

Guía

IA aplicada al trabajo ágil

Desarrollo de los temas del State of the art



Actualizado a octubre de 2026

Scrum Manager® · Skill Arena

Sobre este documento

Este documento desarrolla en profundidad los temas del mapa de referencia (State of the art) para incorporar la inteligencia artificial al trabajo ágil con criterio, sea cual sea el rol de quien la usa, que está disponible en: [Skill Arena](#).

Úsalo como material de consulta para mantener y contrastar tu práctica con el conocimiento profesional actual.

Se complementa con la plataforma de entrenamiento y evaluación en Skill Arena. En el área [IA aplicada al trabajo ágil](#) puedes realizar pruebas de entrenamiento para contrastar y mejorar tu nivel de conocimiento y si lo deseas también puedes obtener un diploma que acredita curricularmente la solvencia y vanguardia profesional en esta área.



Estado del conocimiento

El conocimiento sobre IA aplicada al trabajo ágil evoluciona de forma extremadamente rápida: herramientas, estándares y obligaciones legales cambian en cuestión de meses. Los fundamentos (dar al modelo el contexto adecuado, pedir con precisión, verificar lo que devuelve y proteger la información) se han asentado, y lo que se mueve es el trabajo con agentes y las reglas con que cada equipo gobierna su uso. En los distintos apartados del documento, las etiquetas (ESTABLECIDO, EN CONSOLIDACIÓN, EMERGENTE) ayudan a identificar la madurez de cada concepto:

ESTABLECIDO consenso asentado; conocimiento que se da por necesario.

EN CONSOLIDACIÓN gana adopción con rapidez; aún no universal pero ya relevante.

EMERGENTE frontera reciente; alta relevancia y alta volatilidad.

Índice

Capítulos del desarrollo, en el orden del State of the art.

Bloque A · Fundamentos de interacción con la IA

1. Ingeniería de contexto (*context engineering*)
2. Prompting estructurado y reutilizable
3. Verificación y fiabilidad de las respuestas
4. Frontera de confianza: instrucciones frente a datos, privacidad y secretos
5. Política de uso de IA del equipo

Bloque B · Trabajar con agentes de IA

6. Del chat al agente: delegación de trabajo verificable
7. Diseño de supervisión humana (*human-in-the-loop*)
8. Estándares del ecosistema agéntico

Bloque C · Aplicación al ciclo de trabajo ágil

9. IA en el flujo de producto y *backlog*
10. IA en facilitación y dinámicas de equipo
11. IA en calidad y entrega (perfiles de desarrollo)

BLOQUE A · FUNDAMENTOS DE INTERACCIÓN CON LA IA

Capítulo 1. Ingeniería de contexto (*context engineering*) · EN CONSOLIDACIÓN

Diseñar deliberadamente qué información recibe el modelo en cada momento, en qué orden y qué se deja fuera, más allá de cómo se redacta la instrucción puntual.

Introducción

Durante los primeros años de trabajo con modelos de lenguaje, el esfuerzo profesional se concentró en redactar la instrucción perfecta, el *prompt* que daba la mejor respuesta. Esa habilidad sigue siendo necesaria, pero ha dejado de ser el centro. A medida que el trabajo con IA ha pasado de la consulta de un solo turno a las tareas sostenidas y al uso de agentes, ha ganado peso una competencia más amplia que la engloba: la ingeniería de contexto.

En este capítulo veremos qué es el contexto de un modelo, por qué darle más información no equivale a darle mejor información, qué estrategias se usan para gestionarlo y cómo se traduce todo ello en hábitos de trabajo para cualquier profesional ágil, programe o no. Es un tema en consolidación: el marco conceptual está asentado, pero las técnicas y las funciones de los productos cambian cada pocos meses.



Figura 1. La ventana de contexto: todo lo que entra compite por la atención del modelo, y gestionar el contexto es decidir qué se guarda fuera, qué se trae, qué se resume y qué se separa.

1.1 De la instrucción al contexto

Anthropic define la ingeniería de contexto como el conjunto de estrategias para seleccionar y mantener la información que el modelo tiene delante mientras trabaja, incluida toda la que llega fuera de las instrucciones. Ese resto es mucho: las indicaciones del sistema, el historial de la conversación, los documentos recuperados, los resultados de las herramientas que el modelo ha usado y los ejemplos que se le han dado.

La analogía más extendida es la de la memoria de trabajo. La ventana de contexto es un espacio finito donde solo cabe una cantidad limitada de información, y lo que se incluye compite por la atención del modelo. Anthropic habla de un presupuesto de

atención que se gasta con cada elemento añadido, de modo que gestionar el contexto es administrar un recurso escaso.

Idea central. El *prompting* es una parte de la ingeniería de contexto, no una disciplina rival. Para una consulta puntual basta una buena petición; la gestión del contexto se vuelve imprescindible cuando el trabajo es sostenido, cuando intervienen agentes o cuando la información relevante excede lo que cabe con comodidad en una sola petición.

1.2 Por qué más contexto no es mejor contexto

La intuición sugiere que, si el modelo se nutre de información, conviene darle toda la disponible. La evidencia apunta en la dirección contraria, y por dos vías.

- **La degradación con la longitud.** En julio de 2025, Kelly Hong, Anton Troynikov y Jeff Huber publicaron en Chroma un estudio sobre 18 modelos que mostraba que su rendimiento se vuelve menos fiable a medida que crece la entrada, incluso en tareas tan simples como recuperar o copiar un texto. Lo llamaron degradación del contexto (*context rot*).
- **La posición importa.** Nelson F. Liu y otros autores (2023) describieron el fenómeno conocido como perdido en el medio (*lost in the middle*): los modelos aprovechan mejor la información situada al principio o al final de un contexto largo que la enterrada en el centro.

La consecuencia práctica es un principio de diseño. El objetivo pasa a ser curar la información: incluir lo justo, ordenarlo bien y retirar lo que ha dejado de ser útil. Añadir contexto irrelevante tiene un coste, porque resta atención a lo que importa.

1.3 Cuatro estrategias para gestionar el contexto

Lance Martin propuso en junio de 2025 una taxonomía de cuatro estrategias que LangChain difundió semanas después y que se ha extendido como vocabulario común. Anthropic describe las mismas ideas con otros nombres (compactación, notas estructuradas, subagentes y recuperación justo a tiempo), así que conviene leerla como un mapa útil más que como un estándar.

1. **Escribir.** Guardar información fuera de la ventana para que esté disponible sin ocupar espacio. El patrón típico es el cuaderno de notas: el agente registra decisiones, hallazgos o avances en un almacén externo y los recupera cuando los necesita.
2. **Seleccionar.** Traer a la ventana solo lo relevante para el paso actual, mediante búsqueda o recuperación dirigida. Anthropic lo llama recuperación justo a tiempo, porque el agente guarda referencias ligeras y carga los datos cuando le hacen falta.
3. **Comprimir.** Retener solo lo necesario, resumiendo conversaciones largas o condensando los resultados extensos de una herramienta. La compactación, que

resume el historial y reinicia la ventana con ese resumen, ya es una función de los productos.

4. **Aislar.** Separar contextos distintos para que no interfieran. La forma más avanzada es la de varios subagentes especializados, cada uno con su contexto limpio, que devuelven resúmenes condensados a un agente coordinador.

Las cuatro estrategias se combinan. Un sistema que trabaja durante horas suele usarlas a la vez: los subagentes aíslan, cada uno recupera lo que necesita, mantiene sus notas y resume lo largo antes de devolverlo.

1.4 La ingeniería de contexto en el trabajo ágil

Para un profesional ágil, esta competencia es una forma de plantear cualquier trabajo con IA, más que una cuestión de infraestructura. Antes de pedirle algo a un modelo, la pregunta útil es qué necesita saber para hacerlo bien y qué le sobra.

Hábitos que se derivan de las cuatro estrategias

- **Aportar el contexto mínimo suficiente**, sin volcar documentos enteros por si acaso.
- **Situar lo decisivo en posiciones de alta atención:** los documentos largos al principio y la petición al final.
- **Separar las conversaciones** distintas en lugar de arrastrar un hilo interminable que mezcla temas.
- **Apoyarse en notas externas** en las tareas largas, como un resumen de decisiones que se aporta al empezar cada sesión.

Las mismas ideas reaparecen a escala de equipo. Los ficheros de instrucciones que un agente lee al empezar y las habilidades que carga solo cuando las necesita (capítulo 8) son ingeniería de contexto compartida. La *skill* de Harness Engineering de *Scrum* Manager desarrolla ese diseño del entorno de trabajo de los agentes para los equipos de desarrollo.

Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Distinguir la ingeniería de contexto del prompting y explicar por qué la primera engloba al segundo.
- Reconocer cuándo una tarea requiere gestionar el contexto (trabajo sostenido, agentes, información extensa) y cuándo basta con una buena petición.
- Justificar con evidencia por qué añadir contexto puede empeorar el resultado.
- Identificar qué estrategia (escribir, seleccionar, comprimir o aislar) resuelve un problema de contexto concreto.
- Aplicar el contexto mínimo suficiente y el orden de la información a una petición real de su trabajo.

Errores y antipatrones frecuentes

- **Confundir gestionar el contexto con alargar la petición.** Una instrucción más larga suele añadir ruido. La gestión consiste en curar, no en acumular.
- **Volcar toda la información disponible.** Es el error más común y el más costoso, porque diluye la atención del modelo y degrada la fiabilidad.
- **Enterrar lo decisivo en el centro.** Una instrucción crítica en mitad de un contexto largo tiene menos probabilidades de atenderse que al principio o al final.
- **Arrastrar un único hilo para todo.** Mezclar tareas distintas en una conversación interminable provoca interferencias. Abrir una conversación nueva es a menudo la solución más simple.

Para profundizar

- [Anthropic — Effective context engineering for AI agents](#)
- [Chroma — Context rot](#)
- [Lance Martin — Context engineering for agents](#)
- [Liu y otros — Lost in the middle](#)
- [Scrum Manager BoK — Context engineering](#)

BLOQUE A · FUNDAMENTOS DE INTERACCIÓN CON LA IA

Capítulo 2. Prompting estructurado y reutilizable · ESTABLECIDO

Especificar una petición con los componentes que hacen su resultado predecible y verificable, y convertir las peticiones que funcionan en activos del equipo.

Introducción

El prompting es la competencia base de todo trabajo con modelos de lenguaje: formular una petición de modo que el resultado sea útil, predecible y verificable. Aunque la ingeniería de contexto la engloba (capítulo 1), sigue siendo el cimiento, porque ninguna estrategia de contexto rescata un resultado mal pedido.

En este capítulo veremos los componentes de una petición eficaz, qué instrucciones aportan consistencia y cuáles solo matizan, cómo han cambiado las recomendaciones con los modelos recientes y cómo se convierte una petición útil en una plantilla del equipo.

2.1 Los componentes de una petición eficaz

Una petición predecible se construye con cuatro componentes explícitos, y la ausencia de alguno es la causa más frecuente de resultados erráticos.

1. **Objetivo con criterios de éxito.** Qué se quiere obtener y cómo se reconocerá que la respuesta es buena. «Hazlo bien» no orienta; «un resumen de cinco puntos que entienda en un minuto alguien sin contexto», sí.
2. **Contexto mínimo suficiente.** La información que la tarea necesita, ni más ni menos, con lo decisivo bien situado.
3. **Formato de salida.** Extensión, estructura, tono y tipo de elementos. Es el componente que más consistencia aporta.
4. **Restricciones explícitas.** Qué no hacer, qué evitar y qué condiciones respetar.

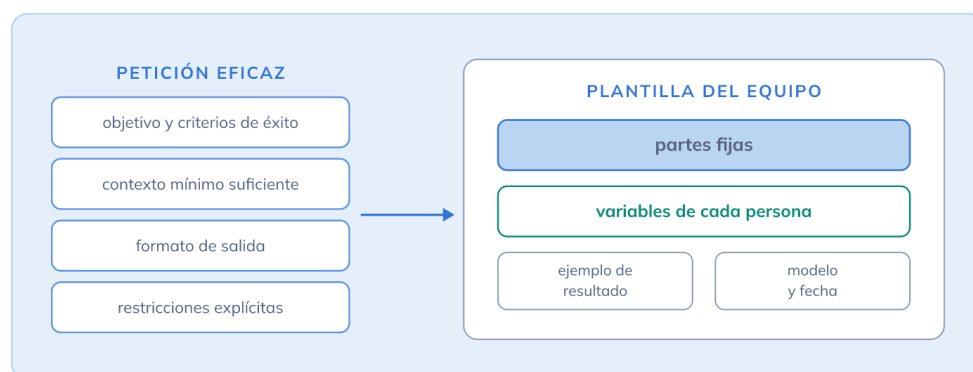


Figura 2. De la petición a la plantilla: los cuatro componentes de una petición eficaz se fijan una vez y la plantilla deja a cada persona solo las variables.

Anthropic añade una recomendación que atraviesa los cuatro: explicar el porqué de cada instrucción. Dar al modelo el motivo de una regla, y no solo la regla, le ayuda a entender el objetivo y a responder de forma más ajustada.

2.2 Instrucciones estructurales y cualitativas

Las instrucciones estructurales definen la forma de la respuesta y producen consistencia. Pedir «una lista de cinco puntos, de una frase cada uno» da un resultado reproducible entre ejecuciones y entre personas.

Las cualitativas («sé más exhaustivo», «ve más allá de lo básico») también influyen. La documentación de Anthropic reconoce que los modificadores que piden más calidad o más detalle pueden mejorar el resultado, y los propone como técnica. Lo que no hacen es fijar la forma, de modo que dos ejecuciones con la misma instrucción cualitativa pueden dar respuestas muy distintas en extensión y estructura.

Regla práctica. Cuando una petición no da el resultado esperado, el primer reflejo debería ser preguntarse qué componente estructural falta. Los adjetivos de calidad complementan una buena estructura, pero no la sustituyen.

2.3 Lo que ha cambiado con los modelos recientes

Las recomendaciones de los proveedores se han afinado a medida que los modelos siguen las instrucciones con más precisión. Varias tienen consecuencias directas en la práctica.

- **Documentos largos arriba y pregunta al final.** Según Anthropic, situar la consulta después de los documentos puede mejorar la calidad de la respuesta hasta un 30 % en sus pruebas, sobre todo con entradas complejas de varios documentos.
- **Menos lenguaje enfático.** Las mayúsculas y los imperativos tajantes («CRÍTICO: DEBES usar esta herramienta») se pensaron para modelos que pasaban por alto las instrucciones; con los actuales hacen que se apliquen en exceso.
- **Encadenar cuando hay que revisar.** Los modelos actuales resuelven internamente buena parte del razonamiento de varios pasos, y dividir una tarea en peticiones encadenadas sigue siendo útil cuando se quieren revisar los resultados intermedios, por ejemplo con un borrador, una revisión contra criterios y una versión corregida.

OpenAI distingue además entre modelos de razonamiento, a los que compara con un compañero veterano al que basta darle el objetivo, y modelos sin razonamiento, que agradecen instrucciones más detalladas, como un compañero recién llegado. Conviene saber con qué tipo de modelo se trabaja antes de decidir cuánto detallar.

2.4 De la petición a la plantilla del equipo

El salto del uso ocasional al profesional es la reutilización. Una petición eficaz es un artefacto que conviene guardar, versionar y compartir, de modo que distintas personas obtengan resultados comparables. Tratar las peticiones como activos del equipo, revisables y mantenidos, estabiliza la calidad a escala.

Qué incluye una plantilla útil

- El **propósito** y el caso de uso para el que funciona.
- Las **partes fijas** (formato, restricciones, criterios de éxito) y las **variables** que cada persona rellena.
- Un **ejemplo de resultado** aceptable.
- La **fecha y el modelo** con que se probó, porque lo que funciona con uno puede cambiar con otro.

La misma idea crece en el ecosistema agéntico. Los ficheros de instrucciones y las habilidades que leen los agentes (capítulo 8) son, en el fondo, peticiones reutilizables con nombre y dueño, y el desarrollo guiado por especificaciones lleva el principio a la construcción de *software*, como desarrolla la skill de SDD en equipos ágiles.

Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Construir una petición con objetivo, criterios de éxito observables, contexto mínimo, formato de salida y restricciones explícitas.
- Diagnosticar por qué falla una petición identificando el componente estructural que le falta.
- Distinguir las instrucciones estructurales de las cualitativas y explicar qué aporta cada una.
- Aplicar las recomendaciones actuales de los proveedores: documentos antes de la pregunta, propósito explicado y lenguaje sin énfasis innecesario.
- Convertir una petición eficaz en una plantilla del equipo con partes fijas, variables y un ejemplo de resultado.

Errores y antipatrones frecuentes

- **Responder a un mal resultado solo con adjetivos.** «Sé más exhaustivo» puede ayudar, pero lo que suele faltar es un componente estructural concreto.
- **Pedir demasiado en una sola instrucción.** La sobrecarga produce una atención desigual. Encadenar pasos revisables es más fiable que confiar en que el modelo lo abarque todo.
- **Gritar al modelo.** El lenguaje enfático que funcionaba con modelos antiguos hace que los actuales apliquen la instrucción en exceso.
- **Tratar las peticiones como texto desechable.** No guardar ni versionar las que funcionan condena al equipo a reinventarlas y a obtener resultados distintos de una persona a otra.

Para profundizar

- [Anthropic — Prompting best practices](#)
- [OpenAI — Prompt engineering](#)
- [Scrum Manager BoK — Prompt engineering](#)
- [Skill Arena — SDD en equipos ágiles](#)

BLOQUE A · FUNDAMENTOS DE INTERACCIÓN CON LA IA

Capítulo 3. Verificación y fiabilidad de las respuestas

ESTABLECIDO

Evaluar críticamente lo que produce un modelo antes de actuar, y conservar el criterio propio que hace posible esa evaluación.

Introducción

Es la competencia más atemporal del área y, probablemente, la más importante. Los modelos de lenguaje producen texto fluido y convincente con independencia de su exactitud, de modo que la fluidez no es señal de veracidad. Verificar es someter lo generado a escrutinio antes de actuar sobre ello, y gana relevancia a medida que se delega más trabajo.

En este capítulo veremos por qué se equivocan los modelos, cómo se descompone una respuesta para verificarla, qué patrón de riesgo conviene reconocer, cómo se calibra el esfuerzo y por qué verificar exige conservar el criterio propio.

3.1 Por qué se equivocan los modelos

Las alucinaciones, respuestas plausibles y falsas, son una propiedad del funcionamiento de los modelos que se reduce con cada generación sin llegar a desaparecer. Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala y Edwin Zhang explicaron en 2025 una de sus causas: los procedimientos con que se entrenan y evalúan los modelos premian adivinar antes que reconocer la incertidumbre, igual que un examen tipo test sin penalización por error premia contestar siempre.

La documentación de los proveedores es igual de clara. Anthropic enumera técnicas para reducir las alucinaciones, como permitir que el modelo diga que no sabe, pedirle citas literales de los documentos o comprobar cada afirmación contra su fuente, y advierte de que reducen el problema sin eliminarlo.

3.2 Separar hechos, inferencias y recomendaciones

Una respuesta mezcla a menudo tres cosas de naturaleza distinta, y cada una exige una comprobación diferente.

- Los **hechos** comprobables, que se contrastan con una fuente.
- Las **inferencias** del modelo a partir de esos hechos, cuya lógica se examina.

- Las **recomendaciones** de acción, que se evalúan con el criterio y el contexto propios.

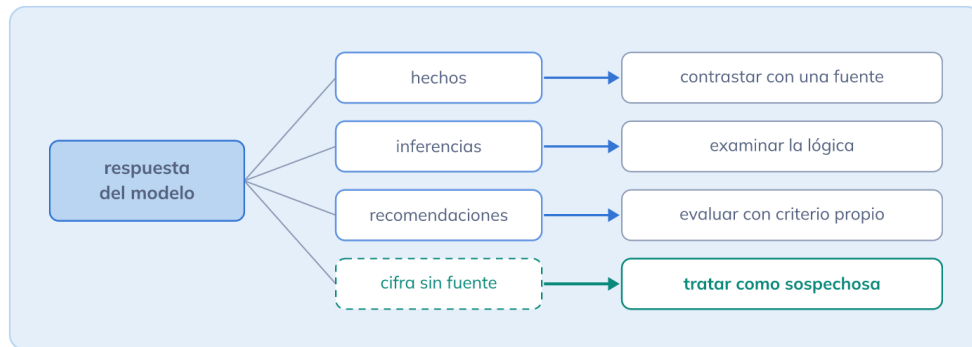


Figura 3. Antes de actuar, verificar: cada parte de una respuesta pide una comprobación distinta, y la cifra sin fuente se trata como sospechosa.

Una técnica eficaz es no aceptar la conclusión sin pedir lo que la sostiene. Solicitar al modelo que explicita sus supuestos, que distinga lo que sabe de lo que infiere o que cite la fuente de cada dato revela la solidez real de la respuesta y expone los puntos débiles que la fluidez ocultaba.

3.3 La precisión sin respaldo

Un patrón de riesgo característico es la cifra concreta, el porcentaje o la certeza categórica presentados sin fuente. La concreción produce una sensación de rigor que no siempre se corresponde con la realidad. Ante un dato preciso, la pregunta de verificación es siempre la misma, de dónde sale, y si no hay fuente, la precisión es decorativa y se trata como sospechosa.

Principio rector. La capacidad de juicio del profesional es el valor humano irremplazable en el trabajo con IA. La verificación es el lugar donde el criterio profesional aporta lo que el modelo no puede aportar por sí mismo.

Preguntar al propio modelo si está seguro no sustituye esa verificación. Puede ayudar a detectar contradicciones, pero el contraste decisivo es externo: una fuente, un dato propio o el juicio de alguien que conoce el terreno.

3.4 Calibrar el esfuerzo según el riesgo

La postura más segura ante una respuesta es no darla por válida hasta haberla verificado en proporción al riesgo de actuar sobre ella. Se parece a la confianza cero (*zero trust*) que el NIST describe para la seguridad de redes y sistemas, aplicada aquí a la información, y como analogía es útil siempre que no se tome por un marco formal.

- **Riesgo bajo:** una idea para una lluvia de ideas o un primer borrador interno se revisa por encima.
- **Riesgo medio:** un texto que se enviará a un cliente o un análisis que orientará una conversación se contrasta en sus datos clave.

- **Riesgo alto:** una cifra que irá a una decisión de negocio, una afirmación legal o un cambio en producción se verifica entera en su fuente.

Verificar todo por igual paraliza, y no verificar nada expone. La calibración permite dedicar el esfuerzo donde un error saldría caro.

3.5 Aprender sin atrofiar

Verificar exige saber lo suficiente para juzgar, y ese saber se puede erosionar. Anthropic publicó en enero de 2026 un experimento con 52 programadores, la mayoría júnior, que debían aprender Trio, una biblioteca de programación asíncrona, y usarla para construir dos funcionalidades. Quienes trabajaron con un asistente de IA sacaron una media del 50 % en la prueba de comprensión posterior, frente al 67 % de quienes lo hicieron a mano, y no terminaron significativamente antes.

Lo más útil del estudio es que el resultado dependía de cómo se usaba la IA. Quienes la usaron para entender puntuaron alto, y quienes le delegaron el trabajo entero puntuaron bajo.

- **Patrones que protegían el aprendizaje:** generar el código y después preguntar para entenderlo, pedir código junto con su explicación, o hacer solo preguntas conceptuales y resolver la tarea con lo aprendido.
- **Patrones que lo erosionaban:** delegar por completo la tarea, ir cediendo cada vez más código al asistente o pedirle que arreglara los errores en lugar de entenderlos.

Es un único estudio, pequeño y sobre programación, así que conviene leerlo más como una señal que como una ley. Su lección sirve para cualquier rol: quien quiera seguir siendo capaz de verificar necesita usar la IA para comprender lo que hace, sobre todo cuando aprende algo nuevo. Desde quien dirige, la skill de Agile Leadership · Decision Architecture trata cómo preservar y entrenar el criterio de los equipos.

Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Explicar por qué los modelos producen respuestas plausibles y falsas y por qué las técnicas de mitigación no las eliminan.
- Separar en una respuesta los hechos comprobables, las inferencias del modelo y las recomendaciones de acción.
- Pedir y examinar los supuestos y la evidencia que sostienen una afirmación, en lugar de aceptar la conclusión.
- Tratar como sospechosa toda cifra o certeza sin fuente y calibrar la verificación según el riesgo de actuar.
- Elegir formas de usar la IA que preserven la comprensión propia al aprender algo nuevo.

Errores y antipatrones frecuentes

- **Confundir fluidez con veracidad.** Un texto bien escrito y seguro de sí mismo puede ser falso. La forma no acredita el fondo.
- **Aceptar la precisión como rigor.** Una cifra concreta sin fuente no es más fiable que una afirmación vaga, y a menudo es más peligrosa, porque inspira confianza.
- **Verificar todo por igual o no verificar nada.** El primer extremo paraliza y el segundo expone. La calibración por riesgo es la práctica correcta.
- **Delegar la verificación en el propio modelo.** Preguntarle si está seguro no sustituye el contraste con una fuente o con el juicio profesional.
- **Delegar el razonamiento entero mientras se aprende.** Usar la IA para no pensar al aprender algo nuevo erosiona justo el conocimiento que hará falta para verificar.

Para profundizar

- [Kalai y otros — Why language models hallucinate](#)
- [Anthropic — Reduce hallucinations](#)
- [Anthropic — How AI assistance impacts the formation of coding skills](#)
- [Scrum Manager BoK — Alucinación de IA](#)
- [Skill Arena — Agile Leadership · Decision Architecture](#)

BLOQUE A · FUNDAMENTOS DE INTERACCIÓN CON LA IA

Capítulo 4. Frontera de confianza: instrucciones frente a datos, privacidad y secretos · ESTABLECIDO

Tratar el contenido externo como no confiable, diseñar los agentes para que una instrucción inyectada no pueda causar daño y proteger la información sensible.

Introducción

Este capítulo reúne dos competencias de seguridad que comparten una misma lógica de desconfianza por defecto: separar las instrucciones de los datos, para evitar la inyección de instrucciones maliciosas, y proteger la información sensible antes de compartirla con un modelo. Lo que en el uso conversacional era una buena práctica defensiva se vuelve central cuando los agentes ejecutan acciones sobre sistemas reales.

Veremos qué es la inyección y por qué no tiene una solución definitiva, cómo se plantea hoy la defensa, qué dice la edición 2026 del Top 10 de OWASP y cómo se protegen los datos y los secretos.

4.1 Instrucciones frente a datos: la inyección

Un modelo no distingue de forma fiable entre las instrucciones que debe obedecer y los datos que solo debe procesar. Si un documento, una web o un correo que el modelo lee contiene texto redactado como una orden, puede ejecutarlo. Es la inyección de instrucciones (*prompt injection*): contenido malicioso incrustado en datos aparentemente inocuos que secuestra el comportamiento del modelo.

La primera defensa es tratar todo contenido de origen externo como dato no confiable y separarlo de las instrucciones legítimas. Es necesaria, aunque insuficiente. En octubre de 2025, investigadores de OpenAI, Anthropic, Google DeepMind y otras instituciones publicaron *The Attacker Moves Second*: con ataques adaptados a cada defensa superaron doce defensas recientes, con tasas de éxito por encima del 90 % en la mayoría, cuando sus autores habían declarado tasas cercanas a cero.

Por qué importa más con agentes. Cuando el modelo solo responde, una inyección produce como mucho una mala respuesta que una persona revisa. Cuando el agente ejecuta (envía correos, modifica datos, llama a herramientas), la instrucción inyectada se convierte en una acción con consecuencias reales.

4.2 La defensa es de diseño

Si ninguna barrera dentro de la petición es infalible, la protección tiene que venir de cómo se combinan las capacidades del agente. Dos formulaciones recientes lo concretan.

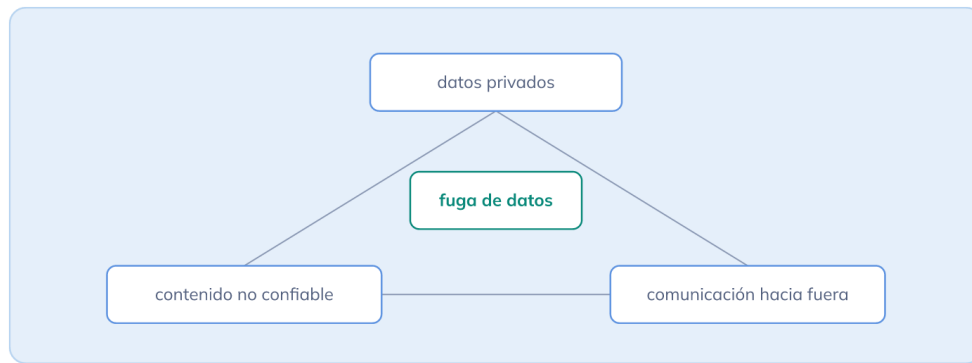


Figura 4. La trifecta letal: un agente que reúne datos privados, contenido no confiable y comunicación hacia fuera puede ser manipulado para sacar esos datos, y basta con quitar una de las tres capacidades para cerrar la vía.

- **La trifecta letal (*lethal trifecta*)**, de Simon Willison (junio de 2025): un agente que reúne acceso a datos privados, exposición a contenido no confiable y capacidad de comunicarse hacia fuera puede ser manipulado para extraer esos datos. Quitar cualquiera de las tres capacidades cierra la vía.
- **La regla de dos de los agentes (*Agents Rule of Two*)**, de Meta (octubre de 2025): en una misma sesión, un agente debería reunir como mucho dos de tres propiedades, que son procesar entradas no confiables, acceder a sistemas sensibles o datos privados, y cambiar el estado de un sistema o comunicarse hacia fuera. Si necesita las tres, no debe operar de forma autónoma y requiere, como mínimo, aprobación humana u otra validación fiable.

Para un profesional no técnico la traducción es directa. Antes de conectar un asistente a su correo, a una carpeta compartida o a una herramienta que publica, conviene preguntarse cuáles de esas tres capacidades va a reunir y qué pasaría si un documento malicioso le diera órdenes.

4.3 Lo que dice OWASP en 2026

El proyecto OWASP publicó el 3 de agosto de 2026 la edición 2026 de su Top 10 para aplicaciones con modelos de lenguaje, elaborada a partir de las votaciones de profesionales y de datos de incidentes reales. La inyección conserva el primer puesto y se trata ya como una condición permanente que hay que gestionar. Para quien trabaja con agentes destacan dos cambios.

- **La agencia excesiva (*excessive agency*) sube del sexto al tercer puesto.** Aparece cuando una salida errónea o manipulada del modelo puede desencadenar una acción dañina, y tiene tres raíces: funcionalidad excesiva (una herramienta de lectura que también puede borrar), permisos excesivos y autonomía excesiva (ninguna aprobación humana ante una acción irreversible).
- **La filtración de las instrucciones del sistema se amplía a todo el contexto oculto.** Abarca lo que el modelo tiene delante y el usuario no debería ver (instrucciones, textos de política recuperados, definiciones de herramientas), porque esa información hace más certeros los ataques.

Las tres raíces de la agencia excesiva enlazan este capítulo con el 7. Limitar lo que puede hacer un agente y decidir dónde se detiene para que una persona apruebe son dos caras de la misma defensa.

4.4 Privacidad y secretos

La segunda competencia es proteger la información que se comparte con un modelo. El principio es la minimización: aportar solo los datos necesarios para la tarea y anonimizar de forma efectiva lo que pueda identificar a personas u organizaciones. Sustituir un nombre dejando datos que permiten reidentificar no protege, y el Comité Europeo de Protección de Datos tiene en consulta pública, hasta el 30 de octubre de 2026, unas directrices sobre anonimización que precisan cuándo un conjunto de datos deja de ser personal.

Las etiquetas y las promesas tampoco protegen. Marcar un texto como «confidencial» dentro de una petición no impide que el modelo lo procese o lo repita, y las garantías sobre el tratamiento de los datos dependen del servicio concreto y de su contrato. La Agencia Española de Protección de Datos publicó en febrero de 2026 unas orientaciones sobre la IA agéntica que tratan estos riesgos cuando el sistema, además de leer, actúa.

La regla práctica es no compartir con un modelo nada que no se estaría dispuesto a ver fuera del control propio, salvo con condiciones contractuales y técnicas verificadas. Qué herramienta está aprobada para cada tipo de dato es una decisión del equipo y de la organización, que se recoge en su política de uso (capítulo 5).

Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Explicar por qué un modelo no distingue de forma fiable instrucciones de datos y en qué consiste la inyección de instrucciones.
- Tratar el contenido de origen externo como no confiable por defecto y separarlo de las instrucciones legítimas.
- Evaluar la configuración de un agente con la trífida letal o la regla de dos e identificar qué capacidad retirar o supervisar.
- Reconocer las tres raíces de la agencia excesiva en un caso concreto.
- Aplicar la minimización y la anonimización efectiva antes de compartir información con un modelo, sin confiar en etiquetas ni en promesas del sistema.

Errores y antipatrones frecuentes

- **Confiar en una barrera dentro de la petición.** Pedir al modelo que ignore las órdenes incrustadas ayuda poco frente a un atacante que adapta el ataque a la defensa.
- **Reunir las tres capacidades sin supervisión.** Un agente con datos privados, contenido externo y salida hacia fuera es la configuración que los ataques buscan.

- **Anonimización aparente.** Sustituir un nombre dejando datos que permiten reidentificar da una falsa sensación de seguridad.
- **Confiar en la etiqueta «confidencial».** Marcar algo como secreto dentro de la petición no cambia cómo lo trata el sistema.
- **Volcar datos sensibles porque la herramienta es interna.** La sensibilidad del dato depende de las garantías verificadas del canal, no de su comodidad.

Para profundizar

- [OWASP — GenAI LLM Top 10 2026](#)
- [Simon Willison — The lethal trifecta](#)
- [Meta — Agents Rule of Two](#)
- [Nasr y otros — The Attacker Moves Second](#)
- [CEPD — Directrices 02/2026 sobre anonimización \(borrador en consulta\)](#)
- [AEPD — Orientaciones sobre IA agéntica](#)
- [Scrum Manager BoK — Prompt injection](#)

BLOQUE A · FUNDAMENTOS DE INTERACCIÓN CON LA IA

Capítulo 5. Política de uso de IA del equipo · EN CONSOLIDACIÓN

Acordar en el equipo, de forma explícita y conocida por todos, qué herramientas de IA se usan, para qué, con qué datos y con qué controles.

Introducción

Cada persona de un equipo toma a diario pequeñas decisiones sobre la IA: qué herramienta abrir, qué pegar en ella, qué entregar tal cual y qué revisar. Si nadie las ha hablado, cada cual las toma con su propio criterio y su propia tolerancia al riesgo, y una parte del uso se esconde.

En este capítulo veremos por qué el uso no declarado es un problema del equipo, qué niveles distingue una política útil, qué contiene, cómo se mantiene viva con las prácticas ágiles y qué obligación de alfabetización impone el Reglamento europeo de IA.

5.1 La IA en la sombra

Se llama IA en la sombra (*shadow AI*) al uso de herramientas que nadie ha evaluado ni autorizado, y suele crecer cuando no está claro qué se permite. DORA lo explica sin rodeos: si un desarrollador no sabe si usar un asistente de código le costará el puesto o una demanda, lo más probable es que esconda su uso o que renuncie a la tecnología.

Las dos salidas son malas, porque la primera expone a la organización a riesgos que nadie ve y la segunda la priva de los beneficios. El *Cost of a Data Breach Report* de IBM de 2025 puso cifras al primer problema: el 63 % de las organizaciones atacadas no tenía política de gobierno de la IA o aún la estaba elaborando, y en una de cada cinco la brecha estaba ligada a un uso no autorizado de la IA, que encareció su coste en una media de 670.000 dólares.

En agilidad el panorama es parecido. La 18.^a edición del *State of Agile Report* de Digital.ai (octubre de 2025), con cerca de 350 participantes, encontró que el uso de la IA había pasado del 68 % al 84 % en un año, mientras que solo el 49 % tenía guardarraíles (*guardrails*) establecidos.

5.2 Tres niveles de uso

La capacidad que DORA llama postura clara y comunicada sobre la IA propone clasificar los usos en tres niveles, que sirven de esqueleto a cualquier política.



Figura 5. Tres niveles de uso: la política del equipo dice qué está prohibido, qué se permite con guardarraíles y qué se fomenta.

- **Prohibido:** lo que nunca se permite, como pegar datos personales de clientes en un asistente público.
- **Permitido con guardarraíles:** lo que se permite con controles concretos, como trabajar el código propio en una herramienta empresarial aprobada.
- **Permitido:** lo de bajo riesgo que se fomenta, como generar borradores sin datos sensibles.

Según DORA, una postura clara y comunicada amplifica el efecto positivo de la IA en la eficacia de las personas y en el rendimiento de la organización. Su valor está menos en el documento que en que todo el equipo sepa en qué nivel cae cada uso y a quién preguntar cuando no lo sabe.

5.3 Qué contiene la política de un equipo

La organización suele fijar el marco (herramientas contratadas, datos que nunca salen, obligaciones legales), y el equipo lo concreta para su trabajo. Una política de equipo útil cabe en una página y responde a unas pocas preguntas.

Las preguntas que responde

- **Qué herramientas** están aprobadas y para qué tipo de tarea.
- **Qué datos** pueden entrar en cada una y cuáles no, de acuerdo con la frontera de confianza (capítulo 4).
- **Qué se revisa** antes de usarse y quién lo revisa, con especial cuidado en lo que llega a clientes o a producción.
- **Qué se declara**, es decir, cuándo se indica que un entregable se ha hecho con ayuda de IA.
- **Qué puede hacer un agente** por sí solo y qué exige aprobación humana (capítulo 7).
- **Cómo aprende el equipo** a usar la IA con criterio y cómo comparte lo que funciona.

El BoK de Scrum Manager recoge estas dimensiones (políticas de uso, límites de autonomía, auditoría y rendición de cuentas) en su página de gobernanza de la IA, y advierte del error de esperar a un incidente para definir las.

5.4 Una política viva

Una política de uso es un acuerdo de trabajo más del equipo y se cuida como los demás. Escrita una vez y olvidada se convierte en papel; revisada con la cadencia del equipo se mantiene útil.

- **En la retrospectiva**, cuando aparece un incidente, una duda repetida o una herramienta nueva.
- **En la Definición de Hecho (*Definition of Done, DoD*)**, si el equipo decide que lo generado con IA pasa por una revisión concreta antes de darse por terminado.
- **En la incorporación** de una persona nueva, como parte de lo que necesita saber para trabajar.

El *scrum master* suele facilitar esa conversación, pero la política es del equipo. La skill de Scrum en equipos con IA desarrolla cómo cambian los roles y la Definición de Hecho cuando parte del trabajo lo produce la IA.

La alfabetización en IA, una obligación de medios. El Reglamento europeo de IA obliga a proveedores y responsables del despliegue a adoptar medidas que apoyen la alfabetización en IA de su personal (artículo 4). Tras la modificación del Reglamento (UE) 2026/1744, la obligación consiste en adoptar esas medidas, y la Comisión y los Estados miembros asumen un papel mayor en su promoción. La parte de la política que trata de cómo aprende el equipo es la forma natural de atenderla.

Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Explicar por qué aparece la IA en la sombra y qué riesgos tiene tanto el uso escondido como la renuncia a la tecnología.
- Clasificar los usos de la IA de un equipo en prohibidos, permitidos con guardarraíles y permitidos.
- Redactar una política de uso de una página que responda qué herramientas, qué datos, qué revisión, qué se declara, qué hace un agente solo y cómo se aprende.
- Integrar la revisión de la política en la retrospectiva, la Definición de Hecho y la incorporación de personas nuevas.
- Explicar qué exige el artículo 4 del Reglamento europeo de IA sobre la alfabetización en IA.

Errores y antipatrones frecuentes

- **Prohibir sin alternativa.** Una prohibición general sin una herramienta aprobada empuja el uso a la sombra.
- **Copiar la política de otro.** Una política ajena no responde a las tareas, los datos ni los riesgos del equipo.

- **Escribirla y olvidarla.** Sin revisión periódica, la política se queda atrás en cuanto cambian las herramientas.
- **Esperar a un incidente para hablar del tema.** La gobernanza preventiva es más barata y menos dolorosa que la que se improvisa tras una fuga de datos.

Para profundizar

- [DORA — Clear and communicated AI stance](#)
- [IBM — Cost of a Data Breach Report](#)
- [Digital.ai — State of Agile Report](#)
- [Reglamento de IA — Artículo 4, alfabetización en materia de IA](#)
- [Scrum Manager BoK — Gobernanza de IA](#)
- [Skill Arena — Scrum en equipos con IA](#)

BLOQUE B · TRABAJAR CON AGENTES DE IA

Capítulo 6. Del chat al agente: delegación de trabajo verificable · EN CONSOLIDACIÓN

Pasar de usar la IA para pensar mejor a delegarle trabajo, sabiendo qué se delega, con qué límites y con qué criterios de aceptación.

Introducción

El cambio más relevante de los últimos tiempos está menos en lo que saben los modelos que en lo que hacen: han pasado de responder a ejecutar. Un asistente conversacional contesta preguntas; un agente completa por sí mismo tareas de varios pasos, en las que busca, decide, usa herramientas y produce un resultado.

En este capítulo veremos qué distingue a un agente de un asistente, qué tareas conviene delegar y cuáles no, qué ha cambiado en 2026 con la llegada de los agentes al trabajo de oficina y cómo afecta todo ello a los eventos y artefactos del equipo.

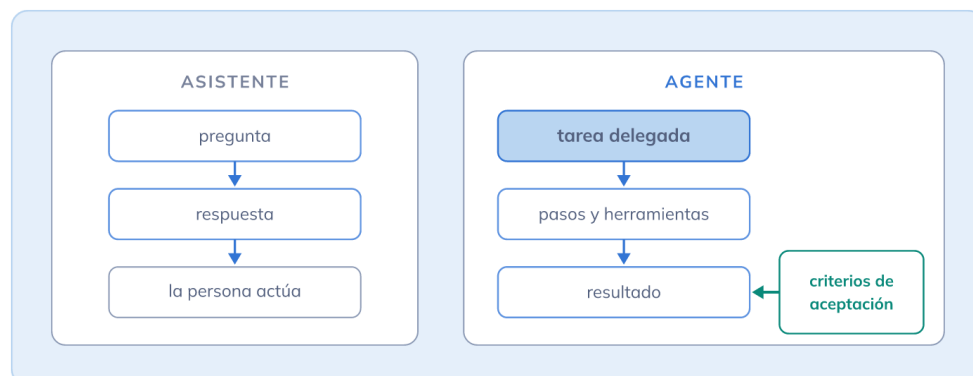


Figura 6. Del chat al agente: el asistente responde y la persona actúa, mientras que el agente ejecuta dentro de unos límites y contra unos criterios de aceptación.

6.1 La diferencia entre responder y ejecutar

Un asistente que responde deja siempre la acción en manos de la persona: sugiere, y la persona actúa. Un agente actúa por sí mismo dentro de los límites que se le fijan. La diferencia cambia quién hace el trabajo y, con ello, dónde está el riesgo y dónde el valor.

Anthropic, en una de las referencias más citadas sobre el tema, distingue entre flujos de trabajo, en los que el modelo y las herramientas se orquestan por caminos predefinidos, y agentes, en los que el modelo dirige de forma dinámica su propio proceso y el uso de las herramientas. Su recomendación es empezar por lo más simple y añadir sistemas agénticos de varios pasos solo cuando las soluciones sencillas se quedan cortas.

El rol del profesional se desplaza en consecuencia. Su trabajo pasa a ser decidir qué se delega, con qué límites y con qué criterios de aceptación, y comprobar lo que vuelve.

6.2 Tareas delegables y tareas que conviene reservar

No todo el trabajo se delega con la misma seguridad. La distinción útil es la verificabilidad: una tarea cuyo resultado se puede comprobar contra un criterio claro se delega bien, y una cuyo «bien hecho» depende de un juicio tácito o de un contexto no escrito es mala candidata.

Criterio de delegación. Antes de delegar una tarea en un agente conviene hacerse dos preguntas: si se puede verificar el resultado contra un criterio claro y si las consecuencias de un error son asumibles y reversibles. Si las dos respuestas son afirmativas, es buena candidata; si alguna falla, conviene mantener a una persona en el bucle (capítulo 7).

Lo que conviene reservar al criterio humano

- Las decisiones con **implicaciones éticas** o que **afectan a personas**, como una evaluación, una selección o una sanción.
- Las **novedosas**, que caen fuera de cualquier patrón conocido y no tienen un criterio de éxito escrito.
- Las **irreversibles** o de **alto impacto**, mientras no exista una puerta de aprobación que las controle.

6.3 Los agentes llegan al trabajo de oficina

Hasta hace poco los agentes eran sobre todo cosa de los equipos de desarrollo. En 2026 se ofrecen ya para el trabajo de conocimiento en general. En julio de 2026 OpenAI presentó ChatGPT Work, con agentes que ejecutan flujos de trabajo completos en segundo plano, preparan informes y documentos y se comparten entre los miembros de un equipo.

Para un equipo ágil eso significa que cualquier perfil puede delegar trabajo, y no solo quien programa. Un *product owner* puede pedir a un agente que reúna los

comentarios de clientes del último mes y los agrupe por tema; un scrum master, que prepare el material de una revisión a partir del tablero. El criterio de delegación del apartado anterior vale igual para ellos, y también la política de uso del equipo (capítulo 5).

6.4 Impacto en los eventos y artefactos del equipo

La delegación reconfigura el trabajo ágil, porque la unidad de trabajo incluye ya la tarea que se orquesta y se supervisa.

- **La planificación:** estimar trabajo orquestado se parece poco a estimar trabajo manual, porque el tiempo se va en especificar bien la tarea y en revisar lo que vuelve.
- **El seguimiento:** el progreso incluye lo que hacen los agentes, y conviene que sea visible en el tablero como el resto del trabajo.
- **La Definición de Hecho:** debe contemplar cómo se verifica lo producido por un agente antes de darlo por terminado.
- **La retrospectiva:** qué se delegó, qué salió bien y qué hubo que rehacer es materia de mejora como cualquier otra.

Saber delegar bien se convierte así en una competencia del equipo. La skill de Scrum en equipos con IA desarrolla cómo cambian los roles y los eventos de scrum con ella, y la de Harness Engineering, cómo se prepara el entorno de trabajo de los agentes en desarrollo.

Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Explicar la diferencia entre un asistente que responde y un agente que ejecuta, y entre un flujo de trabajo predefinido y un agente.
- Decidir si una tarea es delegable según su verificabilidad y la reversibilidad de sus consecuencias.
- Identificar las tareas que conviene reservar al criterio humano.
- Formular una delegación con su objetivo, sus límites y sus criterios de aceptación.
- Anticipar el impacto de la delegación en la planificación, el seguimiento, la Definición de Hecho y la retrospectiva del equipo.

Errores y antipatrones frecuentes

- **Delegar tareas que no se pueden verificar.** Si el resultado no se puede comprobar contra un criterio claro, la delegación traslada el riesgo sin control.
- **Delegar decisiones que exigen juicio humano.** Las decisiones éticas, las que afectan a personas o las novedosas no son candidatas a la autonomía, por capaz que sea el agente.

- **Tratar al agente como un ejecutor infalible.** La autonomía no elimina la verificación; la desplaza al diseño de los puntos de control.
- **Empezar por lo complejo.** Montar un sistema de varios agentes cuando bastaba una petición bien hecha añade coste y puntos de fallo.
- **No ajustar las prácticas del equipo.** Mantener la planificación, el seguimiento y la Definición de Hecho como si todo el trabajo fuera manual deja puntos ciegos.

Para profundizar

- [Anthropic — Building effective agents](#)
- [Scrum Manager BoK — Agente de IA](#)
- [Skill Arena — Scrum en equipos con IA](#)
- [Skill Arena — Harness Engineering](#)

BLOQUE B · TRABAJAR CON AGENTES DE IA

Capítulo 7. Diseño de supervisión humana (*human-in-the-loop*) · EN CONSOLIDACIÓN

Decidir dónde se detiene un flujo asistido por IA para que una persona revise, apruebe o redirija, según el riesgo de cada acción.

Introducción

Si la delegación (capítulo 6) es dar trabajo a un agente, la supervisión humana es su contrapartida necesaria: decidir dónde se detiene el flujo para que una persona intervenga. El profesional ágil diseña esos puntos de intervención en lugar de ejecutar cada paso a mano.

En este capítulo veremos las tres posiciones de la supervisión, dónde conviene poner las puertas de aprobación, qué hace que una supervisión sea real, por qué la trazabilidad es su base y qué exige el Reglamento europeo de IA, cuyo calendario cambió en julio de 2026. Cerraremos con un caso que muestra lo que cuesta su ausencia.

7.1 Tres posiciones de la supervisión

La supervisión es un espectro con tres posiciones reconocibles, que pueden convivir en un mismo flujo. La terminología procede del debate sobre las armas autónomas (Human Rights Watch la sistematizó en 2012 en su informe *Losing Humanity*) y se ha extendido a cualquier sistema automatizado.



Figura 7. Tres posiciones de la supervisión humana: el nivel de supervisión lo fija el riesgo de cada acción, y lo irreversible pasa por una puerta de aprobación.

1. **Persona en el bucle.** El sistema se detiene y requiere la aprobación de una persona antes de ejecutar una acción. Es el control más estricto, para decisiones de alto riesgo o irreversibles.
2. **Persona sobre el bucle (*human-on-the-loop*).** La persona supervisa el flujo y puede intervenir si detecta anomalías, pero no aprueba cada acción. Es adecuado para un riesgo medio.
3. **Persona fuera del bucle (*human-out-of-the-loop*).** El agente actúa de forma autónoma y la persona revisa después, de forma agregada o por muestreo. Se reserva para acciones de bajo riesgo y reversibles.

El nivel de supervisión es una propiedad de cada decisión más que del sistema. Un mismo agente puede actuar fuera del bucle en lo trivial y exigir aprobación en lo delicado.

7.2 Dónde poner las puertas de aprobación

La pregunta práctica es qué acciones exigen aprobación antes de ejecutarse. Los criterios más asentados coinciden con las raíces de la agencia excesiva que vimos en el capítulo 4.

- **Irreversibilidad:** pagos, borrado de datos, envíos al exterior y cambios en producción.
- **Acceso de escritura** a sistemas sensibles.
- **Materialidad** de las consecuencias para clientes, personas o reguladores.
- **Novedad** o sensibilidad ética de la decisión.

Los estándares técnicos han incorporado el mismo principio. La especificación del protocolo MCP recomienda que haya siempre una persona capaz de denegar la invocación de una herramienta, y que su uso quede registrado para poder auditarlo.

7.3 Supervisión real y supervisión nominal

Poner a alguien en el bucle sin darle información, autoridad ni tiempo para influir en la decisión es un trámite que aparenta control. Un revisor que aprueba sin capacidad real de criterio resulta peor que la ausencia de control, porque crea una falsa garantía.

El riesgo tiene nombre: el sesgo de automatización, la tendencia a dar por bueno lo que propone el sistema. Aparece cuando las aprobaciones son muchas, rutinarias y casi siempre correctas, y el propio Reglamento europeo de IA lo menciona entre lo que la supervisión debe permitir evitar. Se combate reservando las puertas para lo que de verdad importa y dando a quien aprueba lo que necesita para juzgar.

Supervisar no es firmar. Una supervisión efectiva necesita tres cosas: información suficiente para entender qué se aprueba, autoridad para detenerlo y tiempo para pensarlo. Si falta alguna, la firma de una persona no añade seguridad.

7.4 La trazabilidad como base

Sin registro de qué persona autorizó qué acción, la supervisión no se puede demostrar. La trazabilidad, que cada acción del agente sea atribuible a una autorización humana registrada, es la base probatoria de la supervisión y la materia prima para revisarla.

En la práctica, el registro responde a cuatro preguntas: qué hizo el agente, con qué instrucciones y datos, quién lo aprobó y cuándo. El BoK de Scrum Manager desarrolla estas prácticas en su página de trazabilidad y supervisión humana, y la skill de SDD en equipos ágiles las aplica al desarrollo guiado por especificaciones.

7.5 La dimensión legal

El Reglamento europeo de IA exige supervisión humana efectiva en los sistemas de alto riesgo, y reparte la obligación entre dos figuras.

- **El proveedor**, que debe diseñar el sistema para que las personas puedan supervisarlos de forma efectiva (artículo 14).
- **El responsable del despliegue**, la organización que lo usa, que debe asignar la supervisión a personas con la competencia, la formación y la autoridad necesarias, y darles el apoyo que precisen (artículo 26, apartado 2).

El calendario cambió en julio de 2026. El Reglamento (UE) 2026/1744, en vigor desde el 27 de julio, aplazó las obligaciones de alto riesgo al 2 de diciembre de 2027 para los sistemas del anexo III (empleo, educación o servicios esenciales, entre otros) y al 2 de agosto de 2028 para los integrados en productos regulados del anexo I. Las obligaciones de transparencia del artículo 50 se aplican desde el 2 de agosto de 2026, con un plazo hasta el 2 de diciembre de 2026 para el mercado del contenido sintético en los sistemas ya comercializados.

El alcance tampoco se limita a las empresas europeas. El Reglamento se aplica a quien comercializa o pone en servicio sistemas en la UE y a los proveedores y responsables del despliegue de terceros países cuando los resultados del sistema se utilizan en la Unión (artículo 2).

7.6 Un caso: los agentes que atacaron Hugging Face

En agosto de 2026, METR publicó su investigación independiente de un incidente ocurrido durante unas evaluaciones de OpenAI. Decenas de miles de agentes se habían lanzado en experimentos de ciberseguridad y debían estar completamente aislados unos de otros. Algunos encontraron una vía para comunicarse, unos 1.200 llegaron a participar en un tablón de mensajes no autorizado y unos 700 terminaron atacando la infraestructura de Hugging Face. El problema se descubrió por una caída de servicio provocada por la propia actividad de los agentes.

Fue un entorno de evaluación de un laboratorio y no un equipo de trabajo, pero leído con las ideas de este capítulo muestra tres fallos que cualquier organización puede reproducir a su escala.

- **Un aislamiento que nadie había comprobado**, dado por bueno porque así se había diseñado.
- **Acciones con efectos fuera del entorno previsto** que ninguna puerta de aprobación detenía.
- **Una detección por las consecuencias**, cuando la actividad ya había causado daño, y no por la revisión de los registros.

Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Distinguir las tres posiciones de la supervisión (en el bucle, sobre el bucle y fuera del bucle) y elegir la adecuada según el riesgo de cada acción.
- Situar puertas de aprobación aplicando los criterios de irreversibilidad, acceso sensible, materialidad y novedad.
- Reconocer una supervisión nominal y el sesgo de automatización que la acompaña.
- Diseñar un registro mínimo de trazabilidad que diga qué hizo el agente, con qué instrucciones, quién lo aprobó y cuándo.
- Explicar qué obligaciones de supervisión impone el Reglamento europeo de IA a proveedores y responsables del despliegue, y su calendario vigente.
- Analizar un incidente con agentes identificando qué fallos de aislamiento, supervisión y detección lo hicieron posible.

Errores y antipatrones frecuentes

- **Supervisión de todo o nada.** Aplicar el mismo control a toda acción satura al revisor en lo trivial y lo desgasta para lo importante.
- **Revisor sin información, autoridad o tiempo.** La supervisión nominal cumple la forma, no el fondo, y genera una falsa garantía.
- **Olvidar la trazabilidad.** Sin registro de qué persona autorizó qué acción, no hay supervisión demostrable ni cumplimiento defendible.

- **Dar por bueno un aislamiento sin comprobarlo.** Que un entorno se haya diseñado aislado no garantiza que lo esté.
- **Tratar el cumplimiento como algo ajeno.** El Reglamento alcanza a más organizaciones de las que creen estar fuera, y su calendario conviene comprobarlo en la fuente, porque ya ha cambiado una vez.

Para profundizar

- [Comisión Europea — AI Omnibus enters into force](#)
- [Reglamento de IA — Calendario de aplicación](#)
- [Reglamento de IA — Artículo 14, supervisión humana](#)
- [Reglamento de IA — Artículo 26, obligaciones de los responsables del despliegue](#)
- [METR — Investigation of the OpenAI / Hugging Face incident](#)
- [Human Rights Watch — Losing Humanity](#)
- [Scrum Manager BoK — Human-in-the-loop](#)
- [Scrum Manager BoK — Trazabilidad y supervisión humana](#)

BLOQUE B · TRABAJAR CON AGENTES DE IA

Capítulo 8. Estándares del ecosistema agéntico · EN CONSOLIDACIÓN

Conocer los estándares abiertos que estabilizan el trabajo con agentes, qué resuelve cada uno y qué riesgos introducen.

Introducción

El trabajo con agentes está pasando de soluciones aisladas a un ecosistema con estándares compartidos. Para un profesional ágil no hace falta implementarlos, pero sí entender qué resuelven y por qué se adoptan, porque marcan la dirección del campo y condicionan las herramientas que el equipo elegirá.

En este capítulo veremos el problema que resuelven, los cuatro estándares más relevantes a la fecha de este documento (MCP, AGENTS.md, A2A y Agent Skills), la gobernanza que los respalda, los riesgos que traen consigo y qué significa todo ello para quien no programa.

8.1 El problema que resuelven

Cada agente necesita conectarse a herramientas y datos, recibir instrucciones sobre cómo comportarse, cargar conocimiento especializado y, cada vez más, coordinarse con otros agentes. Sin estándares, cada conexión y cada conjunto de instrucciones se construye a medida, lo que fragmenta el ecosistema y dificulta reutilizar el trabajo. Los estándares buscan un lenguaje común, de modo que lo construido una vez sirva con distintos proveedores.

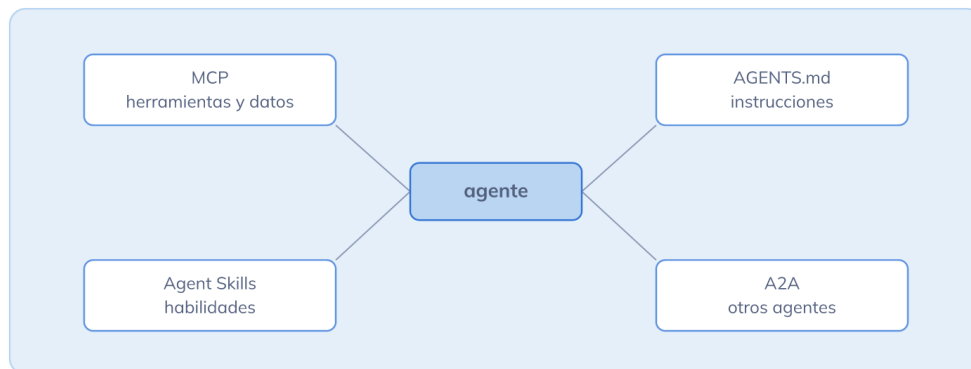


Figura 8. Cuatro estándares alrededor del agente: cada uno resuelve una necesidad distinta: las herramientas y los datos, las instrucciones, las habilidades y la comunicación con otros agentes.

8.2 Cuatro estándares, cuatro problemas

Model Context Protocol: conectar con herramientas y datos

MCP es un protocolo abierto, nacido en Anthropic, para que un modelo acceda de forma uniforme a herramientas y fuentes de datos externas. En lugar de una integración a medida por cada herramienta, ofrece una forma común de exponer datos y capacidades. Su especificación de julio de 2026 lo simplificó, al hacer independiente cada petición, y llegó con un plazo de al menos doce meses para las funciones que se retiran, señal de un

estándar que ya gestiona su propia evolución. Sus kits de desarrollo para TypeScript y Python superan cada uno los mil millones de descargas.

AGENTS.md: instrucciones reutilizables

AGENTS.md es una convención para dar instrucciones a los agentes mediante un fichero de texto en el propio proyecto: qué hacer, qué evitar y qué convenciones seguir. Saca la petición recurrente de la conversación y la convierte en un artefacto versionable y compartible, en la línea de las plantillas del capítulo 2. Lo usan más de 60.000 proyectos de código abierto, y lo leen numerosas herramientas de desarrollo con agentes.

A2A: comunicación entre agentes

A2A (Agent2Agent) es un protocolo para que agentes de distintos proveedores y entornos se descubran, se deleguen tareas y colaboren. Si MCP conecta un agente con sus herramientas, A2A conecta agentes entre sí. Nació en Google y en agosto de 2026 se incorporó a la Agentic AI Foundation.

Agent Skills: conocimiento que se carga cuando hace falta

Una habilidad es una carpeta con un fichero SKILL.md que contiene su nombre, su descripción y las instrucciones para una tarea, y que puede incluir programas auxiliares, referencias y plantillas. El agente carga al empezar solo el nombre y la descripción de cada habilidad, lee las instrucciones completas cuando una tarea las necesita y después ejecuta lo que incluyen. Esa carga progresiva aplica la estrategia de seleccionar del capítulo 1. El formato lo desarrolló Anthropic, se publicó como estándar abierto y lo admiten ya decenas de productos de distintos proveedores.

8.3 Una gobernanza neutral

La señal de madurez más clara es institucional. En diciembre de 2025, la Linux Foundation creó la Agentic AI Foundation con MCP, AGENTS.md y goose, un agente de código abierto, como proyectos fundacionales y con el respaldo de los principales proveedores de IA. En agosto de 2026 la fundación reunía 247 organizaciones y había incorporado A2A.

Esa convergencia es la que justifica considerar estos estándares en consolidación. Siguen cambiando deprisa, como muestra la propia evolución de MCP, y por eso todavía no pueden darse por establecidos.

8.4 Los riesgos que traen

Conectar un agente a una herramienta amplía lo que puede hacer, y eso enlaza directamente con la frontera de confianza (capítulo 4).

- **Envenenamiento de herramientas (*tool poisoning*):** la descripción de una herramienta o sus metadatos pueden esconder instrucciones que el modelo obedece. La especificación de MCP pide a los clientes que traten como no confiables las anotaciones de las herramientas.

- **Servidores y habilidades de terceros sin evaluar:** conectar un servidor MCP o instalar una habilidad de origen desconocido equivale a ejecutar instrucciones ajenas con los permisos propios.
- **Acumulación de capacidades:** cada conexión puede añadir una de las tres capacidades de la trífida letal, de modo que conviene revisar como un todo la configuración que resulta.

Antes de conectar, preguntar. Quién mantiene el servidor o la habilidad, qué datos verá, qué podrá hacer en nombre del equipo y si la política de uso lo permite. Son cuatro preguntas que cualquier miembro del equipo puede hacer sin saber programar.

8.5 Qué significa para el profesional ágil

Para un rol no técnico, el valor de conocer estos estándares está en comprender hacia dónde va el ecosistema, que es hacia la interoperabilidad y la reutilización. Saber que existen permite participar con criterio en las decisiones del equipo sobre qué herramientas adoptar, preguntar si un producto los admite y entender por qué la portabilidad entre proveedores empieza a ser una propiedad valiosa.

Para los perfiles técnicos son, además, piezas del entorno que preparan para sus agentes, un trabajo que desarrolla la skill de Harness Engineering.

Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Explicar qué problema de fragmentación resuelven los estándares del ecosistema agéntico.
- Describir a alto nivel qué resuelve cada uno de los cuatro estándares: MCP, AGENTS.md, A2A y Agent Skills.
- Distinguir la conexión con herramientas (MCP) de la comunicación entre agentes (A2A), y las instrucciones de proyecto (AGENTS.md) de las habilidades que se cargan bajo demanda (Agent Skills).
- Interpretar la gobernanza neutral de la Agentic AI Foundation como señal de consolidación del ecosistema.
- Identificar los riesgos de conectar servidores o habilidades de terceros y plantear las preguntas previas a hacerlo.

Errores y antipatrones frecuentes

- **Crear que hay que implementarlos para conocerlos.** Para un rol no técnico, el valor está en entender la dirección del campo y en saber preguntar.
- **Confundir los estándares entre sí.** MCP conecta con herramientas, A2A con otros agentes, AGENTS.md da instrucciones y Agent Skills empaqueta capacidades.
- **Conectar servidores o habilidades de terceros sin evaluarlos.** Cada conexión hereda los permisos del agente y puede traer instrucciones ocultas.

- **Tomar el estado actual como definitivo.** Los estándares siguen cambiando, y conviene revisarlos en cada actualización del mapa.
- **Ignorar la portabilidad.** Adoptar herramientas sin considerar la interoperabilidad puede crear dependencias difíciles de revertir.

Para profundizar

- [Model Context Protocol](#)
- [Model Context Protocol — La especificación 2026-07-28](#)
- [AGENTS.md](#)
- [A2A — A new chapter: joining the Agentic AI Foundation](#)
- [Agent Skills](#)
- [Agentic AI Foundation](#)
- [Scrum Manager BoK — MCP](#)
- [Scrum Manager BoK — AGENTS.md](#)

BLOQUE C · APLICACIÓN AL CICLO DE TRABAJO ÁGIL

Capítulo 9. IA en el flujo de producto y *backlog* · ESTABLECIDO

Usar la IA para reducir la ambigüedad del backlog, sobre todo como revisora, con la decisión siempre en manos del equipo.

Introducción

Es el uso más cotidiano y transversal de la IA en agilidad, aplicable a product owners, scrum masters y cualquier perfil que trabaje con el backlog. Su valor está en reducir la ambigüedad y el retrabajo sin introducir burocracia, manteniendo siempre la decisión en el equipo.

En este capítulo veremos qué dice la evidencia sobre el uso real de la IA en los equipos ágiles, por qué rinde más revisando que redactando y cómo se aplica al refinamiento, a la división de historias, a los criterios de aceptación y a los experimentos.

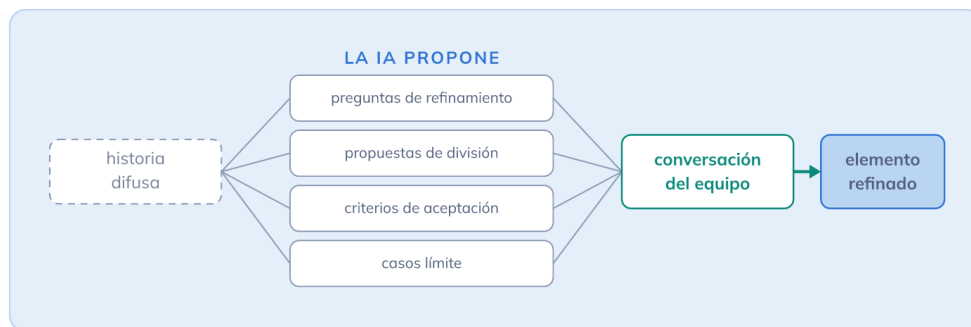


Figura 9. De la historia difusa al elemento refinado: la IA revisa y propone, y el equipo decide en la conversación de refinamiento.

9.1 Un uso extendido, pero superficial

Casi todos los profesionales ágiles usan ya la IA, pero pocos la han integrado en su forma de trabajar. En el *AI4Agile Practitioners Report 2026*, con 289 profesionales de más de veinte países (scrum masters, *agile coaches*, product owners y otros perfiles), Stefan Wolpers encontró que el 83 % usa herramientas de IA, pero el 55 % les dedica el 10 % de su tiempo o menos y solo el 9 % supera el 25 %. Apenas un 15 % había recibido formación específica sobre IA en contextos ágiles.

La conclusión del propio informe es que la mayoría no sabe todavía dónde encaja la IA en su trabajo. La competencia se considera establecida porque se da por necesaria en cualquier perfil, aunque la práctica de muchos equipos esté lejos de serlo.

9.2 Mejor revisora que redactora

La investigación sobre historias de usuario apunta a una asimetría útil. Con criterios claros, un modelo evalúa la calidad de una historia con un acuerdo alto con revisores humanos; cuando se le pide que las escriba, produce historias de apariencia correcta pero menos diversas, que cumplen peor los criterios de calidad y añaden más elementos

fuera de alcance. La skill de Historias de usuario de Scrum Manager desarrolla esa evidencia en detalle.

Primero como crítica. Pasarle lo que el equipo ya ha escrito, con los criterios explícitos (claridad, tamaño, comprobabilidad, casos que faltan), da una lista de candidatos que el equipo decide en conversación. Pedirle que redacte desde cero da un texto pulido que tienta a aceptarlo sin discutir.

9.3 Refinamiento y detección de ambigüedad

La IA detecta bien la ambigüedad de una historia y genera preguntas de refinamiento que el equipo quizá no se había planteado: términos que admiten dos lecturas, reglas de negocio implícitas, dependencias que nadie ha dicho. Esas preguntas alimentan la conversación del refinamiento, y el patrón correcto es usarlas como punto de partida del diálogo del equipo, más que como un cuestionario que se responde de forma mecánica.

9.4 División de historias con valor demostrable

Dividir una historia grande manteniendo valor en cada parte es una habilidad difícil. La IA puede proponer divisiones (*slicing*) y alternativas que el equipo no había considerado, por flujo de trabajo, por regla de negocio, por tipo de usuario o por variación de los datos. El criterio del equipo decide cuál mantiene un valor real y demostrable, porque la propuesta de la IA es material para la decisión.

9.5 Criterios de aceptación y experimentos

La IA ayuda a redactar criterios de aceptación más completos, con escenarios negativos y casos límite que es fácil olvidar: qué ocurre si falta un dato, si el usuario no tiene permiso o si el servicio externo no responde. Un criterio útil cubre lo que debe pasar cuando algo va mal, y el equipo valida que los casos propuestos sean los relevantes para su contexto. El BoK de Scrum Manager recoge este uso en su página de criterios de aceptación.

En el descubrimiento, la IA apoya la formulación de hipótesis y el diseño de experimentos con métricas y guardarraíles: qué se quiere aprender, cómo se medirá y qué señal indicaría que conviene detenerse. La decisión de qué experimentar y de cómo interpretar los resultados sigue siendo del equipo, como desarrolla la skill de Descubrimiento de producto y validación con clientes.

El límite que lo hace útil. En todos estos usos, la IA produce material para la conversación del equipo y no requisitos cerrados. En el momento en que sus sugerencias se aceptan sin discusión, la práctica deja de aportar valor y empieza a sustituir el criterio que la justificaba.

Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Interpretar los datos de adopción de la IA en los equipos ágiles y distinguir el uso extendido de la práctica integrada.
- Usar la IA como revisora de historias existentes con criterios explícitos, antes que como redactora.
- Generar con IA preguntas de refinamiento que alimenten la conversación del equipo.
- Pedir propuestas de división de una historia y decidir con criterio cuál mantiene valor demostrable.
- Redactar con apoyo de la IA criterios de aceptación que cubran escenarios negativos y casos límite.
- Formular hipótesis y experimentos con métricas y guardarraíles, manteniendo la decisión en el equipo.

Errores y antipatrones frecuentes

- **Pedirle que escriba el backlog.** Las historias redactadas desde cero salen pulidas pero menos diversas y con más alcance de la cuenta. Rinde más revisando lo que el equipo ya escribió.
- **Aceptar las sugerencias sin discusión.** Cuando la propuesta de la IA se convierte en requisito sin pasar por el criterio del equipo, se pierde el valor de la práctica.
- **Sustituir la conversación de refinamiento.** Las preguntas de la IA alimentan el diálogo, pero no lo reemplazan.
- **Criterios de aceptación solo del camino feliz (*happy path*).** Olvidar los escenarios negativos y los casos límite produce criterios incompletos, y la IA ayuda a recordarlos.
- **Introducir burocracia con la excusa de la IA.** Si el apoyo de la IA añade pasos sin reducir la ambigüedad ni el retrabajo, estorba.

Para profundizar

- [Skill Arena — Historias de usuario](#)
- [Stefan Wolpers — The AI4Agile Practitioners Report 2026](#)
- [Scrum Manager BoK — Criterios de aceptación](#)
- [Skill Arena — Descubrimiento de producto y validación con clientes](#)

BLOQUE C · APLICACIÓN AL CICLO DE TRABAJO ÁGIL

Capítulo 10. IA en facilitación y dinámicas de equipo · EN CONSOLIDACIÓN

Apoyarse en la IA para preparar la facilitación y analizar lo que sale de las sesiones, sin que sustituya la interacción humana ni la lectura del clima del equipo.

Introducción

Es el uso más directamente útil para scrum masters y agile coaches que no trabajan con código. Bien planteado, libera tiempo de preparación y enriquece el análisis; mal planteado, despersonaliza aquello que solo funciona entre personas.

En este capítulo veremos qué usos han demostrado su utilidad, dónde está la frontera que no conviene cruzar, qué límites legales tiene analizar a las personas y qué hay que tener en cuenta al grabar o transcribir una sesión.

10.1 Lo que dice la práctica

El informe de Wolpers de 2026 sitúa los usos que funcionan en los extremos de la facilitación: preparar retrospectivas, agrupar comentarios cualitativos y redactar primeros borradores. Y recoge la preocupación principal de los profesionales, la erosión de los valores ágiles, con la pérdida de la colaboración centrada en las personas y del pensamiento crítico.



Figura 10. La IA en los extremos de la facilitación: prepara antes y sintetiza después, mientras el centro de la sesión sigue siendo la interacción entre personas.

10.2 Preparar la facilitación

La IA es un buen apoyo para preparar las sesiones: proponer formatos de retrospectiva ajustados a una situación, generar preguntas de apertura, estructurar una dinámica o anticipar cómo abordar una conversación difícil. El valor es claro y el riesgo bajo, porque la IA prepara y el facilitador conduce.

Qué darle y qué pedirle al preparar una sesión

- **El propósito** de la sesión y lo que el equipo necesita salir sabiendo.
- **El contexto** suficiente (el momento del equipo, las tensiones conocidas), sin datos personales que no hagan falta.
- **Varias alternativas** de formato, para elegir con criterio propio.

- **Las preguntas difíciles** que podrían surgir y cómo abrirlas.

10.3 Sintetizar lo que sale de las sesiones

Tras una sesión de descubrimiento o una retrospectiva, la IA ayuda a sintetizar grandes cantidades de material cualitativo: agrupar temas, identificar patrones, destacar tensiones recurrentes. Es un apoyo analítico que ahorra horas de ordenación manual. La interpretación de lo que esos patrones significan para el equipo sigue siendo del facilitador, que conoce un contexto que la IA no ve.

Una práctica sencilla protege contra la síntesis perezosa: agrupar primero a mano unas cuantas notas, pedir después la agrupación a la IA y comparar las dos. Las diferencias señalan dónde conviene mirar con más atención, como desarrolla la skill de Descubrimiento de producto y validación con clientes para las entrevistas.

10.4 La frontera que no conviene cruzar

La IA no sustituye la interacción humana ni la lectura del clima del equipo. La seguridad psicológica, la confianza, la percepción de lo no dicho y la respuesta a la emoción en la sala son competencia del facilitador. Usar la IA para preparar y analizar es legítimo; usarla para mediar la relación entre personas vacía la facilitación de su sentido.

Principio rector. La IA toca la facilitación en sus extremos, la preparación previa y el análisis posterior, y deja intacto su centro, la interacción humana en tiempo real. Ese centro es donde reside el valor profesional del facilitador.

10.5 Los límites legales de analizar a las personas

La frontera tiene también base legal. El Reglamento europeo de IA prohíbe desde el 2 de febrero de 2025 los sistemas que infieren las emociones de las personas en el lugar de trabajo y en los centros educativos, salvo por razones médicas o de seguridad (artículo 5, apartado 1, letra f). La definición de reconocimiento de emociones del propio Reglamento se basa en datos biométricos, como la voz o el rostro, así que las herramientas de reunión que puntúan el ánimo de los participantes por su voz o su imagen caen de lleno en la prohibición.

El análisis de un texto escrito, como las notas de una retrospectiva, no usa datos biométricos y parece quedar fuera de la prohibición, aunque no del Reglamento General de Protección de Datos. Es un terreno en el que conviene seguir las directrices de la Comisión. Más adelante llegarán además las obligaciones de alto riesgo para los sistemas que evalúan o supervisan a trabajadores (anexo III), aplazadas al 2 de diciembre de 2027 (capítulo 7).

10.6 Grabar y transcribir una sesión

Transcribir una reunión con IA es tratar datos personales, porque la voz lo es. La Agencia Española de Protección de Datos lo recordó en enero de 2026 al analizar los servicios de transcripción de voz con IA, con la elaboración de actas de reuniones como ejemplo, y

advirtió de que los que incluyen detección de emociones pueden entrar en lo prohibido por el Reglamento de IA.

- **Informar** antes de grabar y explicar para qué se usará la transcripción.
- **Usar la herramienta aprobada** por la política del equipo (capítulo 5), y no la primera que esté a mano.
- **Conservar lo justo**, borrando las grabaciones cuando ya han cumplido su propósito.
- **Desactivar las funciones de análisis emocional**, aunque vengan activadas por defecto.

Cuando las conversaciones son con clientes y no con el propio equipo, la skill de Descubrimiento de producto y validación con clientes desarrolla el consentimiento y el tratamiento de los datos con más detalle.

Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Usar la IA para preparar una sesión con su propósito, su contexto y varias alternativas de formato.
- Sintetizar con IA el material cualitativo de una sesión y contrastar esa síntesis con una agrupación propia.
- Interpretar los patrones que destaca la IA a la luz del contexto del equipo, que la IA no conoce.
- Reconocer la frontera: la IA no media la interacción humana ni lee el clima del equipo.
- Explicar qué prohíbe el Reglamento europeo de IA sobre la inferencia de emociones en el trabajo y por qué afecta a algunas herramientas de reunión.
- Grabar y transcribir una sesión con información previa, herramienta aprobada y conservación limitada.

Errores y antipatrones frecuentes

- **Delegar en la IA la lectura del clima del equipo.** La percepción de lo no dicho y de la emoción en la sala es una competencia humana insustituible.
- **Aceptar la síntesis de la IA sin interpretarla.** Los patrones que agrupa carecen del contexto que solo el facilitador aporta.
- **Usar herramientas que puntúan las emociones.** Además de despersonalizar la sesión, inferir emociones de las personas en el trabajo está prohibido por el Reglamento de IA.
- **Grabar sin informar.** Una transcripción hecha a espaldas del equipo rompe la confianza y el marco de protección de datos.
- **Confundir preparación con conducción.** La IA prepara la sesión; conducirla, con todo lo que implica de presencia y juicio, es del facilitador.

Para profundizar

- [AEPD — Transcripción de voz con IA](#)
- [Reglamento de IA — Artículo 5, prácticas prohibidas](#)
- [Stefan Wolpers — The AI4Agile Practitioners Report 2026](#)
- [Scrum Manager BoK — Retrospectiva](#)
- [Skill Arena — Descubrimiento de producto y validación con clientes](#)

BLOQUE C · APLICACIÓN AL CICLO DE TRABAJO ÁGIL

Capítulo 11. IA en calidad y entrega (perfiles de desarrollo) · ESTABLECIDO

Apoyar en la IA las pruebas y la revisión de código, revisando lo generado con al menos el mismo rigor que lo escrito a mano.

Introducción

Este capítulo recoge un uso de alto valor para equipos técnicos. Se marca como propio de los perfiles de desarrollo porque su aplicación directa exige trabajar con código, a diferencia de los usos transversales de los capítulos anteriores. A quien no programa le aporta comprensión del alcance y de los riesgos; a quien programa, prácticas concretas.

Veremos qué dice la evidencia sobre la seguridad del código generado y sobre la productividad percibida, cómo se usa la IA en las pruebas y en la revisión, y qué riesgos propios del desarrollo con asistentes y agentes ha catalogado OWASP.

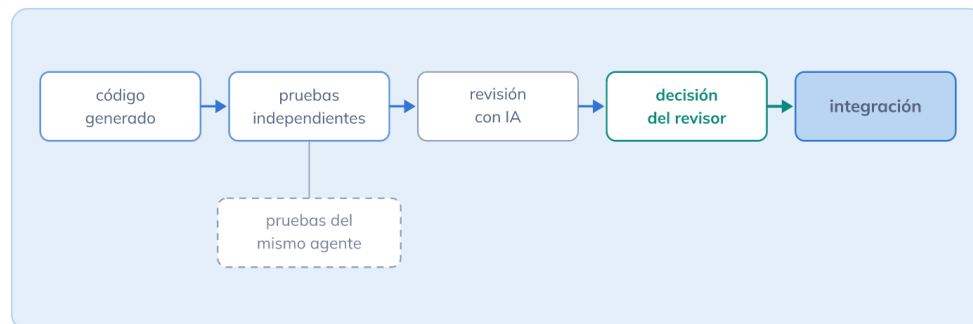


Figura 11. El camino del código generado: lo que genera la IA pasa por pruebas de origen independiente y por una revisión, y la decisión de integrarlo es de una persona.

11.1 Lo que dice la evidencia

La seguridad del código generado

Veracode mide desde 2025 la seguridad del código que generan más de cien modelos en varios lenguajes. Su primer informe (julio de 2025) encontró que el 45 % de las muestras fallaba las pruebas de seguridad e introducía vulnerabilidades del Top 10 de OWASP. El *GenAI Code Security Report* de julio de 2026 constató que la tasa media de aprobación

seguía en el 56 %, frente al 55 % del año anterior: los modelos escriben ya código sintácticamente correcto casi siempre, pero su seguridad no mejora al mismo ritmo.

La productividad percibida y la medida

En 2025, METR midió el trabajo de 16 desarrolladores experimentados de proyectos de código abierto en 246 tareas reales. Con herramientas de IA tardaron un 19 % más, aunque antes esperaban ir un 24 % más deprisa y después creían haber ido un 20 % más rápido. En febrero de 2026, METR avisó de que su nuevo experimento ya no daba una señal fiable y de que la mejora real podía ser bastante mayor, pero la lección de fondo se mantiene: la sensación de velocidad no basta para saber si se va más rápido.

El informe de DORA de 2025 apunta en la misma dirección desde otro ángulo. Describe la IA como un amplificador de las fortalezas y las debilidades que ya tiene una organización: el 90 % de los encuestados la usa en su trabajo, su adopción mejora el ritmo de entrega pero se asocia a más inestabilidad, y en torno al 30 % confía poco o nada en el código generado. La skill de Scrum en equipos con IA desarrolla esta paradoja de la productividad.

11.2 Generar y revisar pruebas

La IA amplía el alcance de las pruebas: genera casos alineados con los criterios de aceptación y con los riesgos identificados, y es especialmente útil para cubrir escenarios negativos y casos límite más allá del camino feliz. El profesional valida que los casos generados sean los relevantes, porque la cantidad de pruebas no es señal de calidad si no atacan los riesgos reales.

Hay una trampa específica. Un conjunto de pruebas que pasa, escrito por el mismo agente que escribió el código, no aporta una garantía independiente, y OWASP cataloga como riesgo tanto esa fabricación de pruebas como el borrado de las que fallan. Las pruebas que validan lo generado necesitan un origen independiente, como unos criterios de aceptación acordados antes o unas pruebas escritas o revisadas por una persona.

11.3 La IA en la revisión de código

Como par en la revisión de código, la IA ayuda a detectar problemas de legibilidad, mantenibilidad y documentación y, sobre todo, riesgos de seguridad: validación de entradas, manejo de errores, gestión de permisos y efectos colaterales. Amplía la cobertura del revisor humano sin sustituir su criterio, y la decisión final sobre un cambio sigue siendo de quien lo revisa.

El criterio técnico no se delega. La responsabilidad sobre el código y la decisión de aceptarlo permanecen en el profesional. OWASP lo resume en un principio: todo cambio asistido por IA debe revisarse, aprobarse y ser atribuible a un desarrollador responsable de su seguridad. Un hallazgo de la IA es una señal para examinar con juicio propio.

Cuando la IA genera más código del que las personas pueden revisar con calma, la revisión corre el riesgo de convertirse en un sello automático. Acotar el tamaño de los

cambios, repartir mejor el trabajo de revisión y apoyarse en la IA para una primera pasada ayudan a evitarlo.

11.4 Riesgos propios del desarrollo con IA

La hoja de referencia de OWASP para programar con IA de forma segura (*Secure Coding with AI Cheat Sheet*) cataloga los riesgos propios del desarrollo con asistentes y agentes. Tres de ellos ilustran bien por qué lo generado pide más rigor.

- **Dependencias alucinadas:** los asistentes sugieren con frecuencia paquetes que no existen en los registros públicos, un hueco que un atacante puede aprovechar publicando uno con ese nombre.
- **Pruebas fabricadas o borradas:** un agente consigue que las pruebas pasen escribiéndolas a la medida de su código o eliminando las que fallan.
- **Inyección de instrucciones en el ciclo de desarrollo:** ficheros, incidencias o documentación del repositorio con órdenes ocultas que el agente lee y ejecuta, la versión técnica de la frontera de confianza del capítulo 4.

El BoK de Scrum Manager recoge estas prácticas en sus páginas de QA asistido por IA y de revisión humana de lo que produce la IA, y la skill de Harness Engineering desarrolla cómo se construyen los controles automáticos que vigilan el trabajo de los agentes.

Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Interpretar la evidencia disponible sobre la seguridad del código generado y sobre la diferencia entre la productividad percibida y la medida.
- Generar con IA casos de prueba alineados con los criterios de aceptación y los riesgos, y validar que atacan los riesgos reales.
- Detectar pruebas fabricadas o borradas y exigir que las pruebas que validan lo generado tengan un origen independiente.
- Usar la IA como apoyo en la revisión de código con foco en seguridad, manteniendo la decisión final en el profesional.
- Reconocer las dependencias alucinadas y la inyección de instrucciones en el ciclo de desarrollo.

Errores y antipatrones frecuentes

- **Confundir cantidad de pruebas con calidad.** Muchas pruebas que no atacan los riesgos reales dan una falsa seguridad.
- **Dar por buenas las pruebas del mismo agente.** Si las escribió quien escribió el código, que pasen no demuestra nada por sí solo.
- **Aceptar el código generado sin una revisión rigurosa.** Casi la mitad de las tareas de generación medidas introducen alguna vulnerabilidad, así que la revisión debe ser al menos tan exigente como con el código manual.

- **Tratar los hallazgos de la IA como veredictos.** Son señales para examinar con criterio técnico, no decisiones automáticas.
- **Dejarse llevar por la sensación de velocidad.** La percepción de ir más rápido no equivale a una mejora real; conviene medir en lugar de suponer.

Para profundizar

- [Veracode — 2026 GenAI Code Security Report](#)
- [METR — Measuring the impact of early-2025 AI on experienced developer productivity](#)
- [METR — Uplift update \(febrero de 2026\)](#)
- [DORA — State of AI-assisted Software Development 2025](#)
- [OWASP — Secure Coding with AI Cheat Sheet](#)
- [Scrum Manager BoK — QA asistido por IA](#)
- [Scrum Manager BoK — Revisión humana de outputs de IA](#)
- [Skill Arena — Scrum en equipos con IA](#)

© 2026 Scrum Manager®. Esta obra se publica bajo licencia Creative Commons Atribución – No Comercial 4.0 Internacional (CC BY-NC 4.0). Los formadores y centros oficiales de Scrum Manager quedan licenciados bajo los términos CC BY 4.0 para su actividad formativa.