Es recomendable que en las sprints dedicadas al piloto, tengan objetivos ajustados de sus proyectos regulares para darles espacio. Un criterio de éxito de gestión es que el piloto se completó en el tiempo asignado y sin sobrecargar a los participantes.

3.5) Señales para escalar o abortar

Durante el piloto, hay que estar atento a señales tempranas. Por ejemplo, si en las primeras dos semanas la herramienta IA genera más retrabajo que ayuda, podría ser señal de abortar anticipadamente. Al contrario, si rápidamente muestra beneficios evidentes, tal vez se decida acelerar su adopción (pasar antes a fase 4 en ese ámbito). La flexibilidad es clave: la planificación no está escrita en piedra; los pilotos pueden hacer pivotear la estrategia (por ejemplo, descubrir un caso de uso mejor del imaginado).

Fase 4: Integración y escalamiento

1) Actividades clave

1.1) Integración de soluciones IA en el pipeline DevOps principal

Las iniciativas que resultaron exitosas en pilotos ahora se incorporan oficialmente en los procesos estándar de la fábrica de software. Esto requiere actualizar flujos de trabajo, documentación y posiblemente herramientas. Por ejemplo, si un asistente de código fue validado, se gestiona su habilitación para todos los desarrolladores (instalación de extensiones en IDE, configuración de autenticaciones, etc.) y se incorpora su uso en la guía de desarrollo (por ejemplo, indicando que se puede usar para soporte o apoyo, pero siempre con revisión humana en código crítico). Si se validó un sistema de pruebas con IA, se integra esa etapa en el pipeline CI/CD de todos los proyectos pertinentes. En síntesis, se pasa de la “prueba de concepto” a la operacionalización: la IA deja de ser experimental para volverse parte del día a día en la entrega de software.

1.2) Escalamiento a más equipos y proyectos

Paralelamente, se extiende la adopción más allá del grupo piloto. Es aconsejable hacerlo de forma gradual y controlada. Por ejemplo, sumar 2 o 3 equipos más en una segunda oleada, con el apoyo de miembros del equipo piloto como mentores. Luego, a medida que esos nuevos equipos alcanzan dominio, continuar hasta cubrir todos los equipos/proyectos objetivo. Durante el escalado, es fundamental mantener un seguimiento cercano: algunos

equipos podrían enfrentar dificultades particulares, y se debe brindar soporte (coaching adicional, resolver problemas técnicos específicos, etc.). El escalamiento no es solo “activar la herramienta en todas partes”, implica también ajustar: seguir comunicando beneficios, compartir historias de éxito interno de los pilotos para motivar a otros, y atender resistencia remanente. Un enfoque estructurado es entrenar a “champions” locales en cada equipo (personas entusiastas que aprendieron en el piloto) para que ayuden a sus compañeros.

1.3) Refuerzo de infraestructura y capacidades técnicas

Al escalar la adopción, las demandas sobre infraestructura pueden crecer. Se debe verificar y fortalecer: ¿los servidores CI/CD soportarán más carga si ahora incorporan pasos adicionales de IA? ¿Hace falta más potencia de cómputo (CPU/GPU) para entrenar modelos o procesar grandes volúmenes de datos continuamente? Por ejemplo, si antes un modelo se entrenaba esporádicamente en el piloto, ahora quizás se entrene con regularidad o con más datos, esto podría requerir instancias de nube más grandes o especializadas. Además, hay que considerar la confiabilidad: incorporar IA puede añadir puntos de falla nuevos (servicios externos, dependencias de red). Por ende, preparar redundancias o planes de contingencia: ejemplo, si un servicio de IA externo no responde, que el pipeline pueda continuar (tal vez con un modo degradado manual). También revisar temas de escalabilidad de licencias: si en piloto se usó una licencia community, ahora se podría necesitar una licencia enterprise o más usuarios. Gestionar la compra y distribución de esas licencias es una actividad de esta fase.

1.4) Gestión de nuevos perfiles y competencias

A mayor escala, puede requerirse integrar nuevo talento o roles. Por ejemplo, si la estrategia incluye desarrollar modelos de Machine Learning propios, quizás deba contratarse o reasignarse un científico de datos o un ML Engineer. O capacitar a personal existente para asumir esas tareas. Si en el diagnóstico se vio falta de habilidades, en esta etapa es crítico que las habilidades hayan sido desarrolladas; de lo contrario, se corre el riesgo de no poder sostener la solución. También pudiera surgir la necesidad de un administrador de la plataforma de IA (para gestionar herramientas, accesos, actualizaciones). Todas estas consideraciones deben ejecutarse: contratar, tercerizar o entrenar según sea el caso.

1.5) Gobernanza en práctica y control de cumplimiento

En escalamiento, se deben poner en marcha los mecanismos de gobernanza definidos en la fase 2. Por ejemplo, si se estableció un comité de IA, en esta etapa empieza a operar revisando periódicamente los avances, riesgos y cumplimiento de políticas. Supongamos que hay una política de “no usar datos de producción reales para entrenar IA sin anonimización”: a medida que se escale, habrá que auditar que los equipos la cumplan. También se consolida la gestión de riesgos dinámica: monitorizar continuamente si emergen nuevos riesgos o si alguno identificado se materializa, y tomar acciones. La gobernanza también implica medir el progreso global de la adopción: porcentajes de proyectos usando IA, feedback de calidad, etc., para informar a la dirección. Un punto de control es ética y seguridad: asegúrese de que, en el fervor de escalar, no se salten pasos de revisión ética o pongan datos en riesgo. Por ejemplo, si se empieza a usar masivamente un servicio de IA en la nube, corroborar que los contratos y medidas de seguridad estén en orden para ese volumen de uso.

1.6) Monitoreo unificado de métricas y desempeño

Con la IA integrada a gran escala, es posible comenzar a ver el impacto en las métricas globales de la organización. Se consolidan los datos de todos los equipos para evaluar mejoras en KPIs estratégicos definidos. Por ejemplo, comparar las métricas DORA promedio de la organización antes vs después: idealmente se observarán despliegues más frecuentes, menores tiempos de entrega y recuperaciones más rápidas gracias a automatizaciones inteligentes. De igual forma, métricas de calidad (densidad de defectos, satisfacción del cliente) deberían mostrar tendencia positiva si la IA ha estado bien aplicada. La función de monitoreo aquí es mantener un dashboard central (posiblemente presentado a alta gerencia) que muestre cómo la adopción de IA está contribuyendo a los objetivos. Esto se logra integrando las herramientas de métricas ya en uso (Jira, CI/CD, etc.) con tableros ejecutivos (quizá en PowerBI, Grafana, Google Data Studio) que combinen vistas de distintos equipos. El dashboard podría mostrar, por ejemplo: "Frecuencia de despliegue promedio: subió de 1 vez/semana a 3 veces/semana en 6 meses", "Lead time mediano: bajó de 7 días a 3 días", junto con historias cualitativas de éxito. Esta visibilidad ayuda a asegurar apoyo continuo y a detectar áreas rezagadas que necesiten ayuda extra durante el escalado.

2) Herramientas específicas recomendadas

2.1) Pipelines CI/CD reutilizables y escalables

Con la IA integrada, conviene optimizar la forma en que se la incorpora en múltiples proyectos. Herramientas como Jenkins ofrecen jobs

parametrizados o shared libraries que permiten definir un paso (por ejemplo, ejecutar pruebas con IA) y reutilizarlo en todos los pipelines. Otras herramientas modernas como GitHub Actions permiten crear actions reutilizables. Asimismo, tecnologías como Tekton o Argo Workflows (en Kubernetes) podrían estandarizar flujos CI/CD con pasos de IA para todos los equipos. El objetivo es evitar configurar manualmente cada pipeline de cada equipo, más bien proveer plantillas: por ejemplo, un template de pipeline YAML con el bloque de “análisis con IA” listo para incluir. Esto garantiza consistencia en la adopción y facilita el mantenimiento a futuro.

2.2) Dashboards centralizados

Aprovechar Grafana u otras herramientas de visualización para tener un panorama de la adopción. Grafana podría conectarse a múltiples fuentes: a la base de datos de Jenkins/GitLab para extraer métricas de proceso, a herramientas de seguimiento de tareas para métricas de velocidad, e incluso a hojas de cálculo o formularios donde se recolecta feedback. Un dashboard bien diseñado mostrará tanto métricas técnicas (por ejemplo, DORA) y, si se activó la capa verde, también métricas ambientales. Herramientas como PowerBI o Tableau también podrían usarse para combinar estos datos en reportes ejecutivos periódicos.

2.3) Integraciones y API management

Al escalar el uso de servicios IA externos (APIs de OpenAI, AWS AI Services, etc.), conviene gestionar eficientemente las integraciones. Por ejemplo, usar un API Gateway corporativo para centralizar las llamadas a servicios de IA, aplicando políticas comunes (límites de tasa de uso,

autenticación, logging de peticiones). Esto da más control y seguridad. También, si se tiene múltiples equipos llamando a la misma API, TI podría proveer una librería común que simplifica invocar la IA con buenas prácticas incorporadas (reintentos, manejo de errores, etc.).

2.4) Continuidad y actualización de herramientas IA

Establecer un calendario de revisión y actualización de las herramientas adoptadas. Por ejemplo, si se usa un modelo pre-entrenado, planificar re-entrenamientos con nuevos datos cada cierto tiempo para mantener su eficacia; si un proveedor lanza una nueva versión (Copilot mejora su modelo), probarla primero en un entorno controlado antes de desplegarla a todos. En esta fase, es útil tener ambientes de prueba permanente para IA: así como se tienen entornos de dev, QA y prod, puede haber un entorno “lab” donde se validan las actualizaciones de la herramienta IA antes de masificar.

2.5) Monitoreo y alerta inteligente

Aprovechar la escala para implementar AIOps a nivel de operaciones. Por ejemplo, ahora que hay muchos despliegues y componentes, usar herramientas de monitoreo avanzadas (como mencionadas Dynatrace, Datadog con AI) para correlacionar eventos. Se puede configurar que, al escalar a producción con IA, se active un monitoreo intensivo las primeras semanas para asegurarse que todo vaya bien, con IA detectando anomalías de performance o errores. Esto cierra el círculo de usar IA también en la etapa de operaciones, que inicialmente era un área de oportunidad.

3) Recomendaciones prácticas y criterios de validación

3.1) No descuidar la calidad y seguridad

Durante la integración masiva, hay que mantener los estándares. Un riesgo es que, por introducir nuevas herramientas, se relajen ciertos controles. Por ejemplo, si antes todo código pasaba por revisión de pares, ahora con IA generando partes, podría tentarse a saltarse revisiones. Esto sería un error. La recomendación es integrar la IA dentro del marco de calidad existente, no en sustitución. Un criterio para monitorear es que las métricas de calidad no empeoren con la IA; idealmente mejoren. Si la tasa de fallos en despliegues (Change Failure Rate) era 10% y tras integrar IA sube a 20%, algo anda mal (quizá introdujo complejidad). Debemos ver la tendencia opuesta: fallos iguales o menores, despliegues más confiables. Si alguna métrica de calidad se degrada, detenerse y hacer root cause analysis (¿la IA genera un problema? ¿falta capacitación?). La seguridad también: monitorear que no se introduzcan vulnerabilidades; por ejemplo, si se usa IA para código, podría sugerir una librería insegura – se debe continuar usando escáneres de seguridad y agregarlos al pipeline. En resumen, la incorporación debe fortalecer o al menos mantener la calidad, nunca comprometerla.

3.2) Ajustar la estructura organizacional si es necesario

A medida que la IA se vuelve parte central de las operaciones, considerar cambios formales en la estructura. Algunas organizaciones crean un Centro de Excelencia de IA o designan un AI Lead por departamento. Esto institucionaliza el conocimiento y soporte. Por ejemplo, un CoE (Centro de Excelencia) de IA puede encargarse de mantener actualizadas las mejores

prácticas, entrenar a nuevos empleados en el uso de las herramientas IA, evaluar nuevas tecnologías emergentes, etc. No todas las empresas necesitan un equipo separado, pero sí al menos roles definidos para continuar impulsando la iniciativa. Un criterio de madurez logrado sería que la responsabilidad de IA esté claramente asignada (más allá del proyecto inicial) y que existan procesos establecidos para su gobernanza continua.

3.3) Métricas de adopción y valor realizado

Al finalizar el escalamiento, se debe evaluar en qué medida se alcanzaron los objetivos estratégicos fijados. Esta es la validación más importante de la fase. Por ejemplo: si se quería +20% productividad, ¿lo muestran las métricas? Si se apuntaba a 30% menos tiempo de entrega, ¿se logró? Muchas veces, la plena realización de beneficios puede tomar más tiempo que la fase de escalado en sí, pero ya deberían verse tendencias claras. Si hay desviaciones significativas (metas no alcanzadas), analizar por qué: ¿fueron demasiado optimistas? ¿Hubo obstáculos no previstos? La estrategia podría ajustarse en consecuencia. Un indicador global puede ser el ROI: comparar el costo total de la iniciativa (horas dedicadas, licencias, etc.) vs los beneficios tangibles (ahorro de tiempo, más proyectos entregados, ingresos adicionales). Un ROI positivo validará la inversión.

3.4) Adopción cultural medida

Más allá de lo técnico, es bueno medir la adopción cultural. Por ejemplo, se podría hacer una encuesta similar a la del diagnóstico para ver cómo cambiaron las percepciones: ¿Ahora el equipo confía más en la IA? ¿La ve

imprescindible, opcional o molesta? Si tres de cada cuatro desarrolladores opinan que las herramientas IA son útiles y las usan a diario, es un triunfo cultural. La tesis encontró que luego de adoptar IA ampliamente, muchos profesionales la emplean diariamente y la normalizan en su flujo. Queremos ver eso replicado internamente: que se vuelva natural. Si detectamos áreas donde la IA no despegó (por ejemplo, un equipo que sigue sin usarla), investigar las causas (¿liderazgo del equipo escéptico? ¿caso de uso inapropiado?) e intervenir con apoyo específico.

3.5) Cierre formal de proyecto y transición a BAU (Business as usual)

Tras escalar con éxito, conviene cerrar el proyecto “adopción de IA” como tal y transicionar a que pase a ser parte del funcionamiento normal de la organización. Esto implica documentar la historia del proyecto, resultados obtenidos, y definir estructuras de soporte para el futuro. Un criterio de finalización puede ser una auditoría interna o externa que certifique que la empresa ha incorporado IA en sus procesos de manera eficaz. Por ejemplo, quizás buscar una certificación como ISO 56002 de gestión de la innovación, o al menos un reconocimiento interno. Esta formalidad da un sentido de logro al equipo y consolida la adopción.

Fase 5: Operación sostenible y mejora continua

1) Actividades clave

1.1) Operación diaria con IA integrada

En esta fase final, la adopción de IA deja de verse como un proyecto con principio y fin, para convertirse en parte intrínseca de la operación normal de la fábrica de software. Todas las prácticas implementadas siguen ejecutándose, ahora potenciadas por IA donde corresponda. Por ejemplo, los pipelines CI/CD rutinarios incluyen sus etapas de IA (pruebas automáticas, análisis de código), los desarrolladores usan asistentes de IA como parte de su IDE usual, las herramientas de monitoreo con IA vigilan las aplicaciones en producción, etc. El foco del equipo de tecnología está en mantener estables y optimizadas estas operaciones. Cualquier nueva iniciativa de desarrollo de software ya considerará el uso de IA por defecto. En esta etapa también se establecen tareas de mantenimiento: así como se da mantenimiento a aplicaciones, se mantiene la “infraestructura de IA” (actualizar versiones de herramientas, renovar tokens de API, re-entrenar modelos propios cuando sus predicciones empiecen a degradar, etc.). Se podría asignar permanentemente a DevOps la responsabilidad de la salud de las integraciones IA en los pipelines.

1.2) Monitoreo continuo de desempeño y valor logrado

La mejora continua requiere medir constantemente. Ya con la adopción a plena escala, se realiza un seguimiento regular de KPIs para asegurarse de que los beneficios se sostienen o aumentan. Esto incluye seguir midiendo las métricas DORA u operativas (¿Se mantiene alta la frecuencia de despliegues? ¿Los tiempos de entrega siguen mejorando? ¿La tasa de fallos disminuyó?) y también indicadores de negocio relacionados (¿Se cumplen mejor los plazos de proyectos? ¿Aumentó la satisfacción del cliente o del equipo de desarrollo?). Un punto crítico es verificar que la introducción de IA efectivamente produjo los resultados esperados. En esta fase debemos

corroborar esas mejoras con nuestros propios datos. Si algún indicador no muestra progreso o empeora, es una señal para una retrospectiva: ¿la IA no se utiliza adecuadamente? ¿O tal vez las metas eran demasiado ambiciosas? La mejora continua implica hacer ajustes basados en datos. Por ejemplo, si después de 6 meses la frecuencia de despliegues se estancó por debajo de lo objetivo, se investiga y quizás se identifica un cuello de botella restante (tal vez la IA en pruebas funciona bien, pero la aprobación de cambios sigue siendo manual y lenta; la mejora podría venir de automatizar esa aprobación con ciertas políticas). Es un ciclo de retroalimentación permanente.

1.3) Incorporación de métricas ambientales (si aplica la sostenibilidad)

En caso de haberse activado la capa verde, la operación sostenible significa que junto con los KPIs tradicionales, la organización monitorea sus indicadores ambientales vinculados al uso de IA. Por ejemplo, mediar cuales proveedores son carbono neutro (opcional cuales y cuantos clientes lo exigen). Estos datos se integran en los tableros de control. De esa forma, por cada objetivo de rendimiento se puede ver el costo ambiental asociado. La idea es que la conversación de negocio incluya estas métricas “en la misma sala” que costo, tiempo y alcance. Por ejemplo, al evaluar un sprint: no solo si fue rápido y de calidad, sino ¿fue energéticamente eficiente? La operación sostenible busca un equilibrio: mantener los beneficios de IA al mismo tiempo que se minimiza el impacto ambiental negativo. Esto puede llevar a medidas en la fase operativa, como utilizar servidores en regiones geográficas con energía renovable, etc.

1.4) Mejora continua e innovación incremental

La adopción de IA no termina nunca realmente; las tecnologías evolucionan y siempre surgen nuevas oportunidades. Por tanto, se establece un proceso formal o informal para identificar mejoras de forma continua. Esto puede estar apoyado por las retrospectivas de los equipos (en metodologías ágiles, en cada sprint se puede discutir “¿hay alguna tarea que la IA podría hacer mejor por nosotros?”), por un equipo de innovación que pilotee nuevas herramientas emergentes, o por concursos internos de ideas. La filosofía DevOps de Kaizen (mejora constante) se extiende a IA: quizás hoy tenemos X integrado, mañana sale una herramienta y que hace algo mejor, debemos estar dispuestos a probarla. En la operación se incluirá por ejemplo actualizar modelos con datos más recientes para mejorar su precisión (si se usa ML), refinar prompts de sistemas generativos a medida que se aprende qué formulaciones dan mejores resultados, etc. Un ejemplo de mejora continua: supongamos que tras 1 año usando IA en codificación, se tiene una gran cantidad de sugerencias rechazadas por los devs; se podría extraer esas sugerencias, analizarlas y ajustar los filtros o el modelo para que no repita los mismos errores. Lo importante es no quedarse estáticos: la operación sostenible implica evolucionar con la tecnología y lecciones aprendidas.

1.5) Soporte y gestión de incidentes en sistemas con IA

Una vez en operación, habrá casos en que las herramientas de IA mismas presenten fallos o generen resultados inesperados. Es vital integrarlas en el proceso de gestión de incidencias de TI. Por ejemplo, si el asistente de despliegue automatizado hace una acción errónea, se debe registrar como incidente, hacer post-mortem y ajustar. Puede requerir soporte del proveedor si es una solución externa. Hay que tener definidos SLAs y planes de soporte para las soluciones IA: ¿a quién se contacta si Copilot falla un día? ¿Qué hacer

si el servicio de IA de pruebas no está disponible horas antes de una entrega urgente? La operación madura contempla estos escenarios; quizás se mantienen planes de respaldo y se presiona a proveedores para mejorar disponibilidad.

2) Herramientas específicas recomendadas

2.1) Plataformas APM/AIOps en producción

Para mantener aplicaciones y servicios en operación con alta fiabilidad, herramientas de Application Performance Monitoring (APM) combinadas con AI son grandes aliados. Por ejemplo, Datadog ofrece detección de anomalías impulsada por machine learning en métricas de performance; New Relic y Dynatrace hacen análisis de causa raíz automatizado. Integrar estas plataformas permite que, con la mayor frecuencia de cambios lograda, el equipo de operaciones pueda identificar problemas proactivamente. Un caso concreto: tras muchos despliegues automatizados, usar AI Ops para correlacionar un aumento en tiempo de respuesta con cierto despliegue reciente y sugerir revertirlo antes de que afecte a usuarios. Esto reduce el MTTR aún más al tener “bots vigilantes” 24/7.

2.2) Extensión de dashboards incluyendo KPI de negocio y sostenibilidad

A nivel gerencial, usar herramientas tipo Power BI, Tableau o Google Data Studio para crear informes ejecutivos periódicos. Estos informes deberían combinar: métricas técnicas (las DORA, por ejemplo), métricas de negocio (cumplimiento de tiempos de proyecto, costos vs presupuesto,

satisfacción del cliente, etc.) y métricas ambientales (cumplimiento regulatorio con proveedores carbono neutrales). Por ejemplo, un reporte mensual al comité ejecutivo que muestre: “Implementamos 50 despliegues este mes (vs 30 hace un año), el tiempo de comercialización bajó 25%, entregamos con 15% menos bugs, y todo ello incrementó solo marginalmente el uso de energía en 5% gracias a optimizaciones verdes”. Presentar así los datos demuestra no solo el valor agregado de IA sino el control sobre su impacto. Estas herramientas permiten automatizar la actualización de datos y generar visualizaciones comprensibles para dirección.

2.3) Herramientas de medición de huella de carbono y eficiencia energética

Si se activó la sostenibilidad, se pueden emplear tanto servicios del proveedor de nube como herramientas independientes. Los grandes proveedores cloud ofrecen calculadoras de carbono: AWS Customer Carbon Footprint Tool, Azure Sustainability Calculator, Google Cloud Carbon Footprint. Estas herramientas dan reportes de las emisiones asociadas al uso de sus servicios – integrarlas nos permite obtener datos reales de nuestras cargas IA en la nube. También existen librerías como CodeCarbon (Python) que se integran en scripts de ML para registrar cuánta energía consumen y cuántos CO₂e producen, teniendo en cuenta la región geográfica. Usar estas librerías en nuestros procesos de entrenamiento de modelos, por ejemplo, agrega automáticamente logs de emisiones. Otra herramienta son los termómetros inteligentes y medidores si hay laboratorios de servidores: hardware IoT que mide temperatura, eficiencia de enfriamiento, etc., para optimizar aire acondicionado en centros de cómputo.

2.4) Plataformas de colaboración continua

La mejora continua también se facilita con herramientas tipo Retrospective Boards (por ejemplo: FunRetro, TeamRetro) para capturar lecciones aprendidas cada sprint, o Idea Management (por ejemplo: UserVoice, ideation boards en Teams) donde cualquiera pueda sugerir cómo usar IA en algo nuevo. Estas plataformas dan visibilidad a las ideas y tracking a su implementación. Por ejemplo, un desarrollador sugiere en el board: “¿Y si usamos ChatGPT para generar las notas de versión a partir de los commits?”; si tiene votos de otros, el equipo DevOps puede probarlo en la siguiente iteración. La organización debe proveer canales para que estas ideas surjan y se evalúen.

3) Recomendaciones prácticas y criterios de validación

3.1) Normalización y cultura

En la operación sostenible, el éxito verdadero se ve cuando la IA ya está completamente normalizada en la cultura organizacional. Los nuevos colaboradores al unirse aprenden tanto la pipeline DevOps como las herramientas de IA asociadas sin diferenciarlas. Un indicador cualitativo: que en las conversaciones diarias nadie diga “estamos haciendo un proyecto de IA”, sino que simplemente hablan de proyectos de software donde algunas tareas las realiza una herramienta X (que resulta ser IA). Es decir, que la IA esté invisible por lo integrada. Un hito de cultura podría ser actualizar los valores o manifiesto de la empresa para mencionar la innovación continua y el uso responsable de IA como parte de su identidad.

3.2) Revisión periódica de políticas y riesgos

La fase operacional no significa relajación. Se deben agendar revisiones regulares de las políticas de IA y del análisis de riesgos. Por ejemplo, semestralmente revisar: ¿han surgido nuevas preocupaciones éticas o legales? ¿Estamos cumpliendo todo? Ajustar las políticas corporativas si es necesario (las regulaciones de IA pueden cambiar rápidamente, o pueden encontrarse nuevos riesgos como las vulnerabilidades específicas de modelos). Un criterio de cumplimiento es pasar auditorías internas o externas: quizás la compañía se someta a una auditoría de cumplimiento de IA ética; aprobarla validará que la operación sigue estándares (similar a una ISO, pero en IA).

3.3) Sostenibilidad del impulso innovador

Mantener el interés y la motivación del personal en seguir mejorando con IA. A veces, tras lograr ciertas metas, puede haber complacencia. Para evitarla, se recomienda establecer incentivos: por ejemplo, premiar ideas innovadoras cada cierto tiempo, reconocer a equipos que logran mejoras excepcionales con IA, o incluso pequeños hackathons internos periódicos para experimentar con nuevas herramientas de IA. La mejora continua necesita nutrirse de la curiosidad y creatividad de la gente, no solo de procesos formales. Un criterio observable: número de iniciativas de mejora surgidas “desde abajo”. Si con el tiempo ese número decae, podría ser señal de fatiga o desinterés, y habría que revitalizar con alguna campaña.

3.4) ROI y justificación continua

En la operación estable, la inversión en IA debe seguir justificándose en términos de valor. La dirección podría preguntar: “estamos gastando X al año en licencias de IA, ¿vale la pena?”. Hay que estar preparados con datos para decir sí: por ejemplo, “gracias a IA liberamos N horas de trabajo mensuales, lo que equivale a Y dólares, más que cubriendo el costo”, o “hemos podido asumir 20% más proyectos con el mismo personal gracias a la eficiencia ganada”. Mantener este business case vivo ayuda a que ante recortes presupuestarios u otras prioridades, no se vea a la IA como algo prescindible. Más bien, mostrar cómo está entrelazada con el éxito operativo. Un criterio final puede ser lograr que la adopción de IA contribuya a ventajas competitivas tangibles: tiempos de entrega más rápidos que la competencia, mayor calidad reconocida por clientes, certificaciones o premios por innovación. Cuando la empresa comienza a percibir externamente los frutos (por ejemplo, clientes elogian que entregan software más rápido que otros proveedores), se confirma la sostenibilidad de la iniciativa.

3.5) Adaptabilidad a cambios futuros

La operación sostenible debe ser lo suficientemente flexible para adaptarse a cambios disruptivos. Por ejemplo, si surge una nueva generación de IA mucho más potente, la empresa debería poder adoptarla sin tener que reiniciar todo. Esto implica no casarse totalmente con una herramienta o proveedor, sino construir ciertas abstracciones. Un ejemplo: si se usó un servicio específico de AI y mañana se quiere cambiar a otro, tener interfaces modulares hará más fácil el cambio.

Dimensión opcional: Sostenibilidad ambiental (“capa verde”)

La sostenibilidad ambiental se presenta en el modelo como una capa transversal opcional, simbolizada en el diagrama como la “capa verde”. Esta dimensión aborda las implicaciones ecológicas de la adopción de IA y cómo gestionarlas. Incorporar esta capa es un paso adelantado que puede diferenciar a la fábrica de software y prepararla para futuras exigencias regulatorias o de mercado.

¿Cuándo activar la “capa verde”?

Lo ideal es considerar la sostenibilidad desde temprano, preferiblemente durante la Fase 2 (Planificación estratégica). En ese momento se decide si la organización asumirá compromisos ambientales en su adopción de IA. La decisión puede depender de los hallazgos del diagnóstico: por ejemplo, si se vio que actualmente no se presta atención al tema, pero la empresa tiene políticas de responsabilidad social corporativa o hay indicios de que clientes podrían valorarlo, es conveniente activarla. También puede desencadenarse por factores externos: nuevas regulaciones, requerimientos de un cliente importante que pide soluciones carbono-neutrales, o simplemente una visión estratégica de la dirección de ser pioneros verdes. Aunque es opcional, la recomendación es integrarla desde etapas tempranas como criterio estratégico, para cerrar la brecha antes de que se vuelva una exigencia impostergable. Dicho de otro modo, mejor prepararse en la tranquilidad de la proactividad que verse forzado luego. Si por alguna razón no se activó al inicio,

se puede incorporar más adelante (por ejemplo, al pasar a operación, al notar un alto consumo de recursos), pero será más difícil readaptar.

¿Cómo activarla e integrarla?

La capa verde añade un flujo paralelo de actividades ambientales a las fases principales:

- A) En Planificación estratégica: se establecen objetivos ambientales vinculados a la adopción de IA. Por ejemplo: “alcanzar un 50% de uso de energía renovable en los procesos de IA en 2 años”. Se designa un responsable (o varios) de sostenibilidad dentro del equipo del proyecto. Estos objetivos se incluyen en el plan de adopción como parte de los criterios de éxito (no separados). También se planean las iniciativas necesarias: ¿Se requerirá migrar infraestructura? ¿Buscar proveedores verdes? ¿Implementar mediciones nuevas? Además, desde aquí se sensibiliza a stakeholders: incluyendo discurso de IA responsable en la visión, y quizás asegurando presupuesto específico (por ejemplo, para compensaciones de carbono si hiciera falta). Como base, se puede realizar un análisis de ciclo de vida preliminar: estimar dónde en todo el ciclo de IA habrá mayor impacto (¿en entrenamiento de modelos? ¿en ejecución? ¿en hardware?), para enfocar esfuerzos.
- B) En Diagnóstico inicial (si no se hizo antes): aunque la capa se active formalmente en planificación, vale la pena llenar la evaluación inicial para incluir un rápido diagnóstico ambiental. Esto da una línea base ambiental. Si no se hizo en Fase 1, al activar la capa verde se podría ejecutar como mini-etapa: recabar esos datos base y prácticas actuales, para luego compararlos tras implementar medidas.

- C) Durante Pilotos: se incorporan consideraciones verdes en cada piloto que corresponda. Por ejemplo, si un piloto implica entrenar un modelo, se medirá la energía consumida por ese entrenamiento (usando herramientas como CodeCarbon) y se procurará ejecutarlo en horarios o entornos más eficientes. Si un piloto usa un servicio cloud, se elige a propósito una región verde. Llenar un checklist para seleccionar proveedores.
- D) En Integración y escalamiento: aquí la capa verde toma mayor protagonismo práctico. Se implementan acciones para reducir y compensar el impacto conforme la IA se extiende:
- Selección de proveedores e infraestructura verde: al integrar nuevas herramientas o mover cargas a la nube, elegir proveedores con fuerte compromiso ambiental. Considerar el checklist verde de proveedores y aplicarlo en las adquisiciones durante escalamiento.
 - También, consolidar infraestructura: si se tenían servidores subutilizados para IA, migrar a contenedores en cluster compartido para mejorar uso. Todas estas acciones se planifican y ejecutan en escalamiento para que cuando IA esté masiva, esté lo más afinada posible en consumo.
 - Compensaciones y políticas corporativas: en esta etapa se podrían instaurar políticas como: "Por cada proyecto de IA implementado, invertir X% del presupuesto de TI en certificados de energía renovable o reforestación". Algunas empresas optan por la vía de la compensación voluntaria mientras resuelven cómo reducir directamente sus emisiones. Además, se pueden alinear con metas globales (por ejemplo, unirse a iniciativas tipo Climate Neutral Now, o compromisos de AI sostenible). Escalar IA sosteniblemente es también educar a proveedores y clientes: una práctica podría ser incluir en propuestas comerciales un apartado de "Sostenibilidad de la solución: nuestra fábrica de software desarrollará su sistema usando prácticas carbono-neutrales...", lo cual puede sumar puntos con clientes concienciados.

E) En Operación continua: la sostenibilidad se vuelve parte de la monitorización regular y de la cultura de mejora:

- Fomentar cultura sostenible: al igual que con calidad o seguridad, incorporar la sostenibilidad en la cultura es clave. Esto puede significar hacer charlas internas de concienciación, reportar los logros de reducción de huella en reuniones de la empresa, e incluso ligar parte de los incentivos.

Ver el checklist de proveedores carbono-neutral (Kit_Adopción_IA.xlsx): cuando se evalúa contratar servicios en la nube o herramientas SaaS de IA, un checklist de sostenibilidad puede ser útil para comparar opciones.

C) Procedimiento de implementación

1) Prototipo

1.1) Contexto y objetivo

Se ejecutó un programa por fases para integrar un asistente de código con foco en productividad sin degradar la calidad. En el diagnóstico (F1) se identificó alta disposición (UTAUT-IA 3.63/5), se estableció la línea base DORA (LT P50=2.18 días; MTTR P50=3.52 h; CFR=16.14%) y se priorizó Copilot en tres casos de uso: unit tests, resúmenes de PR/changelog, refactor y docstrings.

1.2) Fases

Fase 1 Diagnóstico

Hallazgos clave: disposición positiva a IA (UTAUT-IA global 3.63/5), cuellos de botella en QA y PR reviews, y oportunidad en automatizar unit tests y documentación de cambios.

Decisión: avanzar a planificación con Copilot como iniciativa prioritaria (quick wins + apuesta estratégica).

Fase 2 Planificación estratégica (cerrada)

OKRs (SMART): reducir LT P50 $\geq 15\%$ sin subir defectos; CFR $\leq$ baseline; +10% despliegues/sem; adopción $\geq 70\%$; aceptación útil $\geq 60-65\%$.

Gobernanza y controles: políticas PI/licencias, gates de cobertura, librería de jobs YAML aprobados, trazabilidad en PR.

Roadmap del piloto: Ares (unit tests+docstrings), Circe (PR/changelog), Boreas (refactor acotado).

Caso de negocio preliminar: recomendación GO al piloto con escenarios (conservador/base/agresivo) y umbrales de salida.

Fase 3 Pilotos controlados

Resultados cuantitativos (vs. baseline):

- Ahorro en tiempo de desarrollo (ID-12): 18.8% (meta $\geq 15\%$).
- Cobertura de pruebas (ID-13): +8 pp (meta $\geq +5$ pp).
- Defectos críticos prevenidos (ID-14): 3.

- Adopción equipo piloto (ID-15, AE%): 75.7%.
- Satisfacción equipo (ID-16, SAT): 4.10/5.
- Tasa de éxito de pilotos (ID-17): 66.7% (2 de 3 con GO).

Decisiones por piloto: GO en Ares y Circe; Boreas a micro-piloto adicional por adopción <70%.

Fase 4 Integración y escalamiento

Cobertura de equipos: 70% (7/10).

Mejora DORA (agregado): $\Delta FD +12.0\%$ · $\Delta LT -16.1\%$ · $\Delta CFR -5.8\%$ · $\Delta MTTR -4.8\%$.

Uso diario (roles objetivo): 72.5%.

Fase 5 Operación sostenible y mejora continua

Estabilidad de proceso: $CV_{LT}=0.055$ (≤ 0.15).

Calidad postproducción: 0.12 defectos críticos/release (6 en 50).

Voz del cliente y equipo: NPS 36.7%; SAT interno 4.05/5.

Kaizen: 7 innovaciones implantadas en el trimestre; capa "verde" no activada (KPI-28 N/A).

1.3) Conclusión

El piloto cumplió sus metas y mostró mejoras sostenidas en flujo y calidad al escalar (LT, CFR, MTTR), con adopción diaria por encima del umbral.

Con los parámetros financieros del piloto (30 seats; 70% adopción; 24 min/día con 16.8 min útiles) se proyecta un ahorro explicado en el análisis financiero.

Se da una recomendación de mantener el ciclo de mejora continua (F5), reforzar seat management (uso diario $\geq 70\%$).

2) Criterio de expertos

Se consultó a un panel de 2 expertos para enriquecer la investigación con los siguientes comentarios y recomendaciones:

Gustavo Conejo Madrigal, presidente del PMI Capítulo de Costa Rica, Consultor, Co-Founder and CCO de Dolphins Consulting Group, Project / Program Manager / PMO, SMPC, Digital Transformation, Cobit, ITIL, Maestría en Administración de TI, Disciplined Agile, PMI, ISO 22301, CBCP, AI, BI.

ID	Comentario o Sugerencia
1	Explicar o detallar lo que significa DORA y UTAUT.
2	En la parte de Gobernanza y estructura del proyecto pondría algo visual, como una estructura jerárquica.
3	En página 12, matriz de riesgos, yo explicaría un poco sobre el uso de herramientas de riesgos de PMI y de CMMI, o al menos explicaría el por qué debe referenciarse a PMI y CMMI

4	Considerar dentro del documento traducir acrónimos al español, por ejemplo, si se indica FAQs debería indicarse entre paréntesis (Preguntas Frecuentes, por sus siglas en inglés).
5	Página 14, en obtención de quick wins, se menciona la palabra área chica, yo la cambiaría por área pequeña.
6	Página 20 el título principal indica "Recomendaciones prácticas y criterios de validación", sin embargo, en la lectura de los subapartados no se identifica cuáles son los criterios, esto mismo aplica para la fase 4 y fase 5.
7	Me parece muy acertado todo el apartado de Sostenibilidad, creo que eso es un punto muy relevante.

Tabla 3. Comentarios o Sugerencias Gustavo Conejo Madrigal

Roxana Trejos, Sr. Project Manager, Executive MBA, Business Analytics, Magister Scientiae, Information System, Computer Software Engineering, Data Science, Certification, Data-Driven Marketing.

Comentario o Sugerencia

Incluir la depreciación y desactualización de los equipos y servicios utilizados para correr los proyectos de IA.

Actualmente las empresas más importantes del planeta tal como: META, Microsoft, Oracle, entre otras; podrían afrontar grandes pérdidas en sus ganancias dado que no tomaron en cuenta de forma correcta la depreciación de sus data centers.

Por lo que recomiendo incluir una pequeña descripción sobre el tema en las siguientes fases y actividades:

- Fase 2. Agregar una actividad específica al manejo de la depreciación de la infraestructura dado la mejora continua de los proyectos IA.
- Fase 4. Actividad: Refuerzo de Infraestructura y capacidades técnicas.
- Fase 5. Herramientas: Herramienta de medición de huella de carbono y eficiencia energética.

CAPÍTULO VI. Análisis financiero

A) Costos

Para el análisis financiero se utilizarán los datos del prototipo para que pueda servir como insumo para los cálculos necesario.

Los salarios utilizados han sido tomados de los puestos publicados en LinkedIn y se han calculado con las cargas sociales de Costa Rica.

Rol / Recurso	F1	F2	F3	F4	Salario Mensual	% de Cargas Sociales	Cargas Sociales	Salario con Cargas Sociales	Cantidad de Recursos	Costo Mensual	Total horas	Costo por Hora	
PM	22.5	4	2	2	€2,000,000.00	40.16%	€803,200.00	€2,803,200.00	1	€2,803,200.00	30.5	€17,520.00	
Dev (TLs/representantes)	17	0	0	0	€1,300,000.00	40.16%	€522,080.00	€1,822,080.00	1	€1,822,080.00	17	€11,388.00	
QA (equipo)	14	0	0	0	€1,300,000.00	40.16%	€522,080.00	€1,822,080.00	1	€1,822,080.00	14	€11,388.00	
DevOps	18	0	12	20	€1,600,000.00	40.16%	€642,560.00	€2,242,560.00	1	€2,242,560.00	50	€14,016.00	
Seguridad	10	0	3	5	€1,600,000.00	40.16%	€642,560.00	€2,242,560.00	1	€2,242,560.00	18	€14,016.00	
Usuarios Dev/QA (formación)	—	—	84	152	€1,300,000.00	40.16%	€522,080.00	€1,822,080.00	30	€54,662,400.00	236	€341,640.00	
Costo por Fase	€1,139,676.00	€70,080.00	€28,943,040.00	€52,314,720.00				Totales	€12,754,560.00	35	€65,594,880.00	365.5	
Total implementación	€82,467,516.00												

Tabla 4. Costo RRHH

La tabla 7 muestra el costo de operación una vez implementado el prototipo, donde se excluye el salario de los desarrolladores porque ya es un costo incluido en la planilla de la fábrica de software.

Rol / Recurso	F5	Salario Mensual	% de Cargas Sociales	Cargas Sociales	Salario con Cargas Sociales	Cantidad de Recursos	Costo Mensual	Total horas	Costo por Hora
PM	2	€2,000,000.00	40.16%	€803,200.00	€2,803,200.00	1	€2,803,200.00	2	€17,520.00
DevOps	8	€1,600,000.00	40.16%	€642,560.00	€2,242,560.00	1	€2,242,560.00	8	€14,016.00
Seguridad	2	€1,600,000.00	40.16%	€642,560.00	€2,242,560.00	1	€2,242,560.00	2	€14,016.00
Costo por Fase	€175,200.00			Totales	€7,288,320.00	3	€7,288,320.00	12.0	
Total Operación Annual	€2,102,400.00								

Tabla 5. Costo Operación

B) Parámetros

Los parámetros de la tabla 8 fueron utilizados para el análisis financiero son los números que arrojaron el prototipo que sirven como insumo.

Parámetros		
Seats/Usuarios	30	Usu
Adopción	0.7	%
Min/día	24	min
Aceptación útil	16.80	min
Min/día	0.28	h/día
Días/mes	21	días laborables
Retrabajo base	6	horas
Reducción	5%	%
Costo Dev	₡11,388.00	colones
MTTR P50 (h) Antes	3.52	h
MTTR P50 (h) Despues	3.12	h
Calidad		
Incidente crítico evitado	₡89,422.08	mes
Resolución más rápida	₡79,260.48	mes
Calidad a 60%	₡1,023,557.64	mes
70% adopción	₡1,194,150.59	mes
Ahorro por tiempo (mensual)		
Productividad (asistente de código)	123.48	h/mes
Valorizado a tarifa Dev	₡1,406,190.24	mes
Ahorro por retrabajo (mensual)		
Retrabajo evitado	6.30	h/mes
Valorizado a tarifa Dev	₡71,744.40	mes
Ahorro por calidad (mensual)		
Escalado a 75%	₡1,279,447.06	colones
Total ahorro mensual (bruto)	₡2,757,381.70	colones
Ahorro annual (ingresos)	₡33,088,580.35	anual

Tabla 6. Parámetros

C) Flujo de Caja

Con base en el piloto y sus resultados (30 usuarios, adopción efectiva 70%, 24 min/día con 16.8 min útiles, 21 días laborables, reducción de retrabajo 5%, tarifa Dev \$11,388/h y operación F5 8h DevOps + 2h Seguridad + 2h PM/mes, Impuesto sobre la renta 20% y factor de descuento 15%), el ahorro bruto asciende a \$2,757,382/mes (\$33,088,580/año), compuesto por productividad \$1,406,190/mes, retrabajo evitado \$71,744/mes y calidad (críticos evitados + menor MTTR (Tiempo Medio de Reparación)) \$1,194,151/mes (referencia 70%, a 75% sería \$1,279,447/mes).

Deducida la operación e impuestos, el flujo neto anual queda en \$24,788,944, frente a una inversión inicial de \$82,467,516. Con estos flujos, el proyecto entrega VAN 15% = \$628,870 (ligeramente positivo), TIR = 15.32% y payback $\approx$ 4 años, la lectura del ROI anual simple es: 30.1%.

Flujo de Caja						
	0	1	2	3	4	5
INGRESOS (ahorros brutos)	€0	€33,088,580.35	€33,088,580.35	€33,088,580.35	€33,088,580.35	€33,088,580.35
EGRESOS (operación F5)	€0	€2,102,400.00	€2,102,400.00	€2,102,400.00	€2,102,400.00	€2,102,400.00
UTILIDAD NETA, antes de impuestos	€0	€30,986,180.35	€30,986,180.35	€30,986,180.35	€30,986,180.35	€30,986,180.35
Impuesto sobre la renta (20%)	-	€6,197,236.07	€6,197,236.07	€6,197,236.07	€6,197,236.07	€6,197,236.07
Utilidad neta después de impuestos	-	€24,788,944.28	€24,788,944.28	€24,788,944.28	€24,788,944.28	€24,788,944.28
Flujo neto después de impuestos	-	€24,788,944.28	€24,788,944.28	€24,788,944.28	€24,788,944.28	€24,788,944.28
Inversión Inicial	€82,467,516.00					
Flujo Neto en efectivo	-€82,467,516.00	€24,788,944.28	€24,788,944.28	€24,788,944.28	€24,788,944.28	€24,788,944.28
Flujo de caja acumulado	-€82,467,516.00	-€57,678,571.72	-€32,889,627.44	-€8,100,683.16	€16,688,261.13	€41,477,205.41
Factor de descuento (15%)	1	1.15	1.32	1.52	1.75	2.01
Flujo neto DESCONTADO	-€82,467,516.00	€21,555,603.72	€18,744,003.24	€16,299,133.25	€14,173,159.35	€12,324,486.39
VAN	€628,870					
TIR	15.32%					
Payback	4 años					
ROI Anual Simple	30.06%					

Tabla 7. Flujo de Caja

CAPÍTULO VII. Conclusiones y Recomendaciones

A) Conclusiones

- 1) A lo largo de este estudio se desarrolló un modelo de gestión que busca guiar la adopción de inteligencia artificial en fábricas de software, integrando aspectos técnicos, organizativos y ambientales. La propuesta se plantea como una hoja de ruta práctica que comienza con un diagnóstico inicial y culmina en una operación estable y sostenible. Un elemento distintivo del modelo es la incorporación de una “capa verde”, que permite considerar desde el inicio el impacto ambiental de la IA, algo poco explorado en la industria.
- 2) Los datos recolectados muestran que la inteligencia artificial ya está bastante presente en el trabajo diario de muchas fábricas de software. Herramientas como GitHub Copilot y ChatGPT se han vuelto comunes, especialmente en actividades de programación y análisis de requisitos. La mayoría de quienes las utilizan reporta mejoras en productividad, menos errores y una aceleración del proceso de entrega.
- 3) Uno de los hallazgos más claros fue que los principales obstáculos para adoptar IA no están ligados al dinero ni a la tecnología, sino a factores internos de las organizaciones. Falta de conocimiento especializado, temor al cambio, dudas sobre la seguridad de los datos y la ausencia de lineamientos éticos concretos fueron temas recurrentes. Por otro lado, se comprobó que cuando hay una cultura de innovación, apoyo desde la alta dirección y recursos tecnológicos adecuados, los resultados positivos llegan más rápido.
- 4) Un problema que también quedó en evidencia es que muchas empresas no están midiendo de forma sistemática los efectos de la IA. Esto dificulta saber con certeza si la tecnología está dando los resultados esperados. El estudio sugiere establecer métricas desde el inicio (como productividad, calidad,

tiempo de entrega y satisfacción) y hacer seguimiento continuo para tomar decisiones informadas y ajustar la estrategia cuando sea necesario.

- 5) La dimensión ambiental fue otro punto clave. El 76% de las organizaciones participantes no mide el consumo energético ni las emisiones asociadas al uso de IA. Ante este vacío, el modelo propone una alternativa concreta: integrar criterios de sostenibilidad desde las primeras etapas. Esto no solo permite reducir el impacto ecológico, sino que posiciona a las empresas de forma responsable y con ventaja ante futuras exigencias legales o del mercado.
- 6) Esta investigación plantea que el éxito de la inteligencia artificial no debe evaluarse únicamente en función del beneficio económico o de eficiencia. También es importante considerar el impacto social y ambiental que puede generar. La IA, bien gestionada, tiene el potencial de transformar positivamente las organizaciones, pero su implementación debe hacerse con una visión ética, responsable y sostenible.

B) Recomendaciones

- 1) Se recomienda a los directores de proyecto y oficinas de gestión (PMO): incorporar formalmente la inteligencia artificial dentro de sus procesos, utilizando el modelo presentado en este estudio como referencia. Esto implica patrocinar pilotos alineados con los objetivos estratégicos, definir métricas claras desde el principio y asegurarse de que exista una estructura de gobernanza que contemple el uso responsable, la seguridad y la ética.
- 2) Se recomienda a los gerentes de proyecto (PMs): aplicar los aprendizajes de este trabajo en la práctica diaria. Lo ideal es empezar con una etapa de diagnóstico para evaluar cómo está el equipo en cuanto a capacidades, cultura y herramientas. Luego, implementar pilotos controlados en ciclos

ágiles, donde se prueben soluciones de IA en tareas específicas. Es clave fomentar la formación continua y compartir los aprendizajes dentro de la organización.

- 3) Se recomienda a los directores de tecnología (CTOs): establecer una base clara para que la IA pueda integrarse correctamente en la organización. Esto incluye definir qué herramientas están autorizadas, establecer buenas prácticas de desarrollo asistido con IA, garantizar la seguridad de los datos y cuidar que la infraestructura técnica sea adecuada. Además, deberían empezar a evaluar el consumo energético de estas soluciones para promover un uso más sostenible de los recursos tecnológicos.
- 4) Se recomienda a los directores de operaciones (COOs): liderar el paso de la teoría a la práctica. Los proyectos piloto que funcionen bien deben integrarse al día a día de las operaciones. Para lograrlo, se requiere una coordinación estrecha entre distintas áreas y un seguimiento constante de indicadores clave como tiempos de entrega, calidad del producto y satisfacción del cliente. También es necesario estar atentos al impacto ambiental de estas tecnologías, preparando la organización para operar con eficiencia y responsabilidad en el futuro.
- 5) Se recomienda al PMI Capítulo Costa Rica: difundir las buenas prácticas y aprendizajes de este trabajo a través de iniciativas como el Green Playbook para Proyectos con IA. Esto podría incluir talleres, publicaciones y espacios de formación para fortalecer las capacidades de los directores de proyecto y ayudarles a gestionar la IA de forma sostenible. Además, sería útil incorporar herramientas prácticas para medir el impacto y guiar el cambio cultural dentro de las organizaciones.
- 6) Se recomienda a los líderes de innovación y transformación digital: tomar el modelo como base para introducir la inteligencia artificial de forma transversal en la empresa. Esto significa identificar oportunidades donde la IA pueda aportar valor, coordinar esfuerzos entre las áreas técnicas y de

negocio, y establecer indicadores que permitan evaluar el impacto real de estas iniciativas. También deberían actuar como facilitadores del cambio, compartiendo historias de éxito y promoviendo una cultura abierta al aprendizaje y la innovación.

- 7) Se recomienda a la alta dirección de las fábricas de software: integrar la inteligencia artificial como parte esencial de su estrategia organizacional. No se trata solo de implementar tecnología, sino de apoyar activamente su adopción con recursos, liderazgo y visión. También es importante que las decisiones se tomen considerando no solo los resultados económicos, sino también el impacto ético y ambiental de las soluciones, impulsando así una transformación responsable y duradera.

CAPÍTULO VIII. Análisis retrospectivo

Cuando ingresé a la maestría tenía altas expectativas de los cursos y del contenido de los mismo, pero fui identificando que el valor agregado, son los compañeros y la calidad de profesionales que son y con forme llegaron las entregas de trabajo era casi una “competencia” de intentar llegar a la perfección en los entregables, ya que sabías que los compañeros iban a poner la vara muy alta.

Luego el sacrificio familiar y social que significó, tengo una hija pequeña de 3 años que a su corta edad entendió lo que papá estaba haciendo cuando se encerraba por horas en la oficina.

Profesionalmente la MATI me abrió la perspectiva que tenía del mercado laboral, también me hizo darme cuenta de que estoy hecho para dar más, por lo que salí en búsqueda de nuevas oportunidades de crecimiento y curiosamente me pasó que una entrevista laboral coincidió con el desarrollo del TFG de graduación, específicamente con el análisis financiero y muchas preguntas de la entrevista coincidieron con lo que venía trabajando en el análisis, gracias a Dios, al esfuerzo y a la MATI me contrataron.

También la mentoría y profesionalismo de profesores como Harold Leiva nos muestran una senda profesional por donde quiero caminar y vivir.

Este Trabajo Final de Graduación terminó siendo una síntesis muy completa de lo que se trabaja e investiga durante toda la maestría, es la cereza en el pastel luego de casi 2 años de esfuerzo y dedicación.

Me gustaría luego del esfuerzo de la construcción de este “producto” validar la posibilidad de poder publicarlo y sea distribuido a quienes consideren necesario implementarlo en sus fábricas de software.

Por último, espero en un futuro cercano poder colaborar con la MATI en el área que se requiera, para mantener este sentido de pertenencia, puedo aportar mucho valor.

Referencias Bibliográficas

Alford, A. (2024). Study shows AI coding assistant improves developer productivity. InfoQ.

Baig, A. (2023). 15 principales estadísticas y tendencias de la IA. GlobalSign.

Columbus, L. (2024). Gartner's 2024 CEO survey reveals AI as top strategic priority.

Gartner, Inc. (2024). 2024 CEO Survey: The Year of Strategy Relaunches. Gartner.

Singla, A., Sukharevsky, A., Yee, L., Chui, M., & Hall, B. (2024). The state of AI in early 2024: Gen AI adoption spikes and starts to generate value. McKinsey & Company.

Statista. (2023). AI adoption rate by industry worldwide. Statista.

Yee, L., Hall, B., Singla, A., & Sukharevsky, A. (2023). The state of AI in 2023: Generative AI's breakout year. McKinsey & Company.

World Economic Forum. (2023). Global AI Investment Report 2023. WEF.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D. (2001). Manifesto for Agile Software Development. Recuperado de agilemanifesto.org

Capítulo III – Marco Metodológico. (2016). Metodología de la Investigación. Estudiantes UNESUR

Instituto Latinoamericano de la Comunicación Educativa – ILCE. (2023). La ética de la IA: 5 principios fundamentales. Recuperado de [Computing.es](https://www.computing.es)

International Organization for Standardization. (2019). ISO/IEC JTC 1/SC 42 – Artificial Intelligence. ISO.

Project Management Institute (PMI). (2017). A Guide to the Project Management Body of Knowledge (6.^a ed.). PMI.

Project Management Institute (PMI). (2017). Agile Practice Guide. PMI.

Project Management Institute (PMI). (2021). The Standard for Risk Management in Portfolios, Programs, and Projects. PMI.

Slack. (2023). "Gestión de recursos: qué es y cómo optimizarla". Blog de Slack.

Schwaber, K., & Sutherland, J. (2020). Guía Scrum: La Guía Definitiva de Scrum. Scrum.org.

Abad, J. (29 de abril de 2025). Lecciones Aprendidas. Obtenido de <https://www.lecciones-aprendidas.info/>

Arias, F. (2006). Proyecto de investigación: introducción a la metodología científica (5^o ed.).

Avalos, Z. C. (February de 2008). Fábrica Académica de Software.

Azkune, G. (2024). Inteligencia artificial: ¿más artificial que inteligente? Obtenido de Katedra: <https://katedra.eus/es/agenda/inteligencia-artificial-mas-artificial-que-inteligente-3>

Baig, A. (2023). 15 principales estadísticas y tendencias de la IA en 2023.

Blasco, J. L. (16 de julio de 2024). Métricas DORA: Revolucionando la eficiencia de DevOps. Obtenido de OpenWebinars: <https://openwebinars.net/blog/metricas-dora-devops/>

Cristian Rincón-Guio Ing. Esp. Mg. PhD(c), O. J. (2017). Proyectos, gestión y éxito. Una revisión de la literatura.

Dra. Nicole Forsgren, J. H. (2008). Accelerate: The Science of Lean Software and DevOps.

Escobar, C. J. (2024). Fábricas de software - Resumen de conceptos, modelos y enfoques.

Gartner. (2024). Gartner's 2024 CEO survey reveals AI as top strategic priority.

Iberia, D. (2024). La ética de la IA: 5 principios fundamentales.

IBM. (2025). Obtenido de ¿Qué son las redes neuronales?: <https://www.ibm.com/es-es/think/topics/neural-networks>

IBM. (2025). What is software development? Obtenido de <https://www.ibm.com/think/topics/software-development>

InfoQ. (2023). Study Shows AI Coding Assistant Improves Developer Productivity.

Institute, P. M. (2021). Guía del PMBOK® Séptima edición.

Kent Beck, M. B. (2001). Principios del Manifiesto Ágil. Obtenido de Manifiesto por el Desarrollo Ágil de Software: <https://agilemanifesto.org/iso/es/principles.html>

McKinsey Global Survey. (2023). El estado de la IA en 2023: El año clave de la IA generativa.

McKinsey Global Survey. (2024). The state of AI in early 2024: Gen AI adoption spikes and starts to generate value.

Microsoft. (2025). ¿Qué es DevOps? Obtenido de Azure Microsoft: <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-devops>

- Mucci, T. (2025). Obtenido de ¿Qué es la IA predictiva?: <https://www.ibm.com/es-es/think/topics/predictive-ai>
- Pérez, A. (2024). ¿Cómo Definir el Éxito de un Proyecto? Obtenido de COLEVEL: <https://www.ceolevel.com/como-definir-el-exito-de-un-proyecto/>
- Schuster, M. (2023). Métricas DORA y SPACE: 2 talleres de mejora en equipo. Obtenido de echometer: <https://echometerapp.com/es/dora-and-space-metrics/>
- Slack. (11 de julio de 2023). Gestión de recursos: qué es y cómo optimizarla.
- Sutherland, K. S. (2020). Guía de Scrum.
- The University of Texas at Austin. (2025). Obtenido de Ética de la IA: <https://ethicsunwrapped.utexas.edu/glossary/etica-de-la-ia>
- UNESCO. (2024). Obtenido de Artificial Intelligence: <https://www.unesco.org/en/artificial-intelligence>
- White, S. K. (2021). What is CMMI? A model for optimizing development processes. Obtenido de CIO: <https://www.cio.com/article/274530/cmmi-explained.html>
- Balestrini, M. (2006). *Como se Elabora el Proyecto de Investigación*. Consultores asociados.
- Beck, D. F. (2012). *Nursing research: generating and assessing*. Philadelphia: Williams & Wilkins.
- Camacho, H. C. (2008). *La Indagación: Una Estrategia*.
- Consensus. (2024). *Consensus*. Retrieved from About: https://consensus.app/?utm_source=chatgpt.com
- Johnson, S. (2024). How To Use NotebookLM As A Research Tool. *Medium*.

- Kauppaymuthoo, K. J. (2023). Artificial Intelligence Adoption in Software Development, A Qualitative Study on Developers' .
- Martínez-Salgado, C. (2012). El muestreo en investigación cualitativa. Principios básicos y algunas controversias.
- Martínez-Salgado, C. (2012). *El muestreo en investigación cualitativa: principios básicos y algunas controversias*. Retrieved from Scielo: <https://doi.org/10.1590/S1413-81232012000300006>
- NotebookLM. (2025). *NotebookLM*. Retrieved from Descripción General: https://notebooklm.google/?utm_source=chatgpt.com
- OpenAI. (2025). *OpenAI Platform*. Retrieved from Prompt engineering: https://platform.openai.com/docs/guides/prompt-engineering/prompt-engineering?utm_source=chatgpt.com
- Sampieri. (2020). Metodología de la Investigación.
- Sandelowski, M. (1993). *"Rigor or rigor mortis: the problem of rigor in*.
- Shenton, A. K. (2003). Strategies for ensuring trustworthiness in qualitative.
- Sybing, R. (2025). *Dominio de las entrevistas semiestructuradas*. Retrieved from Atlas TI: <https://atlasti.com/es/research-hub/entrevistas-semiestructuradas>
- Yin, R. K. (2009). Case study research: Design and methods (4th Ed.).
- Nishant, R., Kennedy, M., & Corbett, J. (2020). *Artificial intelligence for sustainability: Challenges, opportunities and a research agenda*. International Journal of Information Management, 53, 102104. <https://doi.org/10.1016/j.ijinfomgt.2020.102104>
- Wani, A. H., Raza, N., & Irfan, M. (2024). *Artificial intelligence applications in environmental pollution control: A comprehensive review*.

Environmental Science and Pollution Research.
<https://doi.org/10.1007/s11356-024-32404-z>

Yu, Z., Li, S., & Huang, H. (2024). *The impact of artificial intelligence on ecological footprint and low-carbon transition: Evidence from 67 countries*. Humanities and Social Sciences Communications, 11, 99.
<https://doi.org/10.1057/s41599-024-03520-5>

Anexos

A) Anexo 1: Cuestionario: Adopción estratégica de la Inteligencia Artificial en fábricas de software.

Estimado(a) profesional:

Le invito a participar en esta investigación académica que busca identificar prácticas, beneficios y desafíos en la adopción de Inteligencia Artificial (IA) dentro de fábricas de software de América Latina, así como explorar iniciativas para minimizar su impacto ambiental.

Sus respuestas serán utilizadas únicamente con fines académicos para diseñar un modelo de gestión que oriente la incorporación responsable y sostenible de IA en organizaciones de desarrollo de software.

Pregunta 1- ¿Cuál es su posición actual en la empresa donde se desempeña?
(Opcional)

Sección I – *Prácticas actuales de IA*

Pregunta 2- ¿Qué herramientas o tecnologías de IA utilizan hoy en el ciclo de vida del software? (Selección múltiple)

GitHub Copilot

ChatGPT

Gemini

Herramientas de pruebas basadas en IA

Analítica predictiva en CI/CD

Generación automática de documentación

Otro:

Pregunta 3- Frecuencia promedio de uso de IA en el trabajo diario (Selección única)

Diaria

Semanal

Mensual

Ocasional (≤ 4 veces/año)

Casi nunca

Pregunta 4- Etapas del ciclo de desarrollo donde se integra la IA (Selección múltiple)

Análisis / Refinamiento de requisitos

Programación / Codificación

Pruebas

Despliegue / DevOps

Soporte / Mantenimiento

Otro:

Sección II – Barreras y facilitadores

Pregunta 5- Obstáculos principales para adoptar IA (Selección múltiple)

Falta de habilidades especializadas

Resistencia cultural del equipo

Limitaciones de presupuesto

Seguridad/privacidad

Ausencia de estrategia clara

Ninguno relevante

Otro:

Pregunta 6- Factores que más han facilitado la adopción (Selección múltiple)

Apoyo de la alta dirección

Presupuesto dedicado

Capacitación interna/externa

Cultura de innovación

Infraestructura en la nube con IA

Otro:

Pregunta 7- Nivel de apoyo institucional a las iniciativas de IA (Selección única)

1 Nulo - 5 Total

Pregunta 8- ¿La organización cuenta con políticas o lineamientos sobre uso responsable/ético de IA? (Selección única)

Sí, formales y documentadas

Sí, en desarrollo

No, pero está planificado

No existe ninguna política

Sección III – Impacto y métricas

Pregunta 9- Beneficios tangibles percibidos tras adoptar IA (Selección múltiple)

Mayor productividad (> 20 %)

Reducción de tiempos de entrega

Menos defectos / retrabajo

Mejor precisión en estimaciones

Mejora en satisfacción del equipo

Aún no se perciben beneficios

Pregunta 10- Métricas en las que ha observado mejora gracias a IA
(Selección múltiple)

Frecuencia de despliegue

Lead Time de cambios

Tasa de fallos por cambio

MTTR (Tiempo Medio de Reparación (Mean Time to Repair))

No medimos DORA

No se observó mejora

Otro:

Pregunta 11- Impacto global de la IA en el rendimiento del ciclo de desarrollo de software (Selección única)

Muy positivo

Moderadamente positivo

Neutral

Negativo

No se ha evaluado

Sección IV – IA sostenible / Impacto ambiental

Pregunta 12- ¿Su organización mide o monitorea el impacto ambiental (energía, carbono, agua) derivado del uso de IA? (Selección única)

Sí, de forma sistemática

Sí, pero solo en proyectos puntuales

En fase de planificación

No realiza ninguna medición

Pregunta 13- ¿Qué acciones concretas han implementado para reducir la huella ambiental de la IA? (Selección múltiple)

Uso de data centers con energía renovable

Selección de servicios de IA con certificación carbono-neutro

Optimizar tamaño de modelos / eficiencia de cómputo

Compra de créditos de carbono o agua

Ninguna acción aún

Otro:

Pregunta 14- Al contratar proveedores de servicios de IA, ¿evalúan criterios de sostenibilidad? (Selección única)

Siempre

A veces según el proyecto

Rara vez

Nunca

Pregunta 15- ¿Algún cliente les ha solicitado que las soluciones de IA sean carbono-neutras? (Selección única)

Sí, varios clientes

Sí, un caso aislado

Aún no, pero lo están considerando

No se ha presentado

Pregunta 16- Frente a esas solicitudes (o futuras), ¿Cómo respondería su fábrica de software? (Selección múltiple)

Migrar a proveedores de IA certificados carbono-neutro

Adquirir compensaciones de carbono / agua

Ajustar la arquitectura para minimizar consumo

Negociar condiciones sin comprometer neutralidad

No tiene respuesta definida

Otro:

Pregunta 16- ¿Qué importancia estratégica le otorga su organización a la sostenibilidad ambiental de la IA? (Selección única)

1 Ninguna importancia - 5 Prioridad crítica

¡Muchas gracias por su tiempo y colaboración!

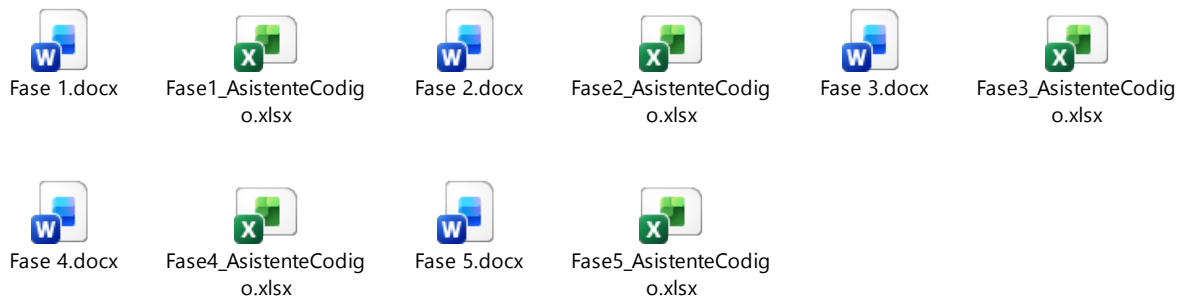
Su participación es invaluable para comprender cómo las fábricas de software adoptan la Inteligencia Artificial.

¡Le agradezco su contribución a esta investigación académica de la Universidad Nacional de Costa Rica!

B) Anexo 3: Kit de adopción de la solución



C) Anexo 4: Piloto Modelo de Adopción



D) Análisis Financiero

