

EQUIPOS DE SOFTWARE · ADOPCIÓN RESPONSABLE DE IA

Cómo los equipos de software de alto desempeño están usando IA hoy.

Una guía práctica para decidir, implementar, medir y gobernar la adopción de inteligencia artificial en equipos ágiles.

AUTOR

Juan Felipe Arboleda Salazar

EDICIÓN

Versión 1.0 · Abril de 2026

CONTACTO

juanf.arboleda@upb.edu.co

PARA

Equipos de software, líderes técnicos, gerentes de tecnología

Créditos y alcance

Esta página documenta el origen, el alcance y los límites de circulación de este ebook. Léase antes de citar, redistribuir o tomar decisiones organizacionales con base en su contenido.

Naturaleza del documento

Este ebook es una **guía profesional aplicada**: un producto de transferencia práctica que traduce hallazgos de literatura y prácticas emergentes en decisiones operativas para equipos de software. Es un material de circulación profesional controlada.

Origen académico

Este ebook nace como una **derivación aplicada** del trabajo de grado desarrollado por el autor en el marco de la Maestría en Gestión Tecnológica de la Universidad Pontificia Bolivariana. Su propósito es traducir hallazgos conceptuales y metodológicos sobre capacidades, productividad e inteligencia artificial en una guía práctica para equipos de software.

DISCLAIMER INSTITUCIONAL

Este documento es una producción académica y profesional independiente del autor, candidato a la Maestría en Gestión Tecnológica. Aunque se deriva de reflexiones y avances asociados a su trabajo de grado, su contenido **no representa necesariamente una posición oficial** de la Universidad Pontificia Bolivariana.

Lo que sí es y lo que no es

SÍ ES

- Una guía aplicada para equipos de software.
- Una traducción operativa de hallazgos de literatura y prácticas emergentes.
- Un material de transferencia práctica.
- Un cuaderno de trabajo con artefactos reutilizables.

NO ES

- Un artículo científico arbitrado.
- La tesis completa del autor.
- Un reporte definitivo de investigación.
- Una publicación oficial de UPB, de la Maestría o de GTI.UPB.

Estado del corpus de literatura

La base conceptual del ebook se apoya en un **proceso de revisión sistemática de literatura**, conducido por el autor bajo las orientaciones metodológicas de Kitchenham (2004) para revisiones en ingeniería de software y los lineamientos PRISMA 2020 (Page et al., 2021). El proceso incluyó las etapas de identificación, depuración, cribado, evaluación a texto completo e inclusión final de estudios, con cri-

terios explícitos de selección, elegibilidad y pertinencia temática. Las cifras detalladas del flujo PRISMA, así como los criterios aplicados en cada etapa, pueden consultarse en la *Nota metodológica* y en el *Anexo metodológico*.

En consecuencia, las recomendaciones operativas de este ebook deben leerse como **buenas prácticas plausibles e hipótesis aplicadas**, no como conclusiones empíricas validadas. La distinción entre evidencia, interpretación y recomendación se hace explícita a lo largo del documento.

Cómo citar este documento

Arboleda Salazar, J. F. (2026). *Cómo Los equipos de software de alto desempeño están usando IA hoy. Una guía práctica para decidir, implementar, medir y gobernar La adopción de IA en equipos ágiles* (Versión 1.0, edición 2026). Documento profesional aplicado.

Contacto

Para feedback, sugerencias, errores detectados o solicitudes de versión actualizada: juanf.arboleda@upb.edu.co.

Nota del autor

Por qué este ebook se parece más a un cuaderno de trabajo que a un ensayo, y para qué sirve cuando un equipo lo abre por primera vez.

Este ebook nace como una **derivación aplicada** de un trabajo de grado en curso. La pregunta que lo originó es práctica: por qué algunos equipos convierten la inteligencia artificial en valor sostenido y otros, con las mismas herramientas, terminan con más deuda técnica de la que empezaron. La respuesta corta —que la IA puede actuar como amplificador del sistema sociotécnico que la rodea— necesita una guía operativa para volverse útil.

La versión que tienes en las manos no es un ensayo. Es un cuaderno de trabajo. Cada sección deja algo accionable: una matriz, un playbook, una plantilla, un protocolo, una métrica. La intención es que, al cerrarlo, un equipo tenga claro qué hacer mañana y qué evitar.

La base conceptual del documento se apoya en un proceso de revisión sistemática de literatura ejecutado por el autor bajo el protocolo Kitchenham (2004) y los lineamientos PRISMA 2020. La parte aplicada del ebook —herramientas, playbooks, plantillas, prompts— se basa en la lectura aplicada de prácticas emergentes en equipos profesionales y en la documentación pública de las herramientas.

Por eso, en cada sección del ebook hago una distinción explícita:

- **Evidencia del corpus.** Cuando una afirmación se apoya en un estudio o marco teórico, se cita.
- **Interpretación aplicada.** Cuando es lectura del autor, se presenta como tal.
- **Recomendación práctica.** Reglas operativas que el equipo puede aplicar; no son estándares universales.
- **Ejemplo operativo.** Casos ilustrativos basados en patrones frecuentes, no en estudios de caso validados.

Si este documento te resulta útil, escríbeme. El feedback de profesionales que estén adoptando IA en sus equipos de manera real es lo que mejorará las próximas versiones.

Tabla de contenido

El ebook se organiza en cinco bloques. Cada bloque tiene una entrada práctica y artefactos listos para usar; los bloques se pueden leer en orden o por consulta puntual según el rol y la necesidad.

BLOQUE I · APERTURA, PANORAMA Y MÉTODO

- 01 Créditos y alcance
- 02 Nota del autor
- 03 Lo esencial en una mirada
- 04 Cómo usar este ebook según tu rol
- 05 Qué hacer mañana
- 06 Introducción
- 07 Nota metodológica · de la RSL a la práctica
- 08 Diagrama PRISMA del proceso de revisión
- 09 Qué sugiere la literatura sobre productividad ágil
- 10 Consideraciones para equipos en Colombia y LATAM
- 11 Qué hacen distinto los equipos de alto desempeño
- 12 Uso individual, de equipo y organizacional

BLOQUE II · HERRAMIENTAS Y ARTEFACTOS OPERATIVOS

- 13 Herramientas núcleo de IA para desarrollo
- 14 Herramientas de prototipado y validación rápida
- 15 Herramientas para QA, seguridad y observabilidad
- 16 Árbol de decisión · qué herramienta usar
- 17 Matriz comparativa de herramientas

- 18** Semáforo de datos en prompts
- 19** Playbooks prácticos
- 20** Prompts profesionales reutilizables

BLOQUE III · PRÁCTICAS, MÉTRICAS Y RIESGOS

- 21** IA en Scrum, Kanban, DevOps y producto
- 22** Mini ejemplos por rol
- 23** Productividad · qué medir y qué no
- 24** Panel mínimo de productividad
- 25** Costos visibles y costos ocultos
- 26** Riesgos, señales tempranas y controles

BLOQUE IV · SÍNTESIS · FRAMEWORK, MADUREZ Y PLAN

- 27** Framework integrador
- 28** Modelo de madurez de adopción
- 29** Diagnóstico rápido
- 30** Roadmap 30 / 60 / 90
- 31** Diez principios
- 32** Quick Start · 30 minutos
- 33** Conclusión

BLOQUE V · ANEXOS METODOLÓGICOS Y PRÁCTICOS

- 34** Anexo metodológico
- 35** Corpus consolidado
- 36** Anexos prácticos · plantillas

37 Trazabilidad de evidencia

38 Referencias

Lo esencial en una mirada

03

Una página tipo Quick Start. Pensada para que un lector ocupado se la lleve sin necesidad de leer el resto.

CINCO HIPÓTESIS PRÁCTICAS

1. La IA no parece sustituir capacidades; tiende a amplificarlas. El resultado depende del sistema sociotécnico que la recibe.
2. La productividad no es solo velocidad. Sin observar calidad, bienestar y valor, la velocidad puede ser falsa productividad.
3. Los equipos con mejores prácticas de revisión y de criterios de aceptación parecen capturar más valor de la IA.
4. Sin gobernanza de datos y de prompts, la IA puede introducir riesgos silenciosos que se revelan en horizontes de meses.
5. El pensamiento crítico es una capacidad que conviene proteger explícitamente; la aceptación pasiva es el principal antipatrón.

CINCO DECISIONES QUE TODO EQUIPO DEBE TOMAR

1. Qué herramientas se autorizan y quién las administra.
2. Qué datos pueden enviarse a servicios externos y cuáles no.
3. Qué tareas pueden delegarse a la IA y cuáles requieren juicio humano.
4. Qué cambios requieren revisión humana obligatoria.
5. Qué métricas se usarán para evaluar impacto en las cuatro dimensiones de productividad.

CINCO ERRORES QUE CONVIENE EVITAR

1. Medir solo velocidad y celebrar mejoras que ocultan deterioros en otras dimensiones.
2. Llevar prototipos a producción sin reescritura, pruebas, seguridad y revisión de arquitectura.
3. Delegar el juicio técnico al modelo en lugar de delegar únicamente la mecánica.
4. Adoptar demasiadas herramientas a la vez sin que el equipo domine ninguna.
5. Ignorar la observabilidad de flujos asistidos por LLM y la seguridad del código generado.

Flujo recomendado de adopción

01

Diagnóstico

02

Piloto controlado

03

Reglas mínimas

04

Integración al SDLC

05

Medición multidimensional

06

Escalamiento responsable

Cómo usar este ebook según tu rol

LECTURA COMPLETA O REFERENCIA PUNTUAL

El documento está diseñado para leerse completo, pero también para usarse como referencia puntual. La tabla siguiente sugiere por dónde empezar según tu rol, qué aplicar de inmediato, qué resultado esperar y qué riesgo principal vigilar.

ROL	SECCIONES RECOMENDADAS	APLICAR DE INMEDIATO	RESULTADO ESPERADO TRAS LEERLO	RIESGO PRINCIPAL A VIGILAR
Desarrollador	Herramientas núcleo · Playbooks prácticos · Prompts profesionales · Riesgos	Prompts de testing, debugging y refactor; checklist de Pull Request (PR) asistido	Flujo de trabajo asistido sin deteriorar comprensión técnica	Aceptar sugerencias sin comprenderlas
Arquitecto de software	Herramientas núcleo · Matriz comparativa · Framework integrador · Modelo de madurez	Plantilla de prompt para decisión arquitectónica; semáforo de datos	Criterios sólidos para elegir herramientas y delimitar usos	Erosión del juicio sistémico por delegación excesiva
Líder técnico	Playbooks prácticos · Riesgos · Diagnóstico rápido · Roadmap 30/60/90 · Modelo de madurez	Política de revisión humana; biblioteca compartida de prompts	Protocolo mínimo de adopción de IA del equipo	Diffs grandes asistidos por IA sin disciplina de revisión
Scrum Master	Equipos de alto desempeño · IA en Scrum/Kanban · Diagnóstico rápido · Roadmap	Retrospectiva temática sobre IA; protocolo de Pull Request (PR) asistido	Visibilidad de los efectos del uso desigual de IA en el equipo	Aparición de Community Smells por uso desigual
Product Owner	Herramientas de prototipado · Playbook 5 · Prompts (historia y AC) · Productividad	Plantillas de historia y de criterios de aceptación; refinamiento asistido	Backlogs más nítidos sin reemplazar conversación con stakeholders	Backlogs vagos amplificados por IA

ROL	SECCIONES RECOMENDADAS	APLICAR DE INMEDIATO	RESULTADO ESPERADO TRAS LEERLO	RIESGO PRINCIPAL A VIGILAR
Gerente de proyecto	Matriz comparativa · Productividad · Panel mínimo · Diagnóstico · Roadmap	Diagnóstico puntuable y panel multidimensional desde el inicio	Decisiones balanceadas más allá de la velocidad	Optimizar solo velocidad e ignorar bienestar y valor
Gerente de tecnología / líder organizacional	Nota metodológica · Framework · Modelo de madurez · Roadmap · Principios · Anexo metodológico	Modelo de madurez como diagnóstico organizacional; políticas de IA	Marco para gobernar herramientas, datos, seguridad y costos	Adoptar IA sin gobernanza ética y de datos

RUTA CORTA PARA TOMADORES DE DECISIÓN

Si dispones de 30 minutos, lee la sección **Lo esencial en una mirada**, el **Modelo de madurez**, el **Diagnóstico rápido puntuable** y el **Roadmap 30/60/90**. Con eso puedes llevar a una próxima reunión un punto de partida defendible y cuantificable.

Qué hacer mañana

05

Diez acciones inmediatas que un equipo puede ejecutar en su próxima semana de trabajo. Ninguna requiere autorización presupuestal mayor; todas dejan un artefacto compartido que sirve como base para crecer.

#	ACCIÓN INMEDIATA	RESPONSABLE SUGERIDO	ARTEFACTO RESULTANTE	SEÑAL DE ÉXITO
1	Definir las dos herramientas iniciales del equipo	Tech Lead + equipo	Lista breve de herramientas autorizadas	Nadie usa herramientas fuera del acuerdo inicial
2	Crear el semáforo de datos del equipo	Seguridad + Tech Lead	Semáforo publicado en wiki o README	El equipo sabe qué no debe enviar a prompts externos
3	Definir y aplicar la etiqueta ai-assisted en PRs	Tech Lead	Convención y plantilla de descripción de PR	Todos los PRs asistidos quedan trazables
4	Aplicar el checklist de PR asistido en el próximo PR	Cualquier desarrollador	PR con checklist completo en la descripción	Revisión humana documentada y verificada
5	Crear la biblioteca compartida de prompts	Tech Lead + voluntarios	Página viva con prompts del equipo y contexto	Tres prompts usados por más de una persona
6	Aplicar el diagnóstico rápido al equipo	Scrum Master / Tech Lead	Diagnóstico individual + sesión grupal	Top 3 capacidades débiles identificadas
7	Seleccionar las cuatro métricas mínimas (una por dimensión)	Gerente de proyecto + Tech Lead	Panel mínimo definido y publicado	Las cuatro métricas se reportan al menos quincenalmente
8	Acordar política de no envío de datos sensibles a prompts externos	Tech Lead + seguridad	Cláusula corta en la política de equipo	Casos dudosos llegan al canal correcto antes de enviarse
9	Programar una retro temática sobre uso de IA	Scrum Master	Acta con tres acuerdos accionables	Acuerdos ejecutados antes de la siguiente retro

#	ACCIÓN INMEDIATA	RESPONSABLE SUGERIDO	ARTEFACTO RESULTANTE	SEÑAL DE ÉXITO
10	Elegir un piloto de bajo riesgo para 30 días	Tech Lead + Product Owner	Documento de alcance del piloto con métricas base	Piloto cerrado en 30 días con reporte claro

REGLA DE ORO

Hacer poco y hacerlo bien. Tres acciones cumplidas valen más que diez iniciadas sin disciplina.

Introducción

06

Un nuevo punto incómodo de la conversación sobre productividad: la IA dejó de ser un experimento para convertirse en parte estructural del flujo de software.

La conversación sobre productividad en el desarrollo de software siempre fue resbaladiza. Los equipos sienten que entregan más, los líderes piden gráficos para confirmarlo y la literatura advierte que la velocidad sin contexto es un mal indicador. En la última década esa conversación se complicó con una capa nueva: la inteligencia artificial dejó de ser un experimento para convertirse en parte estructural del flujo. Hoy un copiloto sugiere funciones, otro modelo escribe pruebas y un tercero genera historias de usuario.

Este ebook nace en ese punto incómodo. Recoge una síntesis de literatura sobre productividad en equipos ágiles y la cruza con prácticas emergentes de adopción de IA. La intención no es repetir un manual de herramientas: hay decenas. La intención es responder, con criterio, a una pregunta concreta: cómo puede un equipo de desarrollo de software usar IA de manera responsable, medible y productiva sin deteriorar calidad, seguridad, aprendizaje ni pensamiento crítico.

La brecha entre teoría y práctica

La literatura revisada sugiere un patrón. La productividad en un equipo ágil no parece explicarse por la suma de métricas individuales ni por la adopción literal de un marco. Tiende a aparecer cuando el sistema sociotécnico funciona: cuando la cultura permite el aprendizaje, cuando el liderazgo ejerce autoridad sin asfixiar la autonomía, cuando la infraestructura técnica reduce fricción y, sobre todo, cuando la red social del equipo —confianza, comunicación, seguridad psicológica— se mantiene en buen estado.

La práctica suele invertir las prioridades: primero las herramientas, luego los procesos y al final lo humano. La IA está acelerando esa distorsión. Muchas organizaciones adoptan asistentes un viernes, miden ahorro de teclas el lunes y declaran transformación el martes. La parte difícil sigue intacta.

La tesis del ebook

La IA no es una solución mágica ni un sustituto del equipo. Puede actuar como amplificador del sistema sociotécnico. Si el equipo tiene buenas capacidades, la IA tiende a multiplicarlas. Si tiene fallas de cultura, liderazgo, calidad, seguridad o gobernanza, la IA puede amplificar también la fragilidad. El resultado depende menos del modelo y más del sistema que lo recibe. Esa idea atraviesa el ebook, pero no se repite por sí misma: cada sección la convierte en una decisión, una métrica, un riesgo o una herramienta concreta.



La IA no sustituye capacidades; las amplifica. Su impacto depende del sistema sociotécnico que la recibe.

Nota metodológica · de la RSL a la práctica

07

Este ebook es una traducción aplicada de hallazgos de literatura a prácticas concretas. No es una revisión académica completa ni un estudio empírico con datos propios validados en campo.

De dónde salen las ideas

La base conceptual proviene de un proceso de revisión sistemática de literatura conducido por el autor bajo el protocolo de Kitchenham (2004) y los lineamientos PRISMA 2020 (Page et al., 2021). El protocolo busca identificar, clasificar y estructurar capacidades organizacionales, tecnológicas y sociotécnicas asociadas a la productividad de equipos ágiles en contextos de desarrollo de software, con una ventana temporal de enero de 2001 a marzo de 2026.

Las cifras del flujo PRISMA se resumen así:



Al cierre de esta edición se completaron las etapas de **identificación, depuración, cribado, recuperación y evaluación a texto completo**. Como resultado del proceso, se consolidó un corpus final de **34 estudios incluidos**, seleccionados a partir de criterios explícitos de elegibilidad, pertinencia temática y aporte al objeto de estudio. El detalle del flujo, los criterios aplicados y la justificación de las exclusiones se documentan en el *Diagrama PRISMA* y en el *Anexo metodológico*.

ACLARACIÓN IMPORTANTE

Las afirmaciones de este ebook se construyen sobre dos fuentes complementarias: **(a) marcos teóricos clásicos** bien establecidos (Trist, 1981; Edmondson, 1999; Teece, 2007; entre otros), citados como guía conceptual; y **(b) prácticas emergentes** documentadas públicamente por equipos profesionales y proveedores de herramientas. Las cifras, efectos reportados y hallazgos provenientes de los estudios incluidos en el corpus final se incorporan únicamente cuando cuentan con trazabilidad metodológica suficiente dentro del proceso de revisión. Por tanto, las recomendaciones del ebook deben entenderse como una síntesis aplicada de evidencia documental, interpretación conceptual y lectura profesional del fenómeno, no como resultados de validación experimental propia ni como afirmaciones causales definitivas.

Cómo leer las afirmaciones del ebook

- **Evidencia del corpus.** Cuando una afirmación se apoya en un marco teórico clásico citado, se identifica con autor y año.
- **Interpretación aplicada.** Cuando es lectura del autor sobre el cruce entre literatura y práctica, se presenta como tal.
- **Recomendación práctica.** Reglas operativas adaptables al contexto del equipo lector; no son estándares universales.
- **Caso ilustrativo.** Etiquetado explícitamente como tal; basado en patrones frecuentes en equipos profesionales, no en un estudio de caso validado.
- **Plantillas, prompts y playbooks.** Artefactos derivados de la lectura aplicada de prácticas emergentes; pensados para ser adaptados.

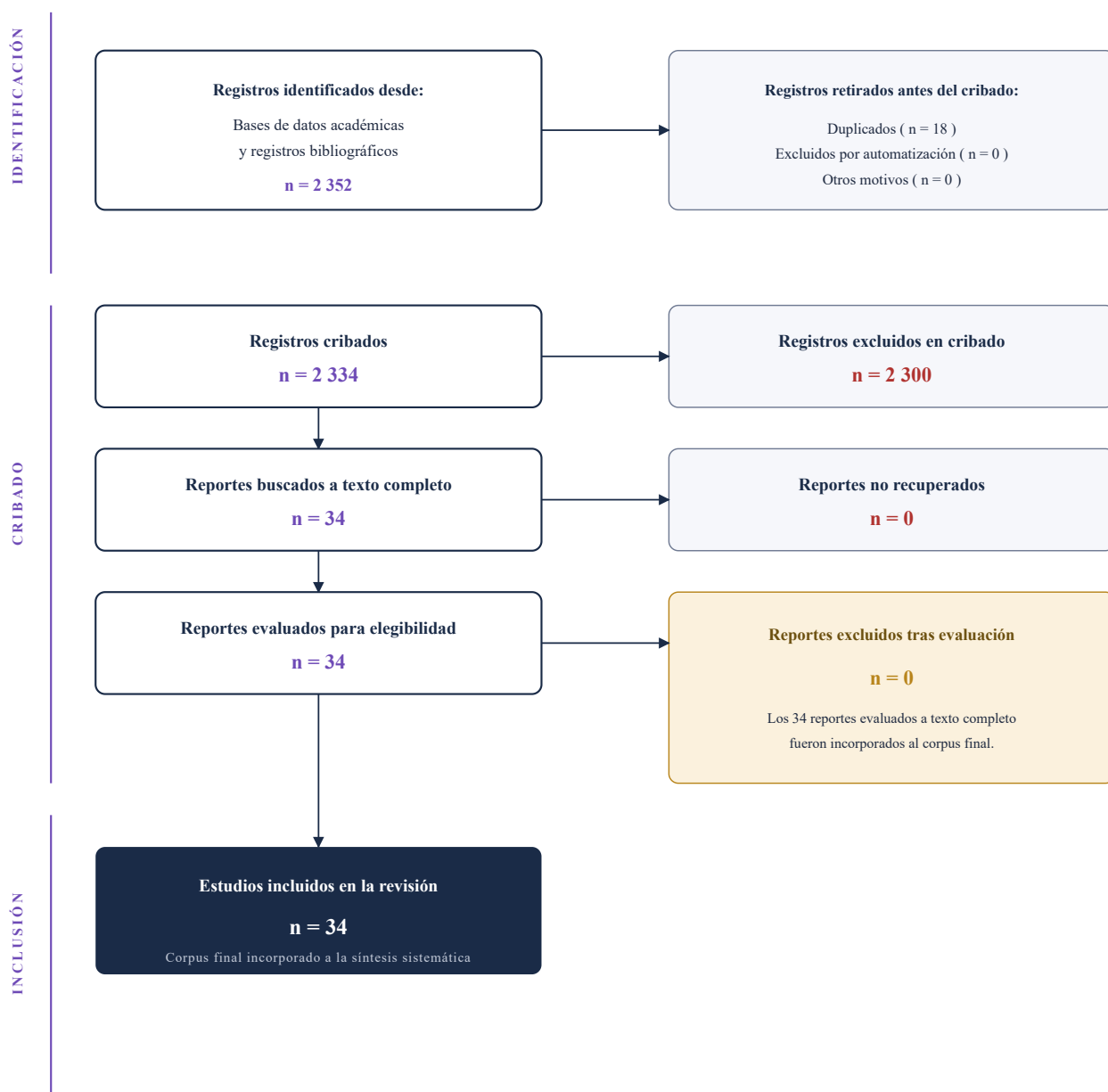
IMPLICACIÓN PRÁCTICA

Las reglas operativas (umbrales, métricas, pasos) deben leerse como **umbrales operativos sugeridos, no estándares universales**. Cada organización debe ajustarlos a su contexto regulatorio, sector, tamaño y madurez tecnológica antes de aplicarlos.

Diagrama PRISMA del proceso de revisión

08

Flujo PRISMA 2020 al cierre de esta edición. Las cifras corresponden al proceso de revisión documental finalizado, incluyendo identificación, depuración, cribado, recuperación, evaluación a texto completo e inclusión final de estudios.



LECTURA DEL DIAGRAMA

El proceso de identificación, depuración y cribado partió de 2 352 registros iniciales. Tras retirar 18 duplicados, se cribaron 2 334 registros, de los cuales 2 300 fueron excluidos por título, resumen y palabras clave. Los 34 reportes restantes fueron recuperados, evaluados a texto completo e incorporados al corpus final de la revisión. En consecuencia, las recomendaciones de este ebook se apoyan en una síntesis sistemática de literatura finalizada, complementada con marcos teóricos consolidados e interpretación aplicada del autor.

Qué sugiere la literatura sobre productividad ágil

09

Antes de hablar de herramientas, conviene fijar el terreno. La literatura sobre productividad en equipos ágiles, leída con atención, sugiere convergencias temáticas relevantes para entender por qué la IA puede amplificar o erosionar el desempeño según el sistema en que se inserte.

Centralidad del subsistema social

La perspectiva sociotécnica de Trist (1981) sostiene que el subsistema técnico y el subsistema social deben optimizarse de forma conjunta; optimizar uno solo tiende a producir disfunciones en el otro. La literatura sobre equipos ágiles —en sus múltiples corrientes— ha venido reforzando esa idea: comunicación efectiva, confianza interpersonal, colaboración interdisciplinar, seguridad psicológica, autoorganización y compromiso colectivo aparecen recurrentemente entre los factores asociados al desempeño sostenido. La literatura revisada sugiere que estos factores aparecen con frecuencia entre los predictores del desempeño de equipo en contextos ágiles.

El compromiso como mediador

Distintos enfoques de la literatura sobre prácticas ágiles apuntan a que las metodologías producen desempeño cuando logran activar compromiso psicológico real en el equipo. Es una formulación coherente con marcos clásicos de comportamiento organizacional: la metodología funciona como vehículo, no como motor. Esa distinción reordena prioridades en la adopción de IA: la herramienta, como la metodología, también es vehículo.

Inteligencia emocional y liderazgo

La literatura sobre habilidades emocionales en equipos de software —y sobre estilos de liderazgo orientados al servicio— apunta a una articulación con prácticas explícitas de liderazgo. La implicación operativa que se mantiene de manera estable: las habilidades emocionales rinden mejor cuando se acompañan de prácticas explícitas de liderazgo, no como rasgo aislado.

Calidad de los requisitos como palanca silenciosa

Hay un cuerpo de literatura que apunta a que la presencia de criterios de aceptación explícitos se asocia con menos errores y menores tiempos de tarea, especialmente en desarrolladores noveles. La inversión en escribir bien los acceptance criteria suele pagarse durante el sprint, particularmente cuando se delega trabajo a IA.

Liderazgo de servicio con mediadores cognitivos

Estudios sobre liderazgo servant en equipos de IT, leídos junto con marcos clásicos de mediación cognitiva (estimulación intelectual, soporte percibido), apoyan la idea de que el estilo de liderazgo no produce desempeño por sí solo: lo produce a través de los mecanismos cognitivos y emocionales que activa.

IA como nueva capa de capacidades tecnológicas

La literatura más reciente sobre integración de IA generativa en pipelines de CI/CD y en flujos de QA sugiere ganancias significativas *cuando* se acompaña de gobernanza de IA responsable, prompt engineering disciplinado y validación humana. Sin esos complementos, la sola presencia del modelo no parece suficiente para producir desempeño sostenido.

Tensiones sistémicas que conviene anticipar

La síntesis identifica seis tensiones que aparecen en cualquier equipo ágil: velocidad versus calidad; autonomía versus alineación; especialización versus multi-skill; ritmo sostenible versus presión de entrega; estandarización versus adaptación contextual; y una tensión emergente especialmente relevante para este ebook: **adopción de IA generativa versus pensamiento crítico humano**. La última no se resuelve eligiendo un lado; se gestiona con prácticas de revisión, validación y aprendizaje.



La productividad de un equipo ágil no se explica solo por su metodología ni por sus herramientas. Se explica por la calidad de las conversaciones que el equipo se permite tener y por la disciplina con la que las repite.

Consideraciones para equipos en Colombia y LATAM

DIFERENCIA: ADOPTAR HERRAMIENTAS VS. DESARROLLAR CAPACIDADES

La literatura disponible sobre productividad ágil e IA proviene en su mayoría de contextos anglosajones, asiáticos o europeos. Las prácticas son útiles pero requieren contextualización para equipos en Colombia y América Latina, donde aparecen condiciones particulares: madurez ágil heterogénea, frecuente uso de células tercerizadas, presupuestos ajustados, presión por entrega rápida, brechas de formación en IA y DevOps, y sectores regulados con exigencias de privacidad fuertes (financiero, salud, seguros, servicios públicos).

La distinción más importante para esta región es la diferencia entre adoptar herramientas y desarrollar capacidades. Comprar licencias es rápido; instalar la disciplina que las hace rentables toma trimestres. Las recomendaciones que siguen apuntan a esa diferencia.

CARACTERÍSTICA DEL CONTEXTO	RIESGO EN ADOPCIÓN DE IA	RECOMENDACIÓN PRÁCTICA
Equipos tercerizados o mixtos (cliente + proveedor)	Uso desigual de herramientas; fuga o asimetría de conocimiento	Política común de IA, biblioteca compartida de prompts y revisión cruzada cliente-proveedor
Madurez ágil heterogénea entre equipos	Adopción rápida que rebasa la capacidad de gobernanza	Pilotos por equipo con criterios claros de avance antes de escalar
Restricciones presupuestales	Selección por costo ignorando coste oculto de fallos	Priorizar dos herramientas dominadas frente a diez subutilizadas
Presión por entrega rápida	Optimización de velocidad que degrada calidad y bienestar	Panel multidimensional desde el primer sprint asistido
Brechas de formación en IA y DevOps	Adopción dispar; riesgo de “expertos accidentales”	Comunidad de práctica interna y rotación de roles en pilotos
Sectores regulados (financiero, salud, seguros, servicios públicos)	Fugas de PII, datos sensibles o IP en prompts externos	Semáforo de datos formal; gateway interno de IA; auditoría regular
Datos en español con jerga local	Salidas del modelo descontextualizadas	Plantillas y prompts en español con glosarios del dominio

CARACTERÍSTICA DEL CONTEXTO	RIESGO EN ADOPCIÓN DE IA	RECOMENDACIÓN PRÁCTICA
Equipos híbridos o distribuidos	Fragmentación de comunicación; pérdida de seguridad psicológica	Protocolos asíncronos explícitos y retros temáticas sobre IA
Dependencia de proveedores externos de modelos	Concentración de riesgo operativo	Acuerdos de servicio con cláusulas de portabilidad y contingencia
Lenguaje importado de literatura internacional	Adopción superficial y mimética	Contextualización local: glosarios, casos regionales, métricas adaptadas

PRECAUCIÓN METODOLÓGICA

Las recomendaciones para Colombia y LATAM deben ajustarse al sector, nivel de regulación, políticas internas de datos y madurez tecnológica de cada organización. No constituyen afirmaciones regionales sin matiz: son hipótesis prácticas que cada equipo debe probar y adaptar a su realidad.

APLICACIÓN PRÁCTICA PARA EQUIPOS EN LATAM

Una primera política de uso de IA puede instalarse con tres elementos: **lista de herramientas autorizadas**, **semáforo de datos del equipo** y **checklist de PR asistido**. Esto no elimina el riesgo regulatorio, pero reduce una parte significativa de los riesgos operativos iniciales y crea una base mínima de control para avanzar con mayor seguridad. El cumplimiento normativo sectorial requiere, además, validación con las áreas legal, de seguridad y de cumplimiento de la organización.

Qué hacen distinto los equipos de alto desempeño

11

Cuando se observa con atención a un equipo que entrega bien, sostenido en el tiempo y bajo presión real, lo que sorprende casi nunca son las herramientas. Lo que sorprende es la disciplina con la que tratan tres recursos invisibles: la atención, el contexto y la confianza.

Tratan la comunicación como capacidad técnica

La comunicación efectiva aparece de forma recurrente en la literatura sobre desempeño de equipos ágiles. En equipos consolidados se traduce en protocolos asíncronos explícitos, reglas claras sobre qué se decide en chat y qué requiere reunión, e intolerancia a la ambigüedad en tickets y pull requests. La consecuencia observable es la reducción del retrabajo derivado de malentendidos.

Ejercen liderazgo de servicio con límites claros

La autonomía sin dirección suele degenerar en fragmentación; la dirección sin autonomía mata la motivación. Los líderes que parecen tener mejor desempeño ofrecen contexto antes que tareas, protegen la capacidad cognitiva del equipo de las urgencias falsas y traducen objetivos en decisiones que el equipo puede tomar localmente.

Invierten en seguridad psicológica como infraestructura

Tener los conflictos correctos sin que cada conversación cueste sprints de reparación emocional. Edmondson (1999) y trabajos posteriores han caracterizado la seguridad psicológica como un habilitador de feedback honesto y aprendizaje. Cuando un desarrollador puede decir “no entendí” o “creo que esto está mal” sin penalización social, el equipo accede al conocimiento real, no al performado.

Tratan el aprendizaje como flujo, no como evento

Pair-rotations, documentación viva, retrospectivas con acuerdos rastreables. Lo distintivo no es entrenar más, sino tener rituales para que el conocimiento de unos pocos se vuelva conocimiento del equipo. La literatura sobre *community smells* sugiere que retros bien hechas pueden detectar silos, lobos solitarios o sobrecarga sobre miembros nuevos antes de que se conviertan en costos visibles.

Mantienen la tensión velocidad-calidad como tensión productiva

No eligen un lado: elevan ambos polos combinando automatización rigurosa (CI/CD, pruebas, observabilidad) con requisitos limpios. La inversión en acceptance criteria explícitos, lejos de ser burocracia, parece ser la infraestructura que permite que la velocidad no degrade calidad.

APLICACIÓN PRÁCTICA · DIAGNÓSTICO BREVE PARA TU EQUIPO (CINCO PREGUNTAS)

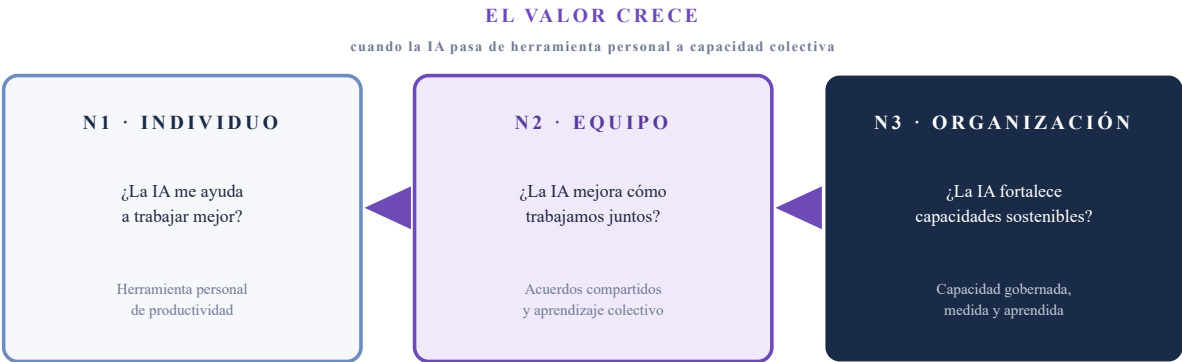
- ¿Tenemos rituales que producen aprendizaje verificable?
- ¿El liderazgo da contexto antes que tareas?
- ¿La infraestructura técnica reduce o crea fricción?
- ¿Las conversaciones difíciles ocurren sin daño?
- ¿Sabemos qué dimensiones de productividad estamos midiendo, más allá de velocidad?

Si una respuesta es débil, esa puede ser la palanca prioritaria —antes de adoptar más IA.

Uso individual, de equipo y organizacional de la IA

12

Una de las confusiones más frecuentes en la adopción de IA es tratar como equivalentes lo que pasa en el escritorio de una persona, lo que pasa en un equipo coordinado y lo que pasa en una organización. Son tres niveles distintos, con preguntas, riesgos e indicadores propios.



N1 · USO INDIVIDUAL

Yo trabajo con IA

Uso de IA por parte de una persona para acelerar tareas propias.

EJEMPLOS

Generar código, explicar errores, crear pruebas, redactar documentación, entender una librería desconocida.

RIESGO PRINCIPAL

Las mejoras individuales no se convierten automáticamente en mejoras del equipo. Una persona puede ahorrar horas mientras el equipo, en conjunto, no avanza más rápido.

INDICADORES ÚTILES

Ahorro de tiempo individual percibido, calidad del output personal, reducción de fricción en tareas repetitivas.

N2 · USO DE EQUIPO

Trabajamos con IA

Uso coordinado de IA mediante acuerdos compartidos, prompts comunes, reglas de revisión y aprendizaje colectivo.

EJEMPLOS

Biblioteca común de prompts, checklist de PR asistido, retrospectivas sobre uso de IA, criterios de aceptación estandarizados, política compartida de datos.

RIESGO PRINCIPAL

Uso desigual entre miembros, formación de silos de conocimiento alrededor de quienes dominan los prompts y dependencia de pocos expertos internos.

INDICADORES ÚTILES

Consistencia de prácticas entre PRs, reducción de retrabajo, calidad de los pull requests, aprendizaje compartido medido en uso real de la biblioteca de prompts.

N3 · USO ORGANIZACIONAL

La organización se vale de IA

IA gestionada como capacidad organizacional: con gobierno, seguridad, observabilidad, costos, estándares y aprendizaje transversal entre equipos.

EJEMPLOS

Política corporativa de IA, lista de herramientas autorizadas, gateway interno de IA, observabilidad LLM, métricas de adopción reportadas a comités, comunidad de práctica entre equipos, gestión integral de riesgos.

RIESGO PRINCIPAL

Burocracia que mata la práctica, adopción cosmética que cumple con políticas pero no cambia el trabajo real, o compra masiva de licencias sin impacto medible.

INDICADORES ÚTILES

Impacto medible en productividad multidimensional agregada, reducción documentada de riesgos, escalamiento controlado entre equipos, capacidad de adaptarse a cambios de proveedor o de modelo.

Tabla comparativa de los tres niveles

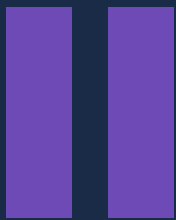
NIVEL	PREGUNTA CLAVE	EJEMPLO DE USO	RIESGO PRINCIPAL	EVIDENCIA DE MADUREZ
Individual	¿La IA me ayuda a trabajar mejor?	Generar tests; explicar errores	Mejoras que no escalan al equipo	Ahorro de tiempo individual con calidad mantenida
Equipo	¿La IA mejora cómo trabajamos juntos?	Biblioteca compartida de prompts; checklist de PR	Uso desigual y silos de conocimiento	Consistencia entre PRs y reducción de retrabajo

NIVEL	PREGUNTA CLAVE	EJEMPLO DE USO	RIESGO PRINCIPAL	EVIDENCIA DE MADUREZ
Organización	¿La IA fortalece capacidades sostenibles?	Política corporativa; gateway de IA; observabilidad	Burocracia o adopción cosmética	Impacto sostenido en productividad multidimensional

IMPLICACIÓN PRÁCTICA

No es razonable saltarse niveles. Una organización que instala gobierno antes de tener prácticas de equipo termina con políticas que nadie aplica. Y un equipo que celebra ganancias individuales sin convertirlas en prácticas compartidas no construye capacidad real.

BLOQUE II



Herramientas y artefactos operativos.

Las decisiones concretas que un equipo puede tomar mañana: qué herramienta usar, en qué condiciones, qué información puede salir del perímetro y qué prompts conviene tener escritos.

Herramientas núcleo de IA para desarrollo de software

13

Las herramientas que siguen son las que se han ganado un lugar en el flujo diario del desarrollo profesional. Cada una resuelve un problema distinto y tiene un perfil de riesgos propio. Mezclarlas sin distinguir su rol es la forma más rápida de gastar recursos sin obtener resultados.

NOTA DE VIGENCIA

Las herramientas descritas corresponden al estado disponible al cierre de esta versión. Dado el ritmo de evolución del ecosistema de IA, se recomienda validar funcionalidades, condiciones de privacidad, costos, integraciones y restricciones contractuales antes de adoptar cualquier solución. Ninguna herramienta es presentada aquí como universalmente superior.

HERRAMIENTA NÚCLEO

ChatGPT (GPT generales)

Cuchillo suizo del desarrollo asistido. Brilla en exploración: redactar primeras versiones de historias, descomponer problemas, traducir documentación, repasar conceptos antes de una decisión arquitectónica. Su mejor uso es como interlocutor crítico fuera del IDE; su valor cae cuando se usa como reemplazo del pensamiento de diseño dentro del flujo de implementación.

HERRAMIENTA NÚCLEO

GitHub Copilot

Autocompletado contextualizado dentro del IDE. Su valor real aparece cuando el repositorio tiene buen tipado, nombres consistentes y convenciones documentadas. Copilot extiende patrones existentes; funciona mejor en bases de código maduras y peor en proyectos con código sucio, donde puede amplificar lo que ya está mal escrito.

HERRAMIENTA NÚCLEO

Cursor

IDE construido alrededor del modelo. Tiene contexto del repositorio entero, permite ediciones multi-archivo coherentes y soporta agentes que ejecutan tareas complejas. Para refactors no triviales, traducción entre stacks o exploración de bases grandes, su rendimiento se distancia del autocompletado clásico. El cuello de botella se desplaza hacia revisión: diffs grandes son riesgosos cuando se mezclan demasiado pronto.

HERRAMIENTA NÚCLEO

Claude Code

Agente orientado a tareas largas y multi-paso. Capacidad distintiva: trabajar con contextos extensos manteniendo coherencia. Donde Copilot completa y Cursor edita, Claude Code planifica y ejecuta: pruebas para módulos completos, diagnóstico de bugs siguiendo trazas, refactors guiados por especificaciones. Requiere cultura previa de planes y especificaciones claras para sostener su valor.

HERRAMIENTA NÚCLEO

OpenAI Codex (orientado a desarrollo)

Ejecución de tareas técnicas en bases de código reales con sandboxes seguros. Su fortaleza está en integración con el ecosistema OpenAI y en disciplina ejecutando pasos verificables. La pregunta crítica antes de adoptarlo es siempre la misma: ¿el equipo tiene mecanismos para validar lo que el agente hizo, o está confiando en su criterio sin contrapeso?

HERRAMIENTA NÚCLEO

Gemini Code Assist

Diferenciado por su integración nativa con Google Cloud y por capacidades en generación de pruebas y comprensión de código. La literatura emergente sobre orquestación de modelos para QA sugiere ganancias importantes cuando se combina más de un modelo y se valida con QA humanos; la ganancia se obtiene por orquestación y validación, no solo por uso del modelo.

CÓMO ELEGIR ENTRE ESTAS HERRAMIENTAS

El criterio no es “*cuál es la mejor*”, es “*cuál encaja con mi flujo y madurez*”. Si necesitas auto-completado en IDE y tu repositorio está limpio, **Copilot**. Si necesitas refactor coherente entre archivos, **Cursor**. Si necesitas ejecutar planes largos con contexto extenso, **Claude Code**. Si necesitas testing automatizado a escala con orquestación, **Gemini Code Assist** con un equipo de QA validando. La sección Árbol de decisión sintetiza este mapeo.

Herramientas de prototipado y validación rápida

14

Hay un grupo de herramientas que cambian la economía del descubrimiento. Su valor no está en el código que generan, sino en el ciclo que habilitan: ver una idea funcionando antes de discutirla en abstracto. Su mayor aporte está en exploración y validación temprana; llevarlas a producción exige revisión técnica, refactorización, pruebas, seguridad y controles de arquitectura.

PROTOTIPADO

v0 (Vercel)

Convierte descripciones en componentes y vistas React. Permite que las conversaciones con stakeholders pasen de “imagina que aquí hay un filtro” a “abre la URL”. Útil para alinear decisiones de producto con prototipos vivos.

PROTOTIPADO

Lovable y Bolt

Generan aplicaciones funcionales completas en minutos —frontend, lógica básica, estado, en algunos casos backend ligero—. Brillan en validación de hipótesis con usuarios reales en horas, no en sprints. Lo que producen no es código de producción; es material de exploración con valor de aprendizaje.

PROTOTIPADO

Replit

Entorno colaborativo en navegador con IA integrada, despliegue inmediato y soporte multilenguaje. Útil para enseñanza, pruebas de concepto y trabajos de investigación interna donde mover archivos entre máquinas es fricción innecesaria.

Conexión con capacidades dinámicas

La literatura sobre capacidades dinámicas (Teece, 2007) describe tres movimientos: sensor oportunidades, capturarlas y reconfigurar la organización para aprovecharlas. Las herramientas de prototipado afectan los dos primeros: bajan el costo de sensor (porque permiten probar ideas antes de comprometer arquitectura) y bajan el costo de capturar (porque convierten ideas validadas en activos digitales en horas). Su impacto sistémico no es la velocidad bruta; es elevar la tasa de experimentación que la organización puede sostener sin colapsar.



El valor del prototipado con IA no está en construir más rápido. Está en descartar más rápido lo que no debería construirse.

Herramientas para QA, seguridad y observabilidad

15

Una vez que la IA forma parte del flujo, aparecen problemas que el ámbito de los prototipos no plantea: cómo se revisa el código sugerido por IA, cómo se detectan vulnerabilidades antes de que entren a producción, cómo se observa el comportamiento de funciones que delegan parte de su lógica a un modelo. El siguiente conjunto de herramientas responde a esas preguntas.

CALIDAD · SEGURIDAD · OBSERVABILIDAD

GitHub con Copilot/Codex en pull requests

La integración de IA en el flujo de pull requests cambia la mecánica de la revisión: sugerencias de redacción, identificación de issues comunes, generación de resúmenes de cambios. La regla operativa: la IA propone, los humanos deciden. La revisión humana sigue siendo el punto de control crítico para cambios arquitectónicos, decisiones de seguridad y modificaciones a contratos públicos.

CALIDAD · SEGURIDAD · OBSERVABILIDAD

Snyk y CodeQL

Snyk se especializa en vulnerabilidades de dependencias y código fuente con remediación asistida por IA. CodeQL, de GitHub, permite escribir consultas semánticas sobre bases de código completas para detectar patrones de bugs y vulnerabilidades. Su valor combinado va más allá de la detección automática: convierten lecciones aprendidas en reglas verificables. Cuando un incidente revela una clase de error, el equipo puede escribir una consulta que la detecte en todo el código futuro.

CALIDAD · SEGURIDAD · OBSERVABILIDAD

LangSmith y Langfuse

Plataformas de observabilidad para aplicaciones que usan LLMs: trazan llamadas, miden latencia, registran prompts y respuestas, evalúan calidad. Una vez que una función del producto depende de un modelo, los bugs dejan de ser determinísticos. Sin observabilidad específica para LLMs, las degradaciones silenciosas son invisibles hasta que un cliente se queja.

CALIDAD · SEGURIDAD · OBSERVABILIDAD

Sentry y Grafana

Observabilidad clásica que sigue siendo el estándar y aumenta su valor con IA. Equipos consolidados construyen dashboards específicos para flujos asistidos: tasa de aceptación de sugerencias, tiempos de respuesta percentil 95, errores correlacionados con tipos de prompt, costos por feature. Sin esos paneles, el equipo no puede saber si la IA está mejorando el producto o degradándolo.

Conexión con capacidades organizacionales

Estas herramientas no son lujos: son la materialización de cuatro capacidades que la literatura identifica como críticas —gobernanza ética y de riesgo, automatización (CI/CD, DevOps), gestión del conocimiento técnico y aprendizaje organizacional. Una herramienta sin la capacidad organizacional que la sostiene es decoración costosa.

Árbol de decisión · qué herramienta usar según el problema

16

Este árbol guía la elección con preguntas simples. La respuesta orienta a una herramienta o a un siguiente filtro.

1 · ¿Necesitas explorar una idea de producto o una interfaz?

- **Sí** Considerar **v0**, **Lovable**, **Bolt** o **Replit**. Son herramientas de exploración; el código generado normalmente requiere refactor antes de producción.
- **No** Ir a la siguiente pregunta.

2 · ¿Necesitas escribir código dentro del IDE?

- **Sí, autocompletado contextual y boilerplate** GitHub Copilot.
- **Sí, con contexto del repo entero o ediciones multi-archivo** Cursor.
- **No** Ir a la siguiente pregunta.

3 · ¿Necesitas refactor multi-archivo o ejecución de tareas largas con plan?

- **Sí, con coherencia en contextos extensos** Claude Code o OpenAI Codex con sandboxes y validación.
- **No** Ir a la siguiente pregunta.

4 · ¿Necesitas asistencia en revisión de PRs o mejora de calidad de código?

- **Sí** GitHub + Copilot/Codex + checklist humano. La IA propone; un humano firma.
- **No** Ir a la siguiente pregunta.

5 · ¿Necesitas reducir riesgos de seguridad y deuda silenciosa?

- **Sí** Snyk (dependencias y código fuente) + CodeQL (consultas semánticas).
- **No** Ir a la siguiente pregunta.

6 · ¿Tu producto usa LLMs en producción?

- **Sí** LangSmith o Langfuse para observabilidad de prompts y trazas + Sentry y Grafana para errores y métricas operativas.
- **No** Ir a la siguiente pregunta.

7 • ¿Necesitas gobernanza organizacional y escalamiento?

→ **Sí** Política de IA, semáforo de datos, matriz de riesgos, diagnóstico puntuable y roadmap 30/60/90. *Las herramientas son la consecuencia, no el punto de partida.*

REGLA PRÁCTICA

Antes de añadir, podar. Si una herramienta no aporta valor en 30 días dentro del equipo, conviene retirarla del flujo.

Matriz comparativa de herramientas

TRES BLOQUES: NÚCLEO · PROTOTIPADO · QA,
SEGURIDAD Y OBSERVABILIDAD

La matriz comparativa se entrega en tres bloques pequeños para que cada uno sea legible. Las columnas usan lenguaje breve; los matices que se pierden en la tabla están en las secciones anteriores.

Bloque 1 · Herramientas núcleo

HERRAMIENTA	MEJOR USO	NO USAR PARA	RIESGO PRINCIPAL	MADUREZ MÍNIMA
ChatGPT	Exploración, redacción, traducción técnica	Reemplazar diseño en implementación	Confiar sin verificación	N1
GitHub Copilot	Boilerplate, fixtures, autocompletado	Bases sucias o sin convenciones	Amplificar deuda preexistente	N2
Cursor	Refactors multi-archivo, repos grandes	Cambios sensibles sin pruebas	Diffs grandes mezclados pronto	N3
Claude Code	Tareas largas con plan explícito	Equipos sin disciplina de especificación	Ejecutar sin validación intermedia	N3
OpenAI Codex	Tareas técnicas verificables en sandbox	Producción crítica sin revisión	Delegar el juicio, no la mecánica	N3
Gemini Code Assist	QA con orquestación; comprensión de código	Integraciones fuera de su ecosistema sin gateway	Atribuir gains al modelo y no a la orquestación	N3

Bloque 2 · Prototipado y validación rápida

HERRAMIENTA	MEJOR USO	NO USAR PARA	RIESGO PRINCIPAL	MADUREZ MÍNIMA
v0	Componentes y vistas React listos para integrar	Producción sin revisión y refactor	Confundir prototipo con código de producción	N1
Lovable	App funcional para validar hipótesis	Promover a producción sin reescritura	Deuda técnica disfrazada de avance	N1
Bolt	Iteración visual con stakeholders	Lógica de negocio crítica	Ilusión de “ya está casi listo”	N1
Replit	PoC, enseñanza, scripts colaborativos	Aplicaciones con cumplimiento estricto	Mezcla de exploración y producción	N1

Bloque 3 · Calidad, seguridad y observabilidad

HERRAMIENTA	MEJOR USO	NO USAR PARA	RIESGO PRINCIPAL	MADUREZ MÍNIMA
GitHub + IA en PRs	Asistencia en revisión, resúmenes	Decisiones arquitectónicas sin revisor humano	Revisión humana superficial por confianza en IA	N3
Snyk	Vulnerabilidades en dependencias y código	Sustituir auditoría humana	Falsa sensación de cobertura	N3
CodeQL	Detección semántica de patrones inseguros	Reemplazar pair review en cambios sensibles	Reglas mal escritas que generan ruido	N4
LangSmith	Trazabilidad y evaluación de prompts/LLMs	Sistemas que no usan LLMs en runtime	Métricas sin acción correctiva	N4
Langfuse	Observabilidad open-source de LLMs	Tráfico bajo donde es overkill	Adopción sin gobernanza de privacidad	N4
Sentry	Errores de aplicación con contexto	Sustituir métricas de negocio	Alertas que se ignoran	N2

HERRAMIENTA	MEJOR USO	NO USAR PARA	RIESGO PRINCIPAL	MADUREZ MÍNIMA
Grafana	Métricas de sistema y dashboards a medida	Reemplazar trazas distribuidas	Dashboards que decoran sin guiar acción	N3

DECISIÓN CRÍTICA · CUÁNDO AÑADIR UNA NUEVA HERRAMIENTA

Verificar tres condiciones: **(a)** resuelve un problema concreto del equipo, no uno teórico; **(b)** el equipo está en el nivel de madurez mínimo requerido (ver Modelo de madurez); **(c)** hay un dueño claro de la integración y de su impacto. Si alguna falla, posponer.

Semáforo de datos en prompts

VERDE · AMARILLO · ROJO

Una de las decisiones más importantes y menos discutidas es qué información sale del perímetro organizacional al usar servicios externos de IA. El semáforo siguiente es una herramienta práctica para tomar esa decisión rápido en el día a día. Cada equipo debe ajustarlo a su política de seguridad, regulación aplicable y proveedor de IA.

TIPO DE INFORMACIÓN	COLOR	¿PERMITIDA EN PROMPTS A SERVICIOS EXTERNOS?	EJEMPLOS TÍPICOS	RECOMENDACIÓN
Información pública	●	Sí	Documentación pública, blogs propios, código open source	Usar libremente
Información interna no sensible	●	Con cuidado	Estructura de módulos, nombres genéricos, convenciones	Anonimizar nombres y contexto innecesario
Código propietario	●	Amarillo / Rojo según contrato	Funciones de negocio, lógica diferencial	Usar gateway interno o instancias dedicadas; si hay duda, no enviar
Datos personales (PII)	●	No	Nombres, correos, números de identidad, direcciones	No enviar a servicios externos sin DPA y anonimización
Secretos técnicos	●	No	Tokens, API keys, credenciales, certificados	Nunca incluir en un prompt; revisar logs
Información contractual	●	No	Cláusulas con clientes, NDAs, condiciones comerciales	Procesar fuera de servicios externos no contratados
Datos de clientes	●	No	Registros, facturación, comportamiento individual	Usar instancias dedicadas o procesamiento local
Datos financieros	●	No	Balances, transacciones, datos de cuentas	Procesamiento interno o entornos certificados

TIPO DE INFORMACIÓN	COLOR	¿PERMITIDA EN PROMPTS A SERVICIOS EXTERNOS?	EJEMPLOS TÍPICOS	RECOMENDACIÓN
Información regulada (salud, financiera, gobierno)	●	No	Historias clínicas, datos protegidos por habeas data, expedientes	Cumplimiento normativo previo y legal aprueba el flujo
Logs de producción	●	Solo redactados	Trazas de errores, requests con datos de usuarios	Filtrar / redactar PII antes de enviar

APLICACIÓN PRÁCTICA

El semáforo se vuelve útil cuando vive donde el equipo trabaja: copia en el README del repositorio, en la wiki interna, en el canal de la herramienta de IA. Y cuando un nuevo desarrollador entra al equipo, el semáforo es uno de los primeros artefactos que recibe.

Playbooks prácticos para usar IA sin perder control

19

Los playbooks que siguen son protocolos operativos. Están escritos para usarse, no para leerse. Cada uno define cuándo aplica, qué pasos seguir y qué riesgos vigilar. Adáptalos a tu contexto; lo importante es que el equipo trabaje con un protocolo escrito y compartido, no con criterios implícitos que cambian por persona.

PLAYBOOK 1

Pull request asistido por IA

Cuándo aplicar. En cualquier PR donde la IA haya generado o modificado código, total o parcialmente. También cuando la IA participa como revisor automático antes del revisor humano.

PASOS

1. Marcar el PR con etiqueta visible (por ejemplo, ai-assisted) y describir en la descripción qué partes fueron generadas por IA y qué prompts se usaron.
2. Generar resumen automático del cambio con la IA y validarlo manualmente: ¿describe lo que realmente hace el código?
3. Ejecutar suite local de pruebas y linters antes de pedir revisión humana.
4. Aplicar el checklist de aceptación; si algún ítem falla, corregir antes de aprobar.
5. Revisor humano lee el código, no solo el resumen. Para cambios sensibles (seguridad, datos, contratos públicos), exigir al menos dos revisores.
6. Si el PR supera 400 líneas modificadas, dividirlo en partes lógicas verificables. *Umbral operativo sugerido, no estándar universal.*

CHECKLIST DE ACEPTACIÓN

- La intención del cambio coincide con el resumen y con los criterios de aceptación de la historia.
- El código sigue las convenciones del repositorio (nombres, estructura, manejo de errores, logging).
- Las pruebas cubren caminos felices, errores y al menos un caso límite.
- No hay credenciales, secretos ni datos personales hardcodeados.
- Si hay dependencias nuevas, están justificadas y revisadas con escaneo de seguridad.
- Las dudas planteadas en revisión están resueltas o registradas como tareas.

RIESGOS

Revisor humano que lee menos atentamente “porque la IA ya lo revisó”. Diffs gigantes generados con poco esfuerzo y revisados con menos.

PLAYBOOK 2

Refactorización asistida por IA

Cuándo aplicar. Antes de iniciar cualquier refactor de un módulo no trivial: deuda técnica acumulada, código sin pruebas, migración entre stacks o frameworks.

PASOS

1. Crear pruebas de caracterización: tests que documentan el comportamiento actual sin juzgarlo. Son la red de seguridad del refactor.
2. Pedir a la IA un plan antes que código. El plan debe describir bloques pequeños y verificables, con criterios de éxito por bloque.
3. Validar el plan con el equipo. Ajustarlo antes de generar líneas de código.
4. Ejecutar el refactor por bloques. Un commit por bloque, un PR pequeño por agrupación lógica.
5. Después de cada bloque, ejecutar la suite y comparar con el comportamiento de las pruebas de caracterización.
6. Documentar el aprendizaje: decisiones, trade-offs, patrones encontrados, riesgos pendientes.

RIESGOS

Refactor que mezcla dos cambios distintos en el mismo PR (riesgo de regresión silenciosa). Pruebas de caracterización débiles que no detectan cambios de comportamiento sutiles.

PLAYBOOK 3

Debugging asistido por IA

Cuándo aplicar. Bugs no triviales: tests flaky, race conditions, fallos en producción con trazas incompletas, comportamientos no determinísticos.

PASOS

1. Reunir contexto: stacktrace completo, logs cercanos, reproducción mínima si existe, hipótesis previas y lo que ya se descartó.
2. Pedir a la IA tres hipótesis ordenadas por probabilidad clínica, no exhaustivas. Para cada hipótesis, exigir el experimento más corto que la confirmaría o descartaría.
3. Ejecutar los experimentos por orden. Documentar cada resultado en un hilo del PR o en una nota anclada al ticket.
4. Una vez identificada la causa raíz, redactar un postmortem corto: qué pasó, por qué, cómo se detectó, cómo se previene en el futuro.
5. Convertir el aprendizaje en regla verificable cuando sea posible (test de regresión, regla CodeQL, alerta en observabilidad).

RIESGOS

Aceptar la primera hipótesis del modelo sin experimento de validación. Resolver el síntoma sin entender la causa raíz, dejando el problema latente.

PLAYBOOK 4

Generación de pruebas con IA

Cuándo aplicar. Al cubrir nuevas funcionalidades, al estabilizar módulos legacy o al cerrar gaps de cobertura útil (no cobertura cosmética).

PASOS

1. Definir el contrato: qué funciones probar, qué entradas son válidas, qué efectos secundarios deben verificarse.
2. Pedir a la IA pruebas estructuradas en tres bloques: camino feliz, errores y casos límite. Exigir mocks tipados y fixtures realistas.
3. Revisar manualmente la cobertura: ¿realmente cubren los riesgos importantes o solo el código que es fácil de probar?
4. Ejecutar pruebas, identificar flakiness y corregir antes de mezclar.
5. Documentar qué tipos de casos no fue posible probar automáticamente y por qué.

RIESGOS

Cobertura alta sin pruebas significativas. Mocks demasiado generosos que ocultan errores reales en integración.

PLAYBOOK 5

Refinamiento de historias de usuario con IA

Cuándo aplicar. En el refinamiento previo al sprint planning, especialmente cuando el backlog tiene tickets en lenguaje vago, ambigüedades de alcance o riesgos no anotados.

PASOS

1. Pasar las historias por la plantilla de prompt para historia de usuario (ver Prompts profesionales reutilizables) para detectar ambigüedad y proponer descomposiciones.
2. Para cada historia, exigir criterios de aceptación medibles. Si la IA no logra escribirlos, eso es señal de que el negocio aún no está listo —no de que la IA falló.
3. Identificar dependencias y riesgos. Dividir historias por riesgo, no solo por tamaño.
4. Validar con Product Owner y equipo en sesión corta. La IA propone, el equipo decide.
5. Versionar la historia: el equipo debe poder ver el antes y el después del refinamiento asistido.

RIESGOS

Aceptar criterios de aceptación generados sin validar si reflejan la realidad del negocio. Sustituir conversación humana con stakeholders por una conversación con el modelo. La IA puede ayudar a estructurar información del negocio, pero no reemplaza la validación con usuarios, clientes y stakeholders.

Prompts profesionales reutilizables

20

Los prompts que siguen están diseñados como plantillas. Cada uno tiene marcadores entre llaves que deben rellenarse con el contexto. La calidad del output depende de la calidad del prompt; tratarlos como activos compartidos del equipo es una de las decisiones con mayor retorno de inversión.

PROMPT 01 · Generación de código con criterios

CODE GENERATION

Necesito una función en `{LENGUAJE}` que `{DESCRIPCIÓN}`.

Restricciones:

- Entradas válidas: `{DETALLE}`
- Casos a rechazar: `{DETALLE}`
- Dependencias permitidas: `{LISTA}`
- Estilo del repositorio: `{CONVENCIONES}`

Devuelve:

1. La función con tipos explícitos.
2. Tres ejemplos de uso.
3. Pruebas en `{FRAMEWORK}`: 3 felices, 3 de error, 1 límite.

No agregues dependencias nuevas sin justificarlas.

PROMPT 02 · Diseño de pruebas

TESTING

Genera pruebas en `{FRAMEWORK}` para la función `{NOMBRE}`.

Estructura los tests en tres describe:

- "happy path" con al menos `{N}` escenarios.
- "error paths" con al menos `{N}` escenarios, incluyendo `{LISTA}`.
- "casos limite" con al menos `{N}` escenarios.

Usa fixtures realistas y mocks tipados.

No expliques el código; entrega los tests listos para ejecutar.

PROMPT 03 · Decisión arquitectónica

ARCHITECTURE

Estoy decidiendo entre {OPCION_A} y {OPCION_B} para {PROBLEMA}.

Restricciones del contexto:

- Carga esperada: {DATO}
- Restricciones operativas: {DATO}
- Habilidades del equipo: {DATO}
- Costo aceptable: {DATO}

Lista 3 trade-offs decisivos para cada opción, con su impacto en costo, complejidad operativa y velocidad de operaciones críticas.

Cierra con: la pregunta que aún no he respondido y que cambiaría la decisión.

PROMPT 04 · Diagnóstico de bug

DEBUGGING

Tengo un bug en {MODULO}. Detalles:

- Síntoma: {DETALLE}
- Frecuencia: {DETALLE}
- Stacktrace: {PEGAR}
- Lo que ya descarte: {LISTA}
- Mis sospechas: {LISTA}

Dame las 3 causas más probables ordenadas por probabilidad clínica. Para cada una, indica el experimento de 5 minutos que la confirma o descarta. No expliques teoría; quiero pasos accionables.

PROMPT 05 · Code review asistido

REVIEW

Revisa este diff como un revisor estricto. Identifica:

1. Riesgos de seguridad (inyección, manejo de secretos, autenticación).
2. Errores funcionales y casos límite no contemplados.
3. Problemas de mantenibilidad (nombres, complejidad, duplicación).
4. Inconsistencias con las convenciones: {LINK_O_PEGAR}

Marca los hallazgos por severidad (alta/media/baja) y sugiere una acción concreta por cada uno. No reescribas el código; indica dónde y por qué.

Diff:

{PEGAR_DIFF}

PROMPT 06 · Historia de usuario

PRODUCT

Convierte esta idea en una historia de usuario formato:

"Como {ROL}, quiero {ACCION}, para {VALOR}".

Detecta ambigüedad y plantea preguntas si el contexto es insuficiente.

Genera:

1. Historia principal.
2. 3-5 criterios de aceptación medibles.
3. Riesgos conocidos y supuestos.
4. Posibles divisiones por riesgo (no por tamaño).

Idea inicial:

{TEXTO}

PROMPT 07 · Criterios de aceptación

PRODUCT

Para esta historia, escribe criterios de aceptación en formato Given/When/Then.

Reglas:

- Cada criterio debe ser verificable automáticamente o por una persona.
- Cubre escenarios felices, errores y al menos un caso límite.
- Identifica supuestos no validados.

Historia: {TEXTO}

PROMPT 08 · Refactorización (plan, no código)

REFACTOR

No quiero código todavía. Dame un plan de refactor para {MODULO}, en bloques pequeños y verificables.

Cada bloque debe incluir:

- Objetivo del bloque.
- Pruebas de caracterización requeridas.
- Riesgo si el bloque falla.
- Criterio para considerarlo terminado.

Estado actual del módulo: {DESCRIPCION}

Constraints: {LISTA}

PROMPT 09 · Documentación técnica viva

DOCS

Genera documentación para {COMPONENTE} con la siguiente estructura:

1. Qué hace y qué no hace.
2. Cómo encaja en la arquitectura.
3. Dependencias internas y externas.
4. Errores conocidos y supuestos.
5. Cómo extenderlo de forma segura.
6. Ejemplos cortos de uso correcto e incorrecto.

Tono: directo, profesional, sin marketing.

Nivel: ingeniero que se incorpora al equipo.

Código de referencia: {PEGAR}

PROMPT 10 · Política de uso de IA en el equipo

GOVERNANCE

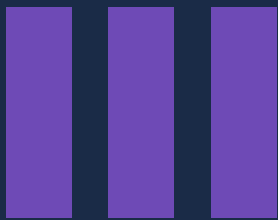
Redacta una primera versión de política de uso de IA para un equipo de desarrollo, con estas secciones:

1. Alcance (qué tareas pueden delegarse, cuáles no).
2. Datos: qué información puede salir a servicios externos y cuál no.
3. Revisión humana: cuándo es obligatoria.
4. Métricas: qué se medirá y dónde se reportará.
5. Excepciones: cómo se solicitan y aprueban.

Contexto del equipo: {DETALLE}

Marco regulatorio aplicable: {DETALLE}

BLOQUE III



Prácticas, métricas y riesgos.

Cómo cambia el balance de las ceremonias ágiles cuando entra la IA, qué medir para evitar la falsa productividad y qué riesgos vigilar a horizontes de meses, no de sprints.

IA en Scrum, Kanban, DevOps y producto

21

Los marcos ágiles fueron diseñados antes de que existiera IA generativa. La pregunta legítima es si siguen siendo apropiados. La respuesta corta es sí, con ajustes. Lo que cambia con la IA no es la lógica del marco —ciclos cortos, inspección y adaptación, equipos auto-organizados— sino el balance interno de las ceremonias y el contenido de algunos roles.

Sprint planning

En equipos que adoptan IA con disciplina, el sprint planning tiende a volverse más exigente, no más rápido. La velocidad de implementación deja de ser el cuello de botella y lo reemplaza la calidad de las historias. Se invierte más tiempo en escribir acceptance criteria que justifiquen lo que se va a generar, porque un ticket vago se convierte en código vago a la velocidad del modelo.

Daily stand-up

El daily cambia poco en estructura y mucho en contenido. Aparecen preguntas nuevas: ¿qué bloqueos tuviste con el agente?, ¿qué sugerencias rechazaste y por qué?, ¿hay alguna funcionalidad donde la IA esté generando deuda? Estas preguntas convierten el daily en mecanismo de aprendizaje colectivo, no solo en reporte.

Retrospectivas

Las retros pasan a discutir el sistema sociotécnico ampliado: prompt patterns que funcionan, errores recurrentes del modelo, decisiones donde la IA llevó al equipo en una dirección equivocada. Si solo dos personas dominan los prompts, se crea un silo invisible. La retro es el lugar donde estos efectos se ven antes de cristalizarse.

Kanban y flujo continuo

En Kanban la IA suele bajar el WIP necesario por desarrollador. El riesgo se traslada a las fases de revisión y verificación, que necesitan reforzarse: PRs grandes asistidos por IA pueden saturar la columna de revisión y romper el flujo. Una regla operativa razonable es ajustar los límites de WIP y la capacidad de revisión cuando se introduce IA.

DevOps

La literatura emergente sobre integración de GenAI en pipelines de CI/CD sugiere que las ganancias en calidad del sistema se asocian a la presencia de gobernanza de IA responsable. Sin gobernanza, los pipelines tienden a volverse cajas negras difíciles de auditar; con ella, la IA es particularmente efectiva en generación y mantenimiento de tests, reducción de tiempos de creación de pipelines y asistencia en remediación de vulnerabilidades.

Gestión de producto

Para Product Owners, la IA cambia tres cosas: la velocidad de generación de hipótesis, la calidad inicial del backlog refinado y la posibilidad de prototipar variantes de funcionalidad antes de discutirlos con stakeholders. La IA puede ayudar a estructurar información del negocio, pero no reemplaza la validación con usuarios, clientes y stakeholders. La conversación con personas reales sigue siendo la fuente primaria del producto.

Cómo evolucionan los roles

SCRUM MASTER

Detecta cuándo la IA está degradando el aprendizaje colectivo y reintroduce prácticas de circulación de criterio.

PRODUCT OWNER

Dedica más tiempo a validar criterios y menos a redactar borradores.

DESARROLLADOR

El porcentaje de tiempo dedicado a escribir baja; el dedicado a leer, validar y diseñar sube.

TECH LEAD

Se vuelve custodio del sistema sociotécnico, no solo del técnico: políticas, prompts compartidos, revisión de calidad.

Mini ejemplos por rol

22

Una práctica concreta, accionable y aplicable mañana, redactada para cada rol del equipo.

ROL · DESARROLLADOR

Antes de aceptar una sugerencia de Copilot para una función crítica, leerla en voz alta. Si no se puede explicar línea por línea por qué hace lo que hace, no se acepta. La IA genera el primer borrador; el desarrollador firma como autor.

ROL · ARQUITECTO

Usar el Prompt 03 (decisión arquitectónica) para listar trade-offs de tres opciones técnicas, no para que el modelo elija. La salida útil es la pregunta no respondida que cambiaría la decisión.

ROL · TECH LEAD

Definir tamaño máximo de PR (por ejemplo, 400 LoC modificados como umbral operativo sugerido) y convertirlo en regla en CI. Hacerlo aplicable a todos por igual; nadie pasa diffs gigantes asistidos por IA sin dividir.

ROL · SCRUM MASTER

En cada cuarta retro, ejecutar una sesión temática sobre IA: quién la usa, quién no, qué prompts circulan. Detectar silos en formación.

ROL · PRODUCT OWNER

Antes de pedir a la IA que escriba acceptance criteria, hablar con un stakeholder. La IA estructura lo que el PO ya entendió; no entiende por sí sola lo que el negocio aún no ha decidido.

ROL · GERENTE DE PROYECTO

Reportar siempre cuatro métricas en el comité: una de velocidad, una de calidad, una de bienestar y una de valor. Si el comité solo mira la primera, abrir conversación sobre por qué.

ROL · GERENTE DE TECNOLOGÍA

Antes de comprar licencias para todo el equipo, ejecutar un piloto de 60-90 días en un solo equipo con métricas multidimensionales. La compra masiva sin diagnóstico suele ser una fuente importante de gasto improductivo.

Productividad · qué medir y qué no medir

23

La productividad en equipos ágiles es multidimensional. Cualquier discusión sobre IA y productividad que ignore esa multidimensionalidad termina optimizando una variable a costa de las demás. La síntesis describe cuatro dimensiones interdependientes —velocidad, calidad, bienestar y valor— como una decisión aplicada de operacionalización, no como una taxonomía universal definitiva.

DIMENSIÓN 1 · VELOCIDAD

Cuán rápido entrega valor el equipo

Métricas útiles. Lead time, cycle time, throughput, tiempo de onboarding, MTTR de bugs.

Riesgo de medirla sola. Empuja a recortar revisión y QA. La IA puede mejorarla cuando acelera tareas mecánicas; puede deteriorarla cuando se acumulan diffs grandes mal revisados o cuando se cierra el síntoma sin causa raíz.

DIMENSIÓN 2 · CALIDAD

Cuán bien hecho está lo que se entrega

Métricas útiles. Defect escape rate, bugs post-release, rework, cobertura útil de pruebas (no nominal), complejidad accidental.

Riesgo de medirla sola. Penaliza releases frecuentes. La IA puede mejorarla con pruebas más amplias y revisión inicial; puede deteriorarla si replica patrones inseguros del corpus o si la cobertura se vuelve métrica vanidosa.

DIMENSIÓN 3 · BIENESTAR

Cuán sostenible es el ritmo del equipo

Métricas útiles. Carga cognitiva percibida (encuesta corta), interrupciones por hora, percepción de autonomía, fatiga por herramientas, satisfacción del desarrollador (DEX).

Riesgo de medirla sola. Indicadores blandos fáciles de descartar en presupuestos. La IA puede mejorar el bienestar reduciendo tareas tediosas; puede deteriorarlo si se acumulan más tareas en el tiempo liberado o si se introducen muchas herramientas a la vez.

DIMENSIÓN 4 · VALOR

Cuánto valor real entrega el producto

Métricas útiles. Uso real de funcionalidades, satisfacción de usuario cualificada, adopción, reducción de fricción del negocio, impacto percibido por stakeholders.

Riesgo de medirla sola. Vanity metrics que no informan decisiones. La IA puede mejorarlo con validación temprana y personalización; puede deteriorarlo si datos sensibles se filtran o si se introducen sesgos no auditados.

APLICACIÓN PRÁCTICA

Mide al menos una métrica de cada dimensión. Un panel responsable contiene mínimo cuatro métricas. Si el equipo solo mide velocidad, la IA va a “mejorar” un número mientras erosiona los otros tres.

Panel mínimo de productividad

24

Este panel mínimo es operativo: cuatro indicadores, uno por dimensión, con responsable y señal de alerta. Adáptalo al contexto del equipo.

V Tarjeta 1 · Velocidad

Métrica mínima. Lead time (de commit a producción).

FRECUENCIA ·
SEMANAL

RESP · TECH LEAD /
GP

Señal de alerta. Lead time baja pero defect escape rate sube.

Acción recomendada. Revisar calidad de PRs asistidos; aplicar Playbook 1.

Q Tarjeta 2 · Calidad

Métrica mínima. Defect escape rate (defectos detectados en producción / total).

FRECUENCIA ·
QUINCENAL

RESP · TECH LEAD /
QA

Señal de alerta. Crecimiento sostenido durante dos sprints.

Acción recomendada. Revisar cobertura útil; introducir o reforzar Snyk/CodeQL.

B Tarjeta 3 · Bienestar

Métrica mínima. Carga cognitiva percibida (encuesta de 1-5 al cierre de sprint).

FRECUENCIA · POR
SPRINT

RESP · SCRUM
MASTER

Señal de alerta. Caída de un punto o más entre sprints consecutivos.

Acción recomendada. Retro temática sobre IA y herramientas; revisar WIP y notificaciones.

\$ Tarjeta 4 · Valor

Métrica mínima. Uso real de la funcionalidad reciente (porcentaje de usuarios objetivo que la usaron al menos una vez en 14 días).

FRECUENCIA · POR
RELEASE

RESP · PRODUCT
OWNER

Señal de alerta. Features que se entregan pero no se adoptan.

Acción recomendada. Sesión de validación con usuarios; revisar refinamiento.

Costos visibles y costos ocultos de adoptar IA

COSTO TOTAL = LICENCIAS + EL TRABAJO QUE LAS HACE PRODUCIR VALOR

La adopción de IA tiene costos visibles y costos ocultos. Las licencias suelen ser el costo más fácil de ver, pero los costos relevantes aparecen en integración, gobernanza, seguridad, revisión, capacitación, observabilidad y deuda técnica si la adopción se hace sin control. Esta sección está pensada para gerentes, líderes técnicos y responsables de tecnología que deben justificar la inversión y, sobre todo, evitar que la inversión se vuelva costo sin retorno.

Idea central

El costo total de adopción de IA no se calcula sumando licencias. Se calcula como:

licencias + integración + gobierno + seguridad + capacitación + observabilidad + revisión humana + deuda técnica evitada / no evitada

Cuando una organización invierte en lo primero y se olvida de lo segundo, el resultado típico es gasto recurrente sin impacto medible.

DECISIÓN	COSTO VISIBLE	COSTO OCULTO	RIESGO SI NO SE GESTIONA	RECOMENDACIÓN
Comprar licencias para todo el equipo de una sola vez	Suscripción mensual	Baja adopción real, falta de reglas, soporte interno, formación	GASTO RECURRENTE SIN IMPACTO MEDIBLE	Iniciar con piloto pequeño y métricas multidimensionales
Usar herramientas de prototipado (v0, Lovable, Bolt)	Bajo costo inicial	Refactor posterior, seguridad, decisiones de arquitectura no tomadas	PROTOTIPO FRÁGIL PROMOVIDO A PRODUCCIÓN	Limitar su uso a validación temprana y descarte rápido
Usar agentes sobre repositorios grandes	Suscripción del agente	Tiempo de configuración, contexto, validación humana de cambios	DIFFS GRANDES QUE PASAN REVISIÓN SUPERFICIAL	Disciplina de plan-antes-de-código y división de PRs

DECISIÓN	COSTO VISIBLE	COSTO OCULTO	RIESGO SI NO SE GESTIONA	RECOMENDACIÓN
Integrar IA en pull requests	Plugin o suscripción	Tiempo de revisión humana real, reglas de aceptación, formación de revisores	APROBACIONES AUTOMÁTICAS SIN CRITERIO	Mantener revisión humana firme; checklist obligatorio
No invertir en observabilidad de LLMs	Cero	Deudas operacionales, debug ciego, incidentes que tardan en detectarse	DEGRADACIÓN SILENCIOSA DEL PRODUCTO	Introducir observabilidad mínima desde el primer flujo en producción
No definir política de datos	Cero	Riesgos regulatorios, fugas de PII o IP, conversaciones difíciles a posteriori	INCIDENTES QUE COMPROMETEN LA OPERACIÓN	Política mínima desde el día uno, aunque sea breve
Capacitar solo a una parte del equipo	Costo bajo de formación	Silos de conocimiento, dependencia de pocos expertos	APRENDIZAJE NO SE COMPARTE NI SE SOSTIENE	Rotación, mentoring interno, comunidad de práctica
Adoptar muchas herramientas al mismo tiempo	Múltiples suscripciones	Fatiga por herramientas, contexto fragmentado, integración compleja	PRODUCTIVIDAD REAL CAE MIENTRAS LA NOMINAL SUBE	Dominar dos herramientas antes de añadir una tercera

REGLA PRÁCTICA

Antes de aprobar una compra de IA, exigir un plan de uso con tres elementos: **quién la operará, qué métricas se reportarán y cuándo se evaluará el impacto**. Sin esos tres, lo más probable es que la licencia no produzca el retorno esperado.

Riesgos, señales tempranas y controles

26

La adopción de IA introduce riesgos específicos. La matriz siguiente organiza los principales, sus señales tempranas y los controles recomendados. La columna “Responsable” es una sugerencia: depende del tamaño y estructura del equipo.

RIESGO 01**Dependencia cognitiva**

SEÑAL TEMPRANA	el equipo evita tareas sin asistencia
SI NO SE CONTROLA	atrofia técnica progresiva
CONTROL	espacios deliberados de trabajo sin IA
RESPONSABLE	Tech Lead
HERRAMIENTA	sesiones de pair programming sin IA; retros temáticas

RIESGO 02**Deuda técnica silenciosa**

SEÑAL TEMPRANA	diffs grandes con baja revisión
SI NO SE CONTROLA	código frágil meses después
CONTROL	checklist de PR + diffs pequeños obligatorios
RESPONSABLE	Toda la cadena de revisión
HERRAMIENTA	Playbook 1; métricas de complejidad

RIESGO 03**Fuga de información sensible**

SEÑAL TEMPRANA	prompts con datos personales o secretos
SI NO SE CONTROLA	incidente regulatorio o reputacional
CONTROL	semáforo de datos + gateway interno
RESPONSABLE	Seguridad / Tech Lead
HERRAMIENTA	DLP, gateways de IA, redacción de prompts

RIESGO 04**Código inseguro generado**

SEÑAL TEMPRANA	patrones inseguros recurrentes
SI NO SE CONTROLA	vulnerabilidades en producción
CONTROL	Snyk + CodeQL + revisión humana
RESPONSABLE	Seguridad / Tech Lead
HERRAMIENTA	escaneos en CI con bloqueo de severidades altas

RIESGO 05**Falsa productividad**

SEÑAL TEMPRANA	velocity sube y calidad baja
SI NO SE CONTROLA	pérdida de confianza del negocio
CONTROL	panel multidimensional de métricas
RESPONSABLE	Gerente de proyecto
HERRAMIENTA	sección Productividad; dashboards balanceados

RIESGO 06**Deterioro del pensamiento crítico**

SEÑAL TEMPRANA	aceptación pasiva de sugerencias
SI NO SE CONTROLA	erosión del juicio del equipo
CONTROL	revisión activa, prompts que pidan trade-offs
RESPONSABLE	Tech Lead / Scrum Master
HERRAMIENTA	Prompts 03 y 05; revisión de criterios

RIESGO 07

Sesgo en soluciones generadas

SEÑAL TEMPRANA	patrones repetidos sin contexto
SI NO SE CONTROLA	decisiones miopes
CONTROL	revisión por pares y diversidad cognitiva
RESPONSABLE	Equipo
HERRAMIENTA	pair programming; retros con alternativas exploradas

RIESGO 08

PRs demasiado grandes

SEÑAL TEMPRANA	diffs > 400 LoC con frecuencia
SI NO SE CONTROLA	revisión superficial
CONTROL	política de tamaño máximo y división
RESPONSABLE	Tech Lead
HERRAMIENTA	reglas en CI; comentarios automáticos

RIESGO 09

Falta de observabilidad LLM

SEÑAL TEMPRANA	quejas de usuarios no reproducibles
SI NO SE CONTROLA	degradación silenciosa del producto
CONTROL	trazas y evals automáticos
RESPONSABLE	DevOps / SRE
HERRAMIENTA	LangSmith, Langfuse

RIESGO 10

Prototipos en producción

SEÑAL TEMPRANA	v0/Lovable promovidos sin reescritura
SI NO SE CONTROLA	fragilidad y deuda crítica
CONTROL	política de promoción + criterios técnicos
RESPONSABLE	Arquitecto
HERRAMIENTA	sección Herramientas de prototipado; gates de calidad

ERROR COMÚN

Confundir “sin incidentes” con “bajo riesgo”. Muchos riesgos de IA se materializan en horizontes de meses (deuda técnica, atrofia, fuga acumulada). Equipos consolidados revisan la matriz de riesgos al menos cada trimestre, incluso cuando todo parece tranquilo.

BLOQUE IV

IV

Síntesis · framework, madurez y plan.

Un mapa integrador del sistema sociotécnico, un modelo de madurez en cinco niveles, un diagnóstico puntuable y un plan 30/60/90 para inclinar la trayectoria del equipo.

Framework integrador · capacidades, IA y productividad

27

El framework integra capacidades organizacionales, tecnológicas y sociotécnicas identificadas en la literatura revisada con prácticas emergentes de adopción de IA. Sirve como herramienta diagnóstica y como guía de inversión: dónde está hoy el equipo y dónde necesita estar para que la IA produzca valor sostenido.

Tres planos, una arquitectura

01

Plano habilitador

Cultura organizacional, liderazgo, infraestructura tecnológica y procesos ágiles. Crea las condiciones del trabajo.

02

Plano mediador

Red sociorrelacional (comunicación, confianza, seguridad psicológica) y capacidad psicoafectiva y cognitiva (motivación, compromiso, pensamiento crítico). Conecta condiciones con resultados.

03

Plano emergente

Productividad multidimensional: velocidad, calidad, bienestar, valor. No es la suma de los planos anteriores; emerge cuando interactúan.

+

Plano transversal · moduladores contextuales

Modalidad distribuida o presencial, diversidad cognitiva, riesgos externos, gobernanza ética, madurez sectorial. Alteran la fuerza con la que las capacidades centrales producen efectos.

Tabla maestra del framework

PLANO	CAPACIDAD	INDICADOR PRÁCTICO	CÓMO AMPLIFICA LA IA BIEN USADA
Habilitador	Cultura ágil y aprendizaje	Retros con acuerdos rastreables	Documentación viva, circulación de conocimiento
Habilitador	Liderazgo servant con límites	OKRs claros, soporte percibido	Más tiempo para mentoring; datos para decisiones
Habilitador	Capacidad tecnológica (CI/CD, DevOps)	Pipelines verdes, observabilidad	Copilot/Cursor/Codex en flujo; Snyk/CodeQL en gates

PLANO	CAPACIDAD	INDICADOR PRÁCTICO	CÓMO AMPLIFICA LA IA BIEN USADA
Habilitador	Procesos ágiles bien aplicados	AC explícitos, cadencia estable	Refinement asistido; borradores de stories
Mediador	Red sociorrelacional	Frecuencia y honestidad del feedback	Reduce ambigüedad; traduce entre roles
Mediador	Capacidad psicoafectiva y cognitiva	Pensamiento crítico, decisiones colectivas	Genera opciones; el equipo decide
Emergente	Productividad multidimensional	Velocidad + calidad + bienestar + valor	Eleva los cuatro si los anteriores están sanos
Transversal	Moduladores contextuales	Modalidad, diversidad, gobernanza, sector	Determinan la dosis y los protocolos viables

Cómo encaja la IA en este framework

La IA no es una capa nueva: puede leerse como un amplificador que se conecta a capacidades existentes. En el plano habilitador, eleva la capacidad tecnológica instrumentada. En el plano mediador, puede tanto amplificar como erosionar la capacidad psicoafectiva y cognitiva: la amplifica cuando libera tiempo para diseño y revisión; la erosiona cuando reemplaza pensamiento crítico por aceptación pasiva. En el plano emergente, su impacto depende del estado de los planos anteriores.

PRECAUCIÓN

La IA puede amplificar capacidades, pero también puede amplificar fragilidades si el sistema sociotécnico está débil. La dirección de la amplificación depende del estado del sistema, no del modelo elegido.

DECISIÓN CRÍTICA · DIAGNÓSTICO ANTES QUE HERRAMIENTA

Si una capacidad del plano habilitador o mediador está débil, fortalecerla suele rendir más que adoptar otra herramienta. Las intervenciones más rentables son las que actúan sobre nodos centrales del sistema (comunicación, seguridad psicológica, compromiso); las periféricas no mueven la dinámica de fondo.

Modelo de madurez de adopción de IA

CINCO NIVELES ACUMULATIVOS · NO SON ETAPAS OBLIGATORIAS

Cinco niveles que sirven para autoubicarse, definir el siguiente paso y comunicar el estado actual. Los niveles no son etapas obligatorias —algunos equipos nunca llegan al 5 y eso puede ser apropiado— pero son acumulativos: el siguiente nivel solo es estable si el anterior está consolidado.

<div>N1</div> <div>Uso individual informal</div> <div>Síntesis. Cada persona usa IA como quiere; sin reglas.</div> <div>Señales. Adopción dispar; resultados anecdóticos.</div> <div>Antipatrones. Reportar gains personales como gains del equipo.</div> <div>Siguiente experimento. Acordar dos herramientas comunes y reglas mínimas.</div> <div>Evidencia. Política mínima escrita y aplicada.</div>	<div>N2</div> <div>Uso asistido básico</div> <div>Síntesis. El equipo usa IA en tareas repetitivas.</div> <div>Señales. Copilot en flujo; sin gobernanza fuerte.</div> <div>Antipatrones. Optimizar velocity sin medir calidad.</div> <div>Siguiente experimento. Introducir checklist de PR y biblioteca de prompts.</div> <div>Evidencia. Métrica multidimensional en el panel.</div>	<div>N3</div> <div>Uso gobernado</div> <div>Síntesis. Reglas, revisión humana y políticas mínimas.</div> <div>Señales. Política de datos; biblioteca de prompts.</div> <div>Antipatrones. Cumplimiento solo en papel.</div> <div>Siguiente experimento. Integrar IA al pipeline (PRs, tests, seguridad).</div> <div>Evidencia. Pipelines asistidos con gates de calidad.</div>	<div>N4</div> <div>Uso integrado al SDLC</div> <div>Síntesis. IA en PRs, testing, CI/CD, seguridad y observabilidad.</div> <div>Señales. LangSmith/Langfuse; CodeQL.</div> <div>Antipatrones. Dependencia operativa frágil de un solo proveedor.</div> <div>Siguiente experimento. Convertir aprendizajes en reglas verificables y comunidad de práctica.</div> <div>Evidencia. Reglas CodeQL desde incidentes; comunidad activa.</div>	<div>N5</div> <div>Uso sistémico adaptativo</div> <div>Síntesis. La IA fortalece aprendizaje y capacidades dinámicas.</div> <div>Señales. Revisión trimestral; mejora continua.</div> <div>Antipatrones. Sobrecarga organizacional; pérdida de foco.</div> <div>Siguiente experimento. Podar herramientas que no aporten valor sostenido.</div> <div>Evidencia. Reducción de herramientas; valor sostenido medido.</div>
--	---	--	---	--

Detalle por nivel

N1 · USO INDIVIDUAL INFORMAL

Prácticas. Cada persona prueba ChatGPT, Copilot o similares según preferencia.

Antipatrones. Reportar gains personales como gains de equipo; suponer que el equipo "ya usa IA".

Siguiente experimento. Acordar dos herramientas para todo el equipo y reglas básicas (semáforo de datos mínimo, checklist breve de PR).

Evidencia. Política mínima escrita y comprobada en al menos cinco PRs.

N2 · USO ASISTIDO BÁSICO

Prácticas. Boilerplate, generación de tests sencillos, drafts de stories.

Antipatrones. Velocity sube y calidad baja sin que nadie lo note; bibliotecas de prompts personales no compartidas.

Siguiente experimento. Política básica de datos, biblioteca compartida de prompts, checklist de PR aplicado y panel multidimensional con cuatro métricas.

Evidencia. Panel mínimo de productividad operando durante al menos un sprint.

N3 · USO GOBERNADO

Prácticas. Playbooks aplicados; métricas multidimensionales; revisión humana obligatoria en cambios sensibles.

Antipatrones. "Teatro de gobernanza" (políticas que existen pero no se aplican); revisión humana superficial.

Siguiente experimento. Integrar IA al pipeline (PRs, tests, seguridad), introducir observabilidad LLM si aplica.

Evidencia. PRs asistidos con etiqueta visible y checklist completo en una mayoría sostenida de los casos (umbral operativo sugerido).

N4 · USO INTEGRADO AL SDLC

Prácticas. PR asistido, testing automático con LLM, escaneos de seguridad, dashboards de calidad LLM.

Antipatrones. Dependencia operativa frágil de un solo proveedor; complejidad sin podar.

Siguiente experimento. Convertir aprendizajes en reglas verificables; cultivar comunidad de práctica; instaurar revisión trimestral.

Evidencia. Reglas (CodeQL, alertas, tests) generadas a partir de incidentes documentados.

N5 · USO SISTÉMICO ADAPTATIVO

Prácticas. Capacidad dinámica de adaptación; aprendizaje organizacional sostenido; revisión continua de la matriz de riesgos.

Antipatrones. Sobrecarga organizacional; pérdida de foco en lo esencial.

Siguiente experimento. Mantener la disciplina; podar lo que no aporta valor sostenido.

Evidencia. Reducción documentada de herramientas redundantes manteniendo o mejorando los indicadores de las cuatro dimensiones.

Diagnóstico rápido · ¿qué tan preparado está tu equipo?

29

Instrumento puntuable. Aplicación individual primero, luego sesión grupal corta. Las brechas entre miembros suelen ser más informativas que el promedio: indican falta de visibilidad, formación desigual o políticas no socializadas.

Cómo aplicarlo

APLICACIÓN INDIVIDUAL

Cada miembro del equipo responde por separado, sin discusión previa, los 15 ítems en escala de 1 a 5. Tiempo estimado: 12-15 minutos.

APLICACIÓN GRUPAL

Después de la individual, sesión de 60 minutos para comparar resultados, discutir brechas y acordar las tres acciones prioritarias.

CUÁNDO APLICAR

Antes del piloto de adopción de IA y, nuevamente, al final del primer y tercer mes. La comparación entre mediciones revela el progreso real.

CÓMO INTERPRETAR BRECHAS

Las brechas grandes entre miembros son más informativas que el promedio: indican falta de visibilidad, formación desigual o políticas no socializadas. Discutirlas explícitamente.

Escala

1. No existe.
2. Existe de forma informal.
3. Existe parcialmente.
4. Existe y se usa.
5. Existe, se usa y se mejora continuamente.

Ítems del diagnóstico

#	DIMENSIÓN	ÍTEM	PUNTAJE (1-5)
01	GOBERNANZA	Existe una política escrita sobre qué tareas pueden delegarse a IA y cuáles no	—
02	GOBERNANZA	Hay reglas claras sobre qué información puede o no enviarse a servicios externos	—
03	CALIDAD	Los criterios de aceptación son verificables y se usan al revisar PRs asistidos	—

#	DIMENSIÓN	ÍTEM	PUNTAJE (1-5)
04	CALIDAD	Las pruebas cubren riesgos críticos, no solo cobertura nominal	—
05	SEGURIDAD	Hay escaneos automáticos en CI con bloqueo de severidades altas	—
06	SEGURIDAD	El equipo sabe qué patrones inseguros nunca aceptar, incluso si la IA los sugiere	—
07	USO DE HERRAMIENTAS	El equipo dominó dos herramientas antes de explorar más; no se está saturado	—
08	REVISIÓN HUMANA	Existe checklist explícito para PRs asistidos por IA y se aplica	—
09	MÉTRICAS	Se miden al menos cuatro métricas, una por cada dimensión de productividad	—
10	APRENDIZAJE	Hay biblioteca compartida y mantenida de prompts, con contexto y resultados	—
11	APRENDIZAJE	Las retros incluyen, al menos cada cuatro sprints, una revisión del uso de IA	—
12	CULTURA	Hay seguridad psicológica para decir “rechacé esta sugerencia porque...”	—
13	PENSAMIENTO CRÍTICO	Los desarrolladores leen y entienden lo que aceptan; no hay aceptación pasiva	—
14	OBSERVABILIDAD	Hay observabilidad específica para flujos asistidos por LLM (o un plan claro)	—
15	LIDERAZGO	El liderazgo discute IA en términos de las cuatro dimensiones, no solo velocidad	—

Interpretación de resultados

Rangos sugeridos como umbral operativo, no como estándar universal.

RANGO TOTAL (15-75)	NIVEL DE ADOPCIÓN	PRIORIDAD SUGERIDA
15 – 30	Adopción improvisada	Reglas mínimas de uso; consolidar dos herramientas; política de datos
31 – 45	Adopción inicial	Checklist de PR asistido; biblioteca de prompts; métrica multidimensional

RANGO TOTAL (15-75)	NIVEL DE ADOPCIÓN	PRIORIDAD SUGERIDA
46 – 60	Adopción gobernada	Integrar IA al pipeline; observabilidad para LLMs; cultura de revisión activa
61 – 70	Adopción integrada	Reforzar comunidad de práctica; reglas desde aprendizajes; revisión trimestral
71 – 75	Adopción sistémica	Mantener disciplina; podar herramientas; cultivar capacidad dinámica

SALIDA ESPERADA DEL DIAGNÓSTICO

Tras aplicar el diagnóstico, el equipo debe quedar con tres salidas concretas: **(1)** Top 3 capacidades más débiles —las tres dimensiones con menor puntaje promedio—; **(2)** Top 3 riesgos prioritarios —cruzar las capacidades débiles con la matriz de riesgos para identificar los riesgos más expuestos—; **(3)** Acción prioritaria de los próximos 30 días —una acción concreta, con dueño y fecha, que ataque el riesgo más alto.

Roadmap de adopción 30 / 60 / 90 días

30

Treinta días no transforman un equipo, pero 90 días pueden inclinar su trayectoria. Este roadmap está pensado para equipos pequeños (3-10 personas) que quieren empezar bien y construir capacidad real, no solo adoptar herramientas.

DÍAS 1 – 30

Diagnóstico y piloto

Objetivo. Establecer línea base y validar valor con riesgo controlado.

ACCIONES

- Aplicar el diagnóstico puntuable.
- Elegir dos herramientas como máximo.
- Definir reglas mínimas de uso (semáforo de datos, checklist de PR).
- Lanzar un piloto controlado en una funcionalidad acotada.

ENTREGABLES

- Diagnóstico aplicado.
- Política mínima.
- Métricas base.
- Reporte breve del piloto.

MÉTRICAS A OBSERVAR

Lead time, defect escape rate, satisfacción del equipo (encuesta corta).

CRITERIOS DE ENTRADA

El equipo acepta dedicar entre el 10 y el 20 % de su capacidad al piloto y aceptar pausarlo si los criterios de detención se activan.

DÍAS 31 – 60

Integración y gobierno

Objetivo. Convertir el piloto en práctica del equipo.

ACCIONES

- Crear la biblioteca compartida de prompts.
- Integrar el checklist de PR asistido como parte del flujo.
- Introducir testing asistido en un módulo seleccionado.
- Formalizar la política de datos.
- Realizar una retro temática sobre IA.

ENTREGABLES

- Biblioteca de prompts versionada.
- Checklist de PR aplicado de manera consistente.
- Política de datos firmada.
- Acta de retro con tres acciones accionables.

MÉTRICAS A OBSERVAR

Cycle time, cobertura útil, carga cognitiva, percepción de autonomía.

DÍAS 61 – 90

Escalamiento responsable

Objetivo. Llevar la práctica al pipeline y al producto.

ACCIONES

- Integrar IA con CI/CD (PR, tests, seguridad).
- Introducir observabilidad para LLMs si aplica.
- Crear o sumarse a una comunidad de práctica interna.
- Evaluar impacto y decidir escalamiento.

ENTREGABLES

- Pipeline asistido con gates.
- Observabilidad LLM mínima.
- Plan de escalamiento o cierre con justificación.

MÉTRICAS A OBSERVAR

MTTR de bugs, complejidad accidental, satisfacción del usuario, impacto percibido.

CRITERIOS DE SALIDA

Reporte del piloto entregado; tres aprendizajes documentados; decisión explícita sobre continuar, ajustar o detener.

SEÑALES PARA DETENER EL ESCALAMIENTO

Detener si se cumplen una o más de las siguientes señales y persisten dos sprints consecutivos:

- Crece el defect escape rate (calidad).
- Aumenta la carga cognitiva (bienestar).
- La revisión humana se vuelve superficial (cultura).
- El equipo no puede explicar los cambios aceptados (pensamiento crítico).

SEÑALES PARA AMPLIAR EL PILOTO

Ampliar si se cumplen, en condiciones reproducibles:

- Mejora el lead time sin deteriorar la calidad.
- Baja el rework.
- Aumenta la satisfacción del equipo.
- Las prácticas quedan documentadas y se sostienen sin presión externa.

REGLAS QUE APLICAN A LOS TRES PERIODOS

Antes de añadir, podar. Toda decisión sobre IA se documenta brevemente: qué se decidió, por qué, cómo se medirá. Nadie adopta nuevas prácticas en sprints con compromisos críticos al cliente. Los aprendizajes del piloto deben quedar visibles para la organización al final de los 90 días.

Diez principios para usar IA en equipos de software

31

Estos principios son las reglas no negociables del autor para esta edición. Cada uno describe el principio, el error que evita y la acción concreta que lo materializa.

01 La IA propone; el equipo decide

Error que evita. delegar el juicio en lugar de la mecánica

Acción concreta. todo cambio asistido por IA tiene un humano responsable que firma y entiende lo que firma.

02 Ningún código crítico entra sin revisión humana

Error que evita. asumir que la IA revisó por nosotros

Acción concreta. checklist explícito de PR asistido y, en cambios sensibles, doble revisión humana.

03 Todo prompt repetible debe convertirse en activo compartido

Error que evita. la calidad del output dependa del prompt suelto del día

Acción concreta. biblioteca de prompts del equipo, mantenida con dueño y versión.

04 Si no hay tests, no hay confianza

Error que evita. aceptar refactor o nuevo código sin red de seguridad

Acción concreta. pruebas de caracterización antes de refactor; tests con la PR del cambio.

05 La velocidad solo importa si no destruye calidad

Error que evita. optimizar velocity y degradar calidad sin notarlo

Acción concreta. métricas de las cuatro dimensiones siempre visibles en el panel del equipo.

06 Información sensible no sale del perímetro sin política clara

Error que evita. fugas accidentales de PII, secretos o IP

Acción concreta. política escrita, semáforo de datos, gateway interno cuando aplique.

07 La IA debe reducir fricción, no ocultar problemas organizacionales

Error que evita. usar IA para parchar disfunciones culturales o de liderazgo

Acción concreta. diagnóstico del sistema antes de añadir herramientas.

08 Los diffs grandes asistidos por IA deben dividirse

Error que evita. revisión superficial de cambios masivos

Acción concreta. regla de tamaño máximo (umbral operativo sugerido, por ejemplo 400 LoC) y división por bloques verificables.

09 La adopción se mide con velocidad, calidad, bienestar y valor

Error que evita. celebrar gains de velocidad mientras se pierde en otras dimensiones

Acción concreta. panel multidimensional desde el primer día del piloto.

10 El pensamiento crítico es una capacidad que debe protegerse

Error que evita. que el equipo se vuelva más rápido y menos pensante

Acción concreta. espacios deliberados sin asistencia; prompts que pidan trade-offs y alternativas; cultura de revisión activa.

Quick Start · Tesis y principios

32

Tres páginas tipo flyer compartible. Para WhatsApp, LinkedIn o correo. Para llevar al próximo café técnico.

TESIS CENTRAL

La IA no sustituye capacidades; las amplifica. Su impacto depende del sistema socio-técnico que la recibe. Si el equipo tiene buenas capacidades, la IA tiende a multiplicarlas. Si tiene fallas de cultura, liderazgo, calidad, seguridad o gobernanza, la IA puede amplificar también la fragilidad.

Cinco principios esenciales

1. La IA propone; el equipo decide.
2. Ningún código crítico entra sin revisión humana.
3. Todo prompt repetible se convierte en activo compartido.
4. La adopción se mide con velocidad, calidad, bienestar y valor.
5. El pensamiento crítico debe protegerse explícitamente.

Árbol corto de decisión de herramientas

Explorar idea o interfaz → v0, Lovable, Bolt, Replit.

Escribir código en IDE con autocompletado → GitHub Copilot.

Refactor multi-archivo o repo grande → Cursor, Claude Code, OpenAI Codex.

Revisar PRs y mejorar calidad → GitHub + IA + checklist humano.

Seguridad → Snyk + CodeQL.

LLMs en producción → LangSmith / Langfuse + Sentry + Grafana.

comparte este Quick Start · Página 1 de 3

Quick Start · Controles operativos

32

CHECKLIST DE PR ASISTIDO (VERSIÓN CORTA)

- PR etiquetado como **ai-assisted** con descripción de partes generadas y prompts usados.
- Diff razonable; si excede el umbral interno del equipo (sugerencia: 400 LoC), dividir.
- Tests locales y linters en verde.
- Cobertura: feliz, errores y al menos un caso límite.
- Sin credenciales, secretos ni PII hardcodeados.
- Convenciones del repositorio respetadas.
- Cambios sensibles: doble revisor humano.

SEMÁFORO DE DATOS (VERSIÓN CORTA)

VERDE Información pública: usar libremente.

AMARILLO Información interna no sensible o código propietario: anonimizar; gateway si hay duda.

ROJO PII, secretos, datos contractuales/financieros/regulados, datos de clientes: no enviar a servicios externos sin política y aprobación formal.

Cinco errores críticos

1. Medir solo velocidad.
2. Llevar prototipos a producción sin reescritura.
3. Delegar juicio técnico al modelo.
4. Adoptar demasiadas herramientas a la vez.
5. Ignorar seguridad y observabilidad de LLMs.

comparte este Quick Start · Página 2 de 3

Quick Start · Diagnóstico y plan de 90 días

32

Diagnóstico rápido (cinco preguntas)

1. ¿Tenemos rituales que producen aprendizaje verificable?
2. ¿El liderazgo da contexto antes que tareas?
3. ¿La infraestructura técnica reduce o crea fricción?
4. ¿Las conversaciones difíciles ocurren sin daño?
5. ¿Sabemos qué dimensiones de productividad estamos midiendo más allá de velocidad?

Si una respuesta es débil, esa puede ser la palanca prioritaria —antes de adoptar más IA.

Roadmap simplificado

30 DÍAS

Diagnóstico, dos herramientas, reglas mínimas, piloto controlado, panel multidimensional.

60 DÍAS

Biblioteca de prompts, checklist de PR aplicado, testing asistido, política de datos, retro IA.

90 DÍAS

Pipeline asistido, observabilidad LLM, comunidad de práctica, decisión de escalamiento basada en evidencia.

TRES ACCIONES PARA MAÑANA

1. Acordar las dos herramientas con las que el equipo va a empezar.
2. Pegar el semáforo de datos en el README del repositorio principal.
3. Aplicar el checklist de PR asistido en el próximo PR del equipo.

comparte este Quick Start · Página 3 de 3

Conclusión

33

La diferencia entre un resultado útil y uno costoso parece estar menos en el modelo y más en el sistema sociotécnico que lo recibe.

Tras revisar literatura y leer prácticas emergentes, la conclusión que sostiene este ebook es prudente. La IA no es ni el atajo que prometen sus vendedores ni la amenaza que dramatiza el discurso público. Funciona mejor como un amplificador. La diferencia entre un resultado útil y uno costoso suele estar menos en el modelo y más en el sistema sociotécnico que lo recibe.

La asimetría que la práctica suele ignorar es la siguiente: las herramientas son visibles, comprables y cambiables; las capacidades sociotécnicas son invisibles, lentas y difíciles de imitar. Cualquier estrategia de adopción de IA que invierta primero en lo visible y posponga lo invisible está optimizando para titulares, no para resultados. Los equipos que tiendan a sostener desempeño en la próxima década serán probablemente los que entiendan esa asimetría y construyan en sentido contrario: primero el sistema, después la herramienta.

La adopción de IA no es un proyecto que se termina; es una capacidad dinámica que se cultiva. Las herramientas cambiarán, los modelos mejorarán, surgirán riesgos nuevos. Lo que sostendrá la productividad no es estar al día con cada lanzamiento, sino haber instalado en el equipo los rituales y los criterios que permiten asimilar lo nuevo sin perder coherencia.

La adopción madura ocurre cuando la IA deja de ser una ayuda individual aislada y se convierte en una práctica de equipo gobernada, medible y aprendida organizacionalmente. Ese tránsito —de lo individual a lo colectivo y luego a lo organizacional— es donde la IA pasa de truco productivo a capacidad sostenible.



La pregunta correcta no es cuánta IA debería usar mi equipo. Es: ¿qué capacidades están haciendo a mi equipo capaz de aprovecharla, y cuáles las están bloqueando?

BLOQUE V



Anexos metodológicos y prácticos.

Detalles del protocolo de revisión, estado del corpus, plantillas reutilizables, trazabilidad de afirmaciones y bibliografía organizada. Para auditoría, replicación y trabajo posterior.

Anexo metodológico

34

Resumen ejecutivo del protocolo de revisión sistemática de literatura conducido por el autor.

1 · Propósito de la revisión

Identificar, clasificar y estructurar las capacidades organizacionales, tecnológicas y sociotécnicas asociadas a la productividad de equipos de desarrollo de software en contextos ágiles, y describir su articulación con variables de desempeño relevantes para la construcción posterior de un modelo sistémico.

2 · Pregunta orientadora

Cómo se define y operacionaliza la productividad o eficiencia de entrega de equipos de desarrollo de software en contextos ágiles; qué capacidades se reportan en la literatura y cómo pueden clasificarse en organizacionales, tecnológicas y sociotécnicas; qué relaciones se describen entre esas capacidades y los resultados de productividad; y qué elementos contextuales se reportan como condicionantes para su transferencia al contexto colombiano.

3 · Bases de datos consultadas

INDEXADAS MULTIDISCIPLINARES

- Scopus
- Web of Science

INGENIERÍA DE SOFTWARE Y CS

- IEEE Xplore
- ACM Digital Library
- ScienceDirect
- SpringerLink
- Wiley Online Library
- Emerald Insight
- MDPI

REGIONALES

- SciELO
- Redalyc

4 · Ecuaciones booleanas (resumen)

Las búsquedas se ejecutaron sobre los campos *title-abstract-keyword* donde la base lo permite. Se usaron tres ecuaciones base, ajustables a la sintaxis de cada base.

Ecuación 1 · Productividad en equipos ágiles

TITLE-ABS-KEY

```
(agile OR "agile software development" OR Scrum OR Kanban OR
"extreme programming" OR DevOps OR "continuous delivery")
AND (team OR "software development team" OR
"software engineering team" OR "delivery team")
AND (productivity OR performance OR efficiency OR
"team performance" OR throughput OR velocity OR
"lead time" OR "cycle time")
```

Ecuación 2 · Capacidades asociadas a productividad

TITLE-ABS-KEY

```
(agile OR "agile software development" OR Scrum OR Kanban OR XP OR DevOps)
AND (team OR "software development team")
AND (productivity OR performance OR "delivery performance")
AND (capability OR "organizational capability" OR
"technological capability" OR "IT capability" OR
sociotechnical OR "socio-technical" OR competence OR
enabler OR determinant OR factor)
```

Ecuación 3 · Contexto colombiano / latinoamericano

TITLE-ABS-KEY

```
[(Ecuación 1) OR (Ecuación 2)]
AND (Colombia OR Colombian OR "Latin America")
```

5 · Ventana temporal e idiomas

Ventana: enero de 2001 a marzo de 2026. El punto de partida se justifica por la consolidación contemporánea del desarrollo ágil a partir del Manifiesto Ágil (Beck et al., 2001). **Idiomas:** inglés y español.

6 · Criterios de inclusión

- Estudios sobre equipos de desarrollo o ingeniería de software como unidad de análisis principal o claramente inferible.
- Contexto ágil o prácticas equivalentes operacionalizables (Scrum, Kanban, XP, DevOps, entrega continua).
- Análisis explícito de productividad, desempeño, eficiencia de entrega o conceptos análogos a nivel de equipo.
- Vínculo con capacidades organizacionales, tecnológicas, sociotécnicas o factores interpretables bajo ese esquema.
- Estudios empíricos, conceptuales, marcos o taxonomías con suficiente estructura para extracción sistemática.
- Publicaciones arbitradas: artículos de revista, conferencias con revisión editorial, capítulos académicos.

- Publicación entre 2001 y marzo de 2026, en inglés o español.

7 · Criterios de exclusión

- Productividad en dominios no relacionados con desarrollo de software sin conexión explícita con equipos de software.
- Foco exclusivo en desempeño individual sin resultados atribuibles al equipo.
- Sin referencia operativa a contexto ágil.
- Blogs, editoriales, opiniones, presentaciones sin texto completo, foros o documentos sin trazabilidad metodológica.
- Sin texto completo disponible tras intentos razonables de acceso.

8 · Proceso de depuración documental

Lógica PRISMA: identificación, cribado por título y resumen, evaluación de elegibilidad a texto completo, e inclusión. Se aplicó **doble cribado independiente sobre una muestra representativa** (entre el 10 % y el 20 % del total) revisada por el director del trabajo de grado, con discusión de discrepancias para alcanzar consenso.

9 · Herramientas de apoyo

GESTIÓN BIBLIOGRÁFICA

Zotero o Mendeley para deduplicación y registro.

CRIBADO DOCUMENTAL

Rayyan (Ouzzani et al., 2016) para apoyo al cribado y organización documental.

BITÁCORA METODOLÓGICA

Repositorio controlado por versiones o cuaderno de investigación con registro de decisiones.

10 · Evaluación de calidad metodológica

Marco base: **Mixed Methods Appraisal Tool (MMAT) 2018** (Hong et al., 2018), complementado con criterios transversales de claridad conceptual y descripción contextual. Escala: 1 (cumple claramente), 0,5 (cumple parcialmente o información insuficiente), 0 (no cumple o no reporta).

11 · Criterios de calidad

- Claridad del objetivo o pregunta.
- Coherencia entre objetivo y diseño metodológico.
- Descripción del contexto y la unidad de análisis.
- Definición u operacionalización de productividad.
- Definición suficiente de capacidades o factores.
- Descripción de fuente de datos y recolección.
- Coherencia del método de análisis.
- Declaración de limitaciones.
- Elementos que permitan valorar transferibilidad.

12 · Categorías de extracción

ID interno; referencia bibliográfica completa; tipo de publicación; país o región del contexto; dominio organizacional; marco o práctica ágil reportada; unidad de análisis; definición conceptual de productividad; indicadores o medidas utilizadas; capacidades organizacionales, tecnológicas y sociotécnicas identificadas; tipo de relación entre capacidades y desempeño; enfoque y diseño metodológico; tamaño de muestra cuando aplique; resultado de calidad metodológica; observaciones sobre transferibilidad al contexto colombiano.

13 · Estrategia de síntesis

Análisis temático con procedimiento explícito (Braun & Clarke, 2006): familiarización, codificación, agrupación, revisión y definición de temas. Mapeo descriptivo del corpus por distribución temporal, tipos de estudio, contextos y marcos ágiles. Esquema híbrido: deducción inicial de macro-categorías (organizacionales, tecnológicas, sociotécnicas) e inducción controlada de subcategorías emergentes; los factores ambiguos se marcan como transversales o mixtos.

14 · Relación entre hallazgos de la RSL y recomendaciones del ebook

Las recomendaciones operativas del ebook se construyen sobre dos fuentes: **marcos teóricos clásicos** con respaldo empírico amplio (Trist, 1981; Edmondson, 1999; Teece, 2007) usados como guía conceptual, y **prácticas emergentes documentadas** por equipos profesionales y proveedores de herramientas.

15 · Limitaciones metodológicas declaradas

- El proceso fue conducido principalmente por el investigador principal con doble cribado parcial. El principal riesgo residual es la subjetividad del investigador en selección, evaluación y codificación.
- La estrategia de búsqueda no fue sometida a revisión por pares externa. El control de rigor se sostiene en documentación íntegra de ecuaciones, filtros y fechas; prueba piloto y refinamiento iterativo de términos; comparación de sensibilidad entre ecuaciones; y bitácora de ajustes.
- La aplicación empírica de las prácticas propuestas en equipos reales corresponde a una fase posterior del trabajo de grado.

Corpus consolidado

35

Razones de exclusión documentadas en el cribado

De los registros excluidos se documentaron las siguientes razones (subtotal de razones registradas: 1 898 sobre el universo cribado):

MOTIVO	REGISTROS	% DEL SUBTOTAL
Fuera del alcance temático	1 370	72,2 %
No aporta al objetivo de la revisión	490	25,8 %
Textos completos no encontrados	15	0,8 %
Limitaciones en lectura del texto completo	15	0,8 %
Metadatos insuficientes / sin resumen	7	0,4 %
Capacidades tecnológicas con impacto indirecto en productividad	1	0,1 %

Anexos prácticos · plantillas reutilizables

36

Las siguientes plantillas están listas para copiar y adaptar. Recomendación: trasladarlas al wiki interno o al sistema de gestión del equipo y mantenerlas con dueño y versión, como cualquier otro activo del producto.

A1 · Plantilla de prompt para historia de usuario

PRODUCT

Convierte esta idea en una historia de usuario en formato:

"Como {ROL}, quiero {ACCION}, para {VALOR}".

1. Si la idea es ambigua, hazme preguntas antes de escribir.
2. Genera la historia principal y posibles divisiones por riesgo.
3. Lista supuestos no validados.
4. No incluyas tareas técnicas; eso lo decide el equipo.

Idea inicial:

{TEXTO}

A2 · Plantilla de prompt para criterios de aceptación

PRODUCT

Para esta historia, escribe criterios de aceptación en formato Given/When/Then.

Reglas:

- Cada criterio debe ser verificable.
- Cubre escenario feliz, errores y al menos un caso límite.
- Identifica supuestos no validados.

Historia: {TEXTO}

A3 · Plantilla de prompt para refactorización (plan)

REFACTOR

No quiero código. Plan de refactor para `{MODULO}`, en bloques pequeños y verificables.

Cada bloque debe incluir:

- Objetivo.
- Pruebas de caracterización requeridas.
- Riesgo si el bloque falla.
- Criterio de terminado.

Estado actual: `{DESCRIPCION}`

Constraints: `{LISTA}`

A4 · Plantilla de prompt para generación de pruebas

TESTING

Genera pruebas en `{FRAMEWORK}` para `{NOMBRE}`.

Estructura: happy path (≥ 3), error paths (≥ 3), casos limite (≥ 1).

Usa fixtures realistas y mocks tipados. Devuelve solo el código.

A5 · Plantilla de prompt para code review

REVIEW

Revisa este diff como un revisor estricto.

Identifica: seguridad, errores funcionales, mantenibilidad, inconsistencias con convenciones.

Marca por severidad (alta/media/baja) y sugiere acción concreta.

No reescribas; señala dónde y por qué.

Diff:

`{PEGAR_DIFF}`

A6 · Plantilla de prompt para decisión arquitectónica

ARCHITECTURE

Decisión: `{OPCION_A}` vs `{OPCION_B}` para `{PROBLEMA}`.

Lista 3 trade-offs decisivos para cada opción y su impacto en costo, complejidad y velocidad.

Cierra con la pregunta no respondida que cambiaría la decisión.

Contexto: `{DETALLE}`

A7 · Plantilla de prompt para diagnóstico de bug

DEBUGGING

Bug en {MODULO}.

- Síntoma: {DETALLE}
- Frecuencia: {DETALLE}
- Stacktrace: {PEGAR}
- Descartado: {LISTA}
- Sospechas: {LISTA}

Dame las 3 causas más probables ordenadas por probabilidad clínica.

Para cada una, el experimento de 5 minutos que la confirma o descarta.

A8 · Plantilla de política básica de uso de IA

GOVERNANCE

POLITICA DE USO DE IA - EQUIPO {NOMBRE}

Versión {x.y} - Vigente desde {FECHA}

1. Alcance.

- Herramientas autorizadas: {LISTA}.
- Tareas que pueden delegarse: {LISTA}.
- Tareas que no pueden delegarse: {LISTA}.

2. Datos.

- Datos prohibidos en prompts a servicios externos:
{PII, secretos, IP de cliente}
- Datos permitidos: {LISTA}.
- Si en duda, no enviar.

3. Revisión humana.

- Obligatoria para: cambios de seguridad, datos, contratos públicos, infraestructura crítica.
- Checklist aplicado: ver Anexo A10.

4. Métricas.

- El equipo reporta mensualmente: lead time, defect escape rate, carga cognitiva, satisfacción de usuario (o equivalentes).

5. Excepciones.

- Solicitar a {ROL} con justificación; aprobación documentada.

6. Revisión.

- Esta política se revisa cada 90 días o ante incidente.

AGENDA - RETRO IA - {SPRINT/PERIODO}

1. Datos del periodo (10 min)
 - Tareas asistidas por IA: _____ %
 - PRs con etiqueta "ai-assisted": _____ %
 - Defect escape rate: _____
 - Carga cognitiva (encuesta corta): _____
2. Qué funcionó (15 min)
 - Prompts y playbooks que produjeron valor.
 - Decisiones donde la IA aceleró sin degradar calidad.
3. Qué no funcionó (15 min)
 - Sugerencias rechazadas y por qué.
 - Errores recurrentes del modelo.
 - Donde la IA empujó a la dirección equivocada.
4. Riesgos detectados (10 min)
 - Señales de la matriz de riesgos.
5. Acuerdos accionables (10 min)
 - Máximo 3 acciones; con dueño y fecha.

CHECKLIST - PR ASISTIDO POR IA

- [] PR etiquetado como "ai-assisted".
- [] Descripción indica qué partes fueron generadas por IA y prompts usados.
- [] Tamaño del diff razonable (umbral sugerido ≤ 400 LoC); si excede, dividir.
- [] Tests locales y linters en verde.
- [] La intención del cambio coincide con resumen y AC.
- [] No hay credenciales, secretos ni PII hardcodeados.
- [] Dependencias nuevas justificadas y revisadas (Snyk/CodeQL).
- [] Las pruebas cubren happy + errores + ≥ 1 caso límite.
- [] Convenciones del repositorio respetadas.
- [] Si hay cambios sensibles (seguridad/datos/contratos): doble revisor humano.
- [] Dudas planteadas en revisión: resueltas o registradas como tareas.

Anexo de trazabilidad de evidencia

DE LA AFIRMACIÓN A LA FUENTE · CÓMO LEER
CADA AFIRMACIÓN IMPORTANTE DEL EBOOK

Este anexo conecta las afirmaciones más importantes del ebook con sus fuentes y con su tipo de evidencia. La columna “Nivel de confianza” expresa cuán sólida es la evidencia para la afirmación tal como se usa en este ebook —no la calidad metodológica del estudio en general.

AFIRMACIÓN USADA EN EL EBOOK	FUENTE	TIPO DE EVIDENCIA	NIVEL DE CONFIANZA PARA EL USO DADO	USO DENTRO DEL EBOOK	OBSERVACIÓN METODOLÓGICA
Tesis: la IA puede actuar como amplificador del sistema sociotécnico	Síntesis del autor a partir del corpus + marcos clásicos	Interpretación aplicada	MEDIO (CONSISTENTE CON LA PERSPECTIVA SOCIOTÉCNICA DE TRIST)	Eje conceptual del ebook	Marcado como interpretación; no causalidad demostrada
La productividad ágil es multidimensional (velocidad, calidad, bienestar, valor)	Síntesis temática del autor	Decisión aplicada de operacionalización	ALTO COMO MARCO; MEDIO COMO CATEGORIZACIÓN EXACTA	Estructura el panel mínimo y la matriz de métricas	La operacionalización por métrica concreta es decisión del autor
Centralidad del subsistema social en equipos ágiles	Marco sociotécnico clásico (Trist, 1981) + síntesis temática	Marco teórico clásico + lectura del corpus	ALTO COMO MARCO CONCEPTUAL	Justifica el énfasis en cultura y seguridad psicológica	Convergencia temática observable; no prevalencia universal
Compromiso de proyecto como mediador entre prácticas ágiles y desempeño	Hipótesis emergente de la literatura revisada	Hipótesis aplicada	MEDIO (CONSISTENTE CON LITERATURA)	Sustenta el principio “metodología es vehículo, no motor”	
Calidad de criterios de aceptación como palanca silenciosa	Hipótesis emergente de la literatura revisada	Hipótesis aplicada	MEDIO	Justifica la inversión en AC y su importancia con IA	

AFIRMACIÓN USADA EN EL EBOOK	FUENTE	TIPO DE EVIDENCIA	NIVEL DE CONFIANZA PARA EL USO DADO	USO DENTRO DEL EBOOK	OBSERVACIÓN METODOLÓGICA
Liderazgo servant con mediadores cognitivos	Hipótesis emergente de la literatura revisada	Hipótesis aplicada	MEDIO	Sustenta la sección sobre liderazgo y la guía por rol	
IA en CI/CD asociada a mejoras en calidad bajo gobernanza responsable	Síntesis del autor a partir de literatura emergente	Hipótesis aplicada	MEDIO-ALTO	Sustenta la sección DevOps	El efecto está mediado por gobernanza; sin ella, los gains no se materializan
La adopción de IA puede erosionar el pensamiento crítico si se delega sin validar	Síntesis del autor a partir del corpus (tensión emergente)	Interpretación derivada de tensiones sistémicas	MEDIO (CONSISTENTE CON LA LITERATURA, NO DEMOSTRADA CON UN ÚNICO ESTUDIO)	Sustenta principio 10 y matriz de riesgos	Marcado como interpretación; no causalidad demostrada
La seguridad psicológica habilita feedback honesto y aprendizaje	Edmondson (1999) y trabajos posteriores	Marco teórico clásico con respaldo empírico amplio	ALTO COMO MARCO	Sustenta la sección “alto desempeño” y los principios	Marco bien establecido; aplicable como guía conceptual
Las capacidades dinámicas explican la adaptación organizacional a tecnologías nuevas	Teece (2007)	Marco teórico	ALTO COMO MARCO CONCEPTUAL	Estructura el modelo de madurez y la sección de capacidades dinámicas	Marco bien establecido; aplicable como guía conceptual
Cifras del flujo PRISMA al cierre de esta edición	Matriz PRISMA propia del autor para esta versión	Reporte directo del proceso de revisión	ALTO DENTRO DEL PROCESO DOCUMENTADO	Sustenta la nota metodológica, el diagrama PRISMA y el corpus	Cifras correspondientes al estado al cierre de la edición

NOTA

El nivel de confianza se refiere a la afirmación tal como se usa en este ebook. Marcos teóricos clásicos pueden tener nivel “Alto como marco” aunque no apliquen como evidencia empírica directa para una práctica concreta.

Referencias

38

Referencias usadas o citadas en el ebook, organizadas por bloque temático y formato APA 7.

Marcos metodológicos y conceptuales

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for Agile Software Development*. <https://agilemanifesto.org/>

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77-101. <https://doi.org/10.1191/1478088706qp063oa>

Edmondson, A. C. (1999). Psychological safety and learning behavior in work teams. *Administrative Science Quarterly*, 44(2), 350-383. <https://doi.org/10.2307/2666999>

Hong, Q. N., Pluye, P., Fàbregues, S., Bartlett, G., Boardman, F., Cargo, M., Dagenais, P., Gagnon, M.-P., Griffiths, F., Nicolau, B., O'Cathain, A., Rousseau, M.-C., & Vedel, I. (2018). *Mixed Methods Appraisal Tool (MMAT), version 2018*. Canadian Intellectual Property Office.

Kitchenham, B. (2004). *Procedures for performing systematic reviews* (TR/SE-0401). Keele University.

Ouzzani, M., Hammady, H., Fedorowicz, Z., & Elmagarmid, A. (2016). Rayyan—A web and mobile app for systematic reviews. *Systematic Reviews*, 5, Article 210. <https://doi.org/10.1186/s13643-016-0384-4>

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., ... Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, 372, n71. <https://doi.org/10.1136/bmj.n71>

Page, M. J., Shamseer, L., Tricco, A. C., et al. (2015). Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement. *Systematic Reviews*, 4, Article 1. <https://doi.org/10.1186/2046-4053-4-1>

Rethlefsen, M. L., Kirtley, S., Waffenschmidt, S., Ayala, A. P., Moher, D., Page, M. J., & Koffel, J. B. (2021). PRISMA-S: An extension to the PRISMA statement for reporting literature searches in systematic reviews. *Systematic Reviews*, 10, Article 39. <https://doi.org/10.1186/s13643-020-01542-z>

Teece, D. J. (2007). Explicating dynamic capabilities: The nature and microfoundations of (sustainable) enterprise performance. *Strategic Management Journal*, 28(13), 1319-1350. <https://doi.org/10.1002/smj.640>

Tranfield, D., Denyer, D., & Smart, P. (2003). Towards a methodology for developing evidence-informed management knowledge by means of systematic review. *British Journal of Management*, 14(3), 207-222. <https://doi.org/10.1111/1467-8551.00375>

Trist, E. L. (1981). *The evolution of socio-technical systems*. Ontario Quality of Working Life Centre.

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. En *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering* (pp. 1-10). <https://doi.org/10.1145/2601248.2601268>

Documentación técnica de herramientas mencionadas

Las herramientas referenciadas en el ebook (ChatGPT, GitHub Copilot, Cursor, Claude Code, OpenAI Codex, Gemini Code Assist, v0, Lovable, Bolt, Replit, Snyk, CodeQL, LangSmith, Langfuse, Sentry, Grafana) se citan a partir de su documentación pública vigente al cierre de esta edición. Por la velocidad de evolución del ecosistema, se recomienda al lector consultar la documentación actualizada de cada proveedor antes de tomar decisiones de adopción.