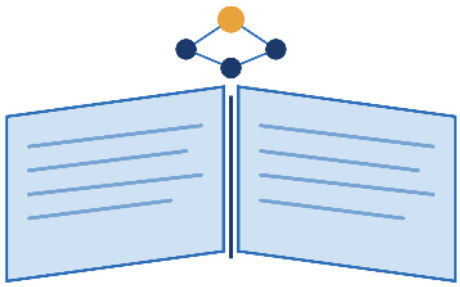
Scrum Manager
Uncovering better ways

Guía

IA aplicada al trabajo ágil

Desarrollo de los temas del State of the art



Actualizado a junio 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, 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 es un campo en plena ebullición que evoluciona de forma extremadamente rápida: prácticas y herramientas aparecen, maduran o quedan obsoletas en cuestión de meses, y solo una parte de los fundamentos puede considerarse asentada. 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.

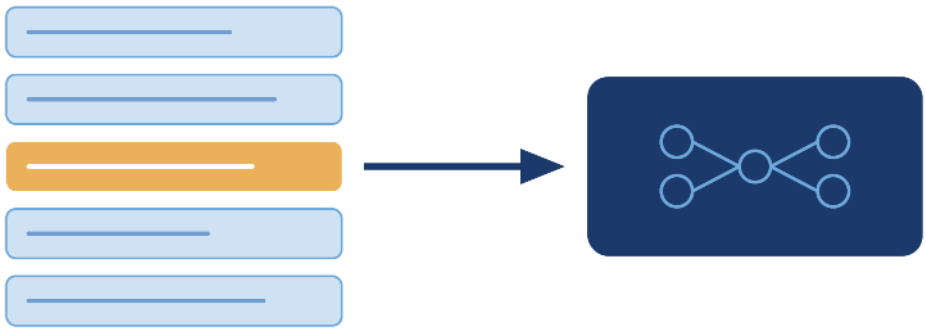
Índice

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

BLOQUE A · FUNDAMENTOS DE INTERACCIÓN CON LA IA

Capítulo 1. Ingeniería de contexto · EN CONSOLIDACIÓN

Diseñar deliberadamente qué información recibe el modelo y en qué orden, no solo cómo se redacta la instrucción puntual.



La ingeniería de contexto gobierna qué información entra en el modelo, en qué orden y qué se omite.

Introducción

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

La ingeniería de contexto es el conjunto de estrategias para decidir qué información se pone a disposición del modelo en cada momento, cómo se ordena y qué se deja fuera. El cambio de foco es sutil pero decisivo: la pregunta deja de ser «¿cómo redacto esta petición?» y pasa a ser «¿qué configuración de información hace más probable el resultado que busco?». Este capítulo desarrolla esa competencia, su relación con el prompting y las estrategias prácticas que la articulan.

Es un tema en consolidación: el marco conceptual está asentado y adoptado por los principales actores del sector, pero las técnicas concretas siguen evolucionando con rapidez. Por eso conviene dominar los principios estables —que son los que este capítulo desarrolla— y mantener una vigilancia activa sobre las herramientas, que cambian más deprisa.

1.1 De la instrucción al contexto

El prompting se ocupa de una unidad concreta: la instrucción que se escribe en un momento dado. La ingeniería de contexto se ocupa de todo lo que el modelo «ve» cuando genera una respuesta, que es bastante más que la instrucción: incluye las indicaciones del sistema, el historial de la conversación, los documentos recuperados, los resultados de herramientas que el modelo haya usado y los ejemplos que se le hayan proporcionado.

La analogía más extendida es la de la memoria de trabajo. Si se entiende el modelo como un procesador, su ventana de contexto es la memoria de trabajo: un espacio finito donde solo cabe una cantidad limitada de información, y donde lo que se incluye compite por la atención del modelo. La ingeniería de contexto es la gestión deliberada de ese espacio escaso.

Idea central. El prompting es un componente de la ingeniería de contexto, no una disciplina rival. Para tareas simples —una consulta puntual, un texto breve— un buen prompt basta. La ingeniería de contexto se vuelve imprescindible cuando el trabajo es sostenido, cuando intervienen agentes o

cuando la información relevante excede lo que cabe cómodamente en una sola instrucción.

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 práctica demuestra lo contrario, por dos razones bien documentadas.

El presupuesto de atención. Los modelos no ponderan todos los elementos por igual. Igual que una persona con memoria de trabajo limitada, priorizan la información reciente o más destacada, y los detalles importantes se diluyen entre el ruido cuando la entrada es muy larga. Añadir contexto irrelevante no es neutro: degrada la capacidad del modelo de atender a lo que importa.

La degradación en contextos largos. A medida que el contexto crece, la fiabilidad de las respuestas decae. El modelo tiende a contradecirse, a omitir instrucciones o a fabricar información. Un fenómeno relacionado es el del «lost in the middle»: la información situada al principio y al final del contexto se procesa con más fiabilidad que la enterrada en el medio. De ahí que el orden importe tanto como el contenido.

La consecuencia práctica es un principio de diseño: el objetivo no es maximizar la información, sino curarla. Incluir lo justo, ordenarlo bien y retirar lo que ha dejado de ser útil.

1.3 Las cuatro estrategias: escribir, seleccionar, comprimir, aislar

El sector ha convergido en un marco de cuatro estrategias complementarias para gestionar el contexto. No son excluyentes: la mayoría de los sistemas serios combinan las cuatro. Conocerlas da un vocabulario común para razonar sobre el problema.

1. **Escribir (write).** Guardar información fuera de la ventana de contexto para que esté disponible sin consumir espacio. El patrón típico es el «cuaderno de notas» (scratchpad): el agente registra decisiones, hallazgos o progreso en un almacén externo y los recupera cuando los necesita, en lugar de arrastrarlo todo en la conversación.
2. **Seleccionar (select).** Traer a la ventana solo la información relevante para el paso actual. Es la estrategia de la recuperación (RAG y similares): en lugar de incluirlo todo, se identifica y se aporta lo pertinente para cada interacción, mediante búsqueda semántica o recuperación dirigida.
3. **Comprimir (compress).** Retener solo los tokens necesarios para la tarea, reduciendo el tamaño sin perder significado: resumir conversaciones largas, extraer los hechos clave, condensar los resultados extensos de una herramienta antes de que entren en el contexto.
4. **Aislar (isolate).** Separar contextos distintos para que no interfieran entre sí. La forma más avanzada es la arquitectura multiagente: en vez de un único agente que lo sostiene todo en una ventana enorme, varios agentes especializados, cada uno con su contexto enfocado.

Patrón de producción habitual. Un sistema multiagente (aislar) en el que cada subagente recupera lo que necesita (seleccionar), mantiene un cuaderno de progreso (escribir) y resume las salidas largas de herramientas para que quepan en su ventana (comprimir). Las cuatro estrategias operando juntas.

1.4 La ingeniería de contexto en el trabajo ágil

Para un profesional ágil, esta competencia no es una cuestión técnica de infraestructura, sino una forma de plantear cualquier trabajo asistido por IA. Antes de pedirle algo a un modelo, la pregunta útil no es «¿qué le escribo?», sino «¿qué necesita saber para hacer esto bien, y qué le sobra?».

En la práctica cotidiana esto se traduce en hábitos concretos: aportar el contexto mínimo suficiente y no volcar documentos enteros «por si acaso»; situar la instrucción y los datos críticos en posiciones de alta atención; separar conversaciones distintas en lugar de arrastrar un único hilo interminable que mezcla temas; y, en tareas largas, apoyarse en notas externas en vez de confiar en que el modelo «recuerde» todo lo dicho. Son aplicaciones directas de las cuatro estrategias al trabajo sin código.

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 en lugar de sustituirlo.
- Reconocer cuándo una tarea requiere gestión de contexto (trabajo sostenido, agentes, información extensa) y cuándo basta con un buen prompt.
- Justificar por qué añadir más contexto puede degradar el resultado, apelando al presupuesto de atención y a la degradación en contextos largos.
- Identificar cuál de las cuatro estrategias —escribir, seleccionar, comprimir, aislar— resuelve un problema de contexto concreto.
- Aplicar el principio de contexto mínimo suficiente y el orden de la información a una petición real de su trabajo.

Errores y antipatrones frecuentes

Confundir la ingeniería de contexto con escribir prompts más largos. Alargar la instrucción no es gestionar el contexto; a menudo lo empeora, porque añade ruido. La gestión consiste en curar, no en acumular.

Volcar toda la información disponible «por si acaso». Es el error más común y el más costoso: diluye la atención del modelo y degrada la fiabilidad. La práctica correcta es la minimización deliberada.

Ignorar el orden de la información. Enterrar la instrucción crítica en mitad de un contexto largo reduce la probabilidad de que el modelo la atienda. Las indicaciones decisivas van en posiciones de alta atención.

Arrastrar un único hilo para todo. Mezclar temas y tareas distintas en una sola conversación interminable provoca interferencia y confusión. Aislar contextos distintos es, a menudo, la solución más simple y eficaz.

Para profundizar

· Anthropic — Effective context engineering for AI agents

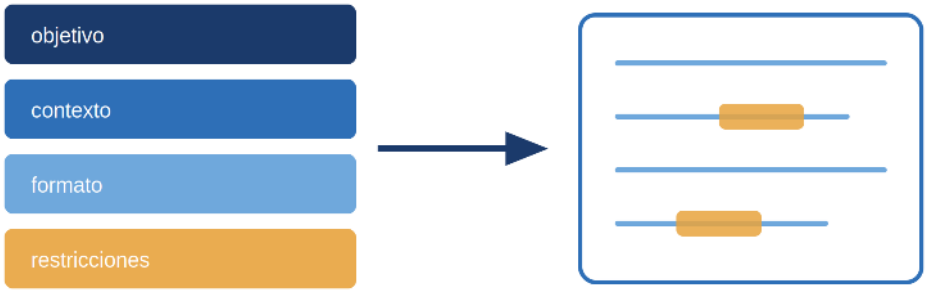
· LangChain — Context engineering for agents

· Anthropic — Documentación de prompt 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.



Una petición eficaz se compone de partes estables que se convierten en una plantilla reutilizable.

Introducción

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

Este capítulo desarrolla los componentes de un prompt eficaz, la distinción entre instrucciones que producen consistencia y las que no, y la práctica de convertir prompts útiles en plantillas reutilizables a escala de equipo.

2.1 Los componentes de una petición eficaz

Un prompt predecible se construye con cuatro componentes explícitos. Su ausencia es la causa más frecuente de resultados erráticos.

5. **Objetivo con criterios de éxito.** Qué se quiere obtener y cómo se reconocerá que la respuesta es buena. Un objetivo sin criterio observable («hazlo bien») no orienta al modelo.
6. **Contexto mínimo suficiente.** La información que el modelo necesita para la tarea, ni más ni menos. Aquí enlaza con la ingeniería de contexto: el exceso degrada tanto como el defecto.
7. **Formato de salida.** La forma esperada de la respuesta: extensión, estructura, tono, tipo de elementos. Es el componente que más consistencia aporta.
8. **Restricciones explícitas.** Los límites: qué no hacer, qué evitar, qué condiciones respetar. Las restricciones reducen el espacio de respuestas no deseadas.

2.2 Instrucciones estructurales frente a cualitativas

Una distinción operativa decisiva: las instrucciones estructurales —las que definen la forma de la respuesta— producen consistencia; las cualitativas —«sé más exhaustivo», «hazlo mejor»— no. Pedir «una lista de cinco puntos, cada uno de una frase» da un resultado reproducible; pedir «sé más completo» depende de una interpretación que varía en cada ejecución.

Regla práctica. Cuando un prompt no da el resultado esperado, el reflejo correcto no es añadir adjetivos cualitativos, sino preguntarse qué componente estructural falta. La mejora viene de especificar mejor la forma, no de insistir en la calidad.

2.3 Descomponer peticiones sobrecargadas

Un prompt que pide demasiado a la vez produce resultados de calidad desigual: el modelo atiende a unas partes y descuida otras. La práctica correcta es descomponer la petición en pasos secuenciales con prioridad explícita, o dividir una tarea compleja en varias interacciones encadenadas. Esto conecta con la estrategia de aislamiento de la ingeniería de contexto.

2.4 De prompt a plantilla reutilizable

El salto que distingue el uso profesional del ocasional es la reutilización. Un prompt eficaz no es texto desechable: es un artefacto que conviene guardar, versionar y compartir con el equipo, de modo que distintas personas obtengan resultados comparables. Tratar los prompts como activos del equipo —revisables y mantenidos como el código— es la práctica que estabiliza la calidad a escala y enlaza directamente con la metodología de especificaciones (SDD) y con los archivos de instrucciones reutilizables del ecosistema agéntico.

Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Construir un prompt con objetivo, criterios de éxito observables, contexto mínimo, formato de salida y restricciones explícitas.
- Diagnosticar por qué un prompt falla identificando qué componente estructural le falta, en lugar de añadir instrucciones cualitativas.
- Distinguir instrucciones estructurales de cualitativas y explicar por qué solo las primeras producen consistencia.
- Descomponer una petición sobrecargada en pasos secuenciales con prioridad explícita.
- Convertir un prompt eficaz en una plantilla reutilizable que dé resultados comparables entre distintas personas.

Errores y antipatrones frecuentes

Responder a un mal resultado añadiendo adjetivos. «Sé más exhaustivo» rara vez mejora; lo que falta casi siempre es un componente estructural concreto.

Pedir demasiado en una sola instrucción. La sobrecarga produce atención desigual. Descomponer es más fiable que confiar en que el modelo lo abarque todo.

Tratar los prompts como texto desechable. No guardar ni versionar los prompts eficaces condena al equipo a reinventarlos y a obtener resultados inconsistentes entre personas.

Omitir el formato de salida. Sin especificar la forma esperada, cada ejecución produce una estructura distinta, lo que impide comparar o automatizar.

Para profundizar

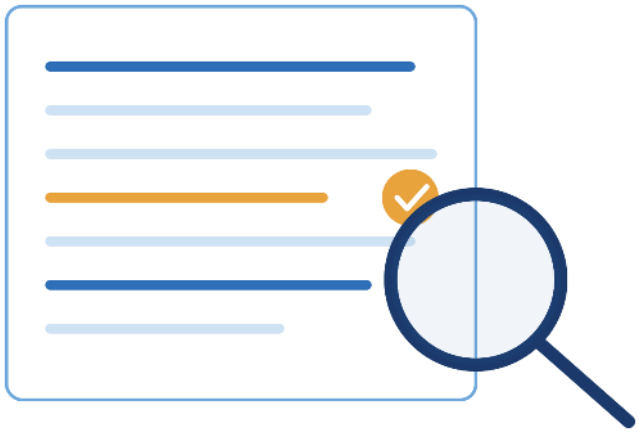
· OpenAI — Prompt engineering guide

· Anthropic — Prompt engineering overview

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.



Verificar antes de actuar: separar hechos de inferencias y detectar la precisión injustificada.

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: la fluidez no es señal de veracidad. Verificar es la práctica de someter lo generado a escrutinio antes de actuar sobre ello. Lejos de perder relevancia con el avance de los modelos, la gana: cuanto más trabajo se delega, más crítico es saber comprobar lo que el sistema entrega.

3.1 Separar hechos, inferencias y recomendaciones

Una respuesta de un modelo mezcla a menudo tres cosas de naturaleza distinta: hechos comprobables, inferencias del modelo a partir de esos hechos, y recomendaciones de acción. La primera disciplina de verificación es separarlas, porque cada una exige un nivel de comprobación diferente. Un hecho se contrasta con una fuente; una inferencia se examina en su lógica; una recomendación se evalúa contra el criterio y el contexto del profesional.

3.2 Pedir los supuestos y la evidencia

Una técnica eficaz consiste en no aceptar la conclusión sin pedir lo que la sostiene: los supuestos de partida y la evidencia. Solicitar al modelo que explicite en qué se basa, o que distinga lo que sabe de lo que infiere, revela la solidez real de la respuesta y expone los puntos débiles que la fluidez ocultaba.

3.3 La precisión injustificada

Un patrón de riesgo característico es la «precisión injustificada»: cifras concretas, porcentajes o certezas categóricas presentadas sin respaldo. La concreción produce una sensación de rigor que no siempre corresponde a la realidad. Ante un dato preciso, la pregunta de verificación es siempre la misma: ¿de dónde sale? Si no hay fuente, la precisión es decorativa y debe tratarse como sospechosa.

Principio rector. La capacidad de juicio del profesional es el valor humano irreemplazable en el trabajo con IA. La verificación no es un trámite de control de calidad: es el lugar donde el criterio

profesional aporta lo que el modelo no puede aportar por sí mismo.

3.4 El enfoque «zero trust» aplicado a la información

La postura más segura ante la salida de un modelo es la de confianza cero por defecto: no se asume válida hasta que se ha verificado en proporción al riesgo de actuar sobre ella. No significa desconfiar de todo por igual —sería paralizante— sino calibrar el esfuerzo de verificación según las consecuencias de un error: una idea para una lluvia de ideas no requiere lo mismo que un dato que irá a una decisión de negocio.

Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- 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.
- Detectar la precisión injustificada y tratar como sospechosa toda cifra o certeza sin fuente.
- Calibrar el nivel de verificación en proporción al riesgo de actuar sobre la información.
- Reconocer que la fluidez del texto no es indicio de su exactitud.

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; a menudo es más peligrosa, porque inspira confianza.

Verificar todo por igual o no verificar nada. Ambos extremos fallan: el primero paraliza, el segundo expone. La calibración por riesgo es la práctica correcta.

Delegar la verificación en el propio modelo sin criterio externo. Preguntarle al modelo si está seguro no sustituye el contraste con una fuente o con el juicio profesional.

Para profundizar

· NIST — AI Risk Management Framework

· Anthropic — Reducing hallucinations

BLOQUE A · FUNDAMENTOS DE INTERACCIÓN CON LA IA

Capítulo 4. Frontera de confianza: instrucciones, datos y privacidad ·

ESTABLECIDO

Tratar el contenido externo como no confiable y proteger la información sensible.

La frontera de confianza separa las instrucciones del profesional de los datos externos no confiables.

Introducción

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

4.1 Instrucciones frente a datos: la inyección de prompts

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

La defensa es tratar todo contenido de origen externo como dato no confiable por defecto, y separarlo explícitamente de las instrucciones legítimas. El riesgo escala con los agentes: una instrucción inyectada que antes solo producía una respuesta indebida puede ahora desencadenar una acción real no deseada sobre un sistema.

Por qué importa más con agentes. Cuando el modelo solo responde, una inyección produce, como mucho, una mala respuesta que el humano revisa. Cuando el agente ejecuta —envía correos, modifica datos, llama a herramientas—, la instrucción inyectada se convierte en una acción con consecuencias reales. La frontera de confianza pasa de higiene a seguridad crítica.

4.2 Privacidad y manejo de secretos

La segunda competencia es la protección de la información que se comparte con un modelo. El principio es de minimización: aportar solo los datos necesarios para la tarea, y anonimizar de forma efectiva lo que pueda identificar a personas u organizaciones. La anonimización efectiva no es maquillaje: sustituir un nombre pero dejar datos que permiten reidentificar no protege.

Un error específico es confiar en etiquetas o promesas: marcar un texto como «confidencial» dentro de un prompt no impide que el modelo lo procese o lo refleje, y las garantías sobre el tratamiento de

datos dependen del servicio concreto, no del deseo del usuario. La regla práctica: no compartir con un modelo nada que no se estaría dispuesto a ver fuera del control propio, salvo bajo condiciones contractuales y técnicas verificadas.

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 qué es la inyección de instrucciones.
- Tratar el contenido de origen externo como no confiable por defecto y separarlo de las instrucciones legítimas.
- Justificar por qué el riesgo de inyección escala cuando un agente ejecuta acciones en lugar de solo responder.
- Aplicar el principio de minimización y la anonimización efectiva antes de compartir información con un modelo.
- Reconocer que las etiquetas de confidencialidad y las promesas del sistema no sustituyen las garantías contractuales y técnicas.

Errores y antipatrones frecuentes

Tratar el contenido externo como instrucciones de confianza. Leer un documento o una web sin asumir que puede contener órdenes incrustadas abre la puerta a la inyección.

Anonimización aparente. Sustituir un nombre dejando datos reidentificables no protege; es falsa sensación de seguridad.

Confiar en la etiqueta «confidencial». Marcar algo como secreto dentro del prompt no cambia cómo lo trata el sistema.

Volcar datos sensibles «porque es una herramienta interna». La sensibilidad del dato no depende de la comodidad del canal; depende de las garantías verificadas de ese canal.

Para profundizar

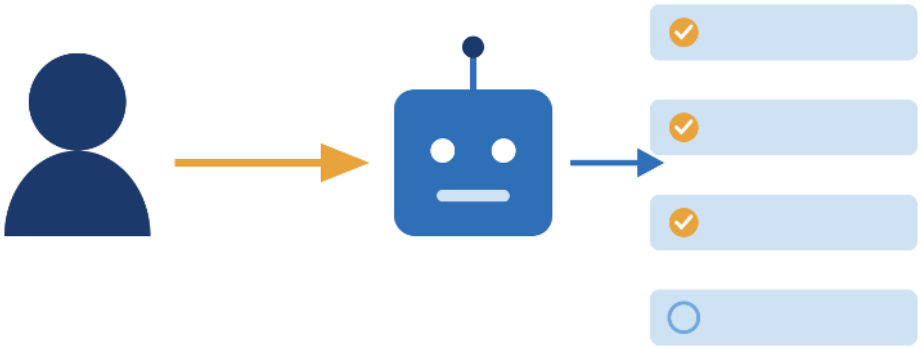
· OWASP Top 10 for LLM Applications

· Reglamento Europeo de IA (EU AI Act)

BLOQUE B · TRABAJAR CON AGENTES DE IA

Capítulo 5. Del chat al agente: delegación de trabajo verificable · EN CONSOLIDACIÓN

Pasar de usar la IA para pensar mejor a delegarle trabajo, sabiendo cuándo y con qué límites.



Del chat al agente: el profesional pasa de ejecutar a delegar trabajo con criterios de aceptación.

Introducción

El cambio más relevante de los últimos meses no es que los modelos sean más capaces, sino que han pasado de responder a ejecutar. Un asistente conversacional contesta preguntas; un agente completa tareas de varios pasos de forma autónoma: busca, decide, usa herramientas y produce un resultado. Este capítulo desarrolla qué significa ese salto para el profesional ágil y cómo se decide qué delegar.

5.1 La diferencia entre responder y ejecutar

Un asistente que responde deja siempre la acción en manos del humano: sugiere, y la persona actúa. Un agente actúa por sí mismo dentro de los límites que se le fijan. La diferencia no es de grado sino de naturaleza: cambia quién hace el trabajo y, por tanto, dónde está el riesgo y dónde el valor. El rol del profesional se desplaza de «hacer» a «decidir qué se delega, con qué límites y con qué criterios de aceptación».

5.2 Tareas verificables y tareas no delegables

No todo trabajo es delegable con la misma seguridad. La distinción útil es la verificabilidad: una tarea cuyo resultado puede comprobarse contra un criterio claro se delega bien; una tarea cuyo «bien hecho» depende de juicio tácito o de contexto no escrito es mala candidata. Conviene delegar lo que se puede verificar y reservar lo que exige criterio humano insustituible: las decisiones con implicaciones éticas, las que afectan a personas, las novedosas que caen fuera de cualquier patrón conocido.

Criterio de delegación. Antes de delegar una tarea a un agente, dos preguntas: ¿puedo verificar el resultado contra un criterio claro? y ¿las consecuencias de un error son asumibles y reversibles? Si ambas respuestas son afirmativas, es buena candidata. Si alguna falla, conviene mantener al humano en el bucle (capítulo 6).

5.3 Impacto en las ceremonias y artefactos ágiles

La delegación reconfigura el trabajo ágil. La unidad de trabajo deja de ser solo «la tarea que hago» e incluye «la tarea que orquesto y superviso». Esto afecta a la planificación (estimar trabajo

orquestado es distinto de estimar trabajo manual), al seguimiento (el progreso incluye lo que hacen los agentes) y a la definición de terminado (que debe contemplar la verificación de lo producido por IA). Saber delegar bien se convierte en una competencia de equipo, no solo individual.

Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Explicar la diferencia de naturaleza entre un asistente que responde y un agente que ejecuta.
- Decidir qué tareas son delegables a un agente según su verificabilidad y la reversibilidad de sus consecuencias.
- Identificar las tareas que exigen criterio humano insustituible y no deben delegarse.
- Reformular su rol de ejecutor a diseñador de la delegación: límites, criterios de aceptación y supervisión.
- Anticipar el impacto de la delegación en la planificación, el seguimiento y la definición de terminado del equipo.

Errores y antipatrones frecuentes

Delegar tareas no verificables. Si no se puede comprobar el resultado 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 autonomía, por capaz que sea el agente.

Tratar al agente como un ejecutor infalible. La autonomía no elimina la necesidad de verificación; la desplaza al diseño de los puntos de control.

No ajustar las prácticas de equipo. Mantener planificación, seguimiento y DoD como si todo el trabajo fuera manual ignora la parte orquestada y crea puntos ciegos.

Para profundizar

· Anthropic — Building effective agents

· Anthropic — Effective context engineering for AI agents

BLOQUE B · TRABAJAR CON AGENTES DE IA

Capítulo 6. Diseño de supervisión humana (human-in-the-loop) · EN CONSOLIDACIÓN

Decidir dónde un flujo asistido por IA debe detenerse para que una persona revise, apruebe o redirija.

La supervisión humana introduce puntos de pausa y aprobación en el flujo asistido por IA.

Introducción

Si la delegación (capítulo 5) es dar trabajo a un agente, la supervisión humana es su contrapartida necesaria: decidir dónde el flujo se detiene para que una persona intervenga. El profesional ágil diseña los puntos donde el humano participa, en lugar de ejecutar cada paso a mano. Este capítulo desarrolla los modelos de supervisión, los criterios para situar las puertas de aprobación y la dimensión de cumplimiento que los acompaña.

6.1 Tres modelos de supervisión

La supervisión no es un interruptor de todo o nada, sino un espectro con tres posiciones reconocibles que pueden coexistir en un mismo flujo:

9. **Humano en el bucle (in the loop).** El sistema se detiene y requiere aprobación humana antes de ejecutar una acción. Es el control más estricto, para decisiones de alto riesgo o irreversibles.
10. **Humano sobre el bucle (on the loop).** El humano supervisa el flujo y puede intervenir si detecta anomalías, pero no aprueba cada acción individual. Adecuado para riesgo medio.
11. **Humano fuera del bucle (out of the loop).** El agente actúa de forma autónoma; el humano revisa, si acaso, de forma diferida y agregada. Reservado para acciones de bajo riesgo y reversibles.

El nivel de supervisión es una propiedad de la decisión, no del sistema: se determina dinámicamente según el riesgo, el contexto y la política. Un mismo agente puede operar fuera del bucle para lo trivial y exigir aprobación en el bucle para lo delicado.

6.2 Dónde poner las puertas de aprobación

La pregunta práctica es qué acciones exigen aprobación humana antes de ejecutarse. Los criterios consolidados: la irreversibilidad (transacciones, borrado de datos, cambios en producción), el acceso de escritura a sistemas sensibles, la materialidad de las consecuencias para clientes o reguladores, y la novedad o sensibilidad ética de la decisión. Cuando concurre alguno de estos factores, la acción se sitúa tras una puerta de aprobación.

Supervisión nominal no es supervisión real. Poner a alguien «en el bucle» sin darle la información, la autoridad y el tiempo para influir de verdad en la decisión no es supervisión: es un trámite que aparenta control. Un revisor que aprueba sin capacidad real de criterio es peor que la ausencia de control, porque crea una falsa garantía.

6.3 La dimensión de cumplimiento

La supervisión humana ha dejado de ser solo buena práctica para convertirse en requisito legal en muchos contextos. El Reglamento Europeo de IA, en su artículo 14, exige supervisión humana efectiva para los sistemas de IA de alto riesgo, con aplicación a partir del 2 de agosto de 2026. El alcance es extraterritorial: afecta a cualquier organización cuyos sistemas se usen en la UE. La trazabilidad —que cada acción del agente sea atribuible a una autorización humana registrada— es la base probatoria de esa supervisión.

Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Distinguir los tres modelos de supervisión (en el bucle, sobre el bucle, fuera del bucle) y elegir el adecuado según el riesgo.
- Situar puertas de aprobación aplicando los criterios de irreversibilidad, acceso sensible, materialidad y novedad.
- Reconocer cuándo una supervisión es solo nominal y por qué eso es peor que su ausencia.
- Explicar la exigencia de supervisión humana del artículo 14 del Reglamento Europeo de IA y su calendario.
- Justificar la trazabilidad como base probatoria de la supervisión.

Errores y antipatrones frecuentes

Supervisión de todo o nada. Aplicar el mismo nivel de control a toda acción ignora que el riesgo varía; satura al revisor en lo trivial y lo desgasta para lo importante.

Revisor sin información, autoridad o tiempo. La supervisión nominal cumple la forma pero no el fondo, y genera una falsa garantía peligrosa.

Olvidar la trazabilidad. Sin registro de qué humano autorizó qué acción, no hay supervisión demostrable ni cumplimiento defendible.

Tratar el cumplimiento como un asunto ajeno. El alcance extraterritorial del Reglamento Europeo de IA afecta a más organizaciones de las que creen estar fuera de su ámbito.

Para profundizar

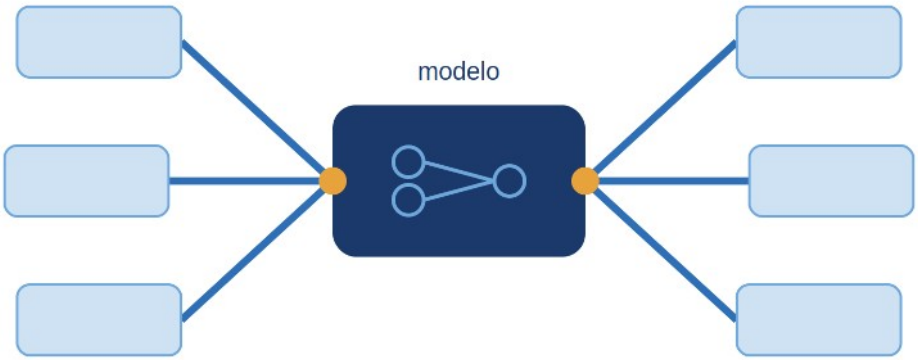
· EU AI Act — calendario de aplicación

· NIST — AI Risk Management Framework

BLOQUE B · TRABAJAR CON AGENTES DE IA

Capítulo 7. Estándares del ecosistema agéntico (MCP, AGENTS.md) · EMERGENTE

Conocer los estándares emergentes que estabilizan el trabajo con agentes.



Los estándares del ecosistema conectan el modelo con herramientas y datos mediante interfaces comunes.

Introducción

El trabajo con agentes está pasando de soluciones aisladas a un ecosistema con estándares compartidos. Para el profesional ágil no es necesario implementarlos, pero sí entender qué resuelven y por qué se están adoptando, porque marcan la dirección del campo. Este capítulo presenta los dos estándares más relevantes a fecha del documento. Es un tema emergente: lo que aquí se describe puede consolidarse o ser superado en pocos meses.

7.1 El problema que resuelven los estándares

Cada agente necesita conectarse a herramientas y datos, y recibir instrucciones sobre cómo comportarse. Sin estándares, cada conexión y cada conjunto de instrucciones se construye a medida, lo que fragmenta el ecosistema y dificulta reutilizar trabajo. Los estándares emergentes buscan un lenguaje común: que un agente pueda conectarse a una herramienta o leer sus instrucciones del mismo modo, sea cual sea el proveedor.

7.2 Model Context Protocol (MCP)

El Model Context Protocol es un estándar abierto para conectar modelos con herramientas y fuentes de datos externas de forma uniforme. Resuelve la conexión: en lugar de una integración a medida por cada herramienta, MCP ofrece una forma común de exponer datos y capacidades a un modelo. Su adopción por los principales proveedores lo ha situado como una de las piezas que estabilizan el ecosistema.

7.3 AGENTS.md

AGENTS.md es una convención para dar instrucciones reutilizables a los agentes mediante un archivo de texto estandarizado en un proyecto: qué hacer, qué evitar, qué convenciones seguir. Externaliza el prompt recurrente fuera de la conversación y lo convierte en un artefacto versionable y compartible, en la línea de tratar los prompts como activos del equipo (capítulo 2).

Contexto de adopción. MCP y AGENTS.md se han integrado en una fundación abierta del sector con respaldo de los principales proveedores de IA. Esa convergencia institucional es la señal de que el ecosistema se estabiliza, aunque su aplicación concreta sea, hoy, más relevante para perfiles

técnicos que para roles no técnicos.

7.4 Qué significa esto para el profesional ágil

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

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 el Model Context Protocol (la conexión a herramientas y datos).
- Describir qué resuelve AGENTS.md (instrucciones reutilizables y versionables para agentes).
- Interpretar la convergencia institucional en torno a estos estándares como señal de estabilización del ecosistema.
- Situar el valor de estos estándares para roles no técnicos en la interoperabilidad y la portabilidad.

Errores y antipatrones frecuentes

Creer que hay que implementarlos para conocerlos. El valor para roles no técnicos está en entender la dirección del campo, no en la implementación.

Confundir MCP y AGENTS.md. Resuelven cosas distintas: MCP conecta con herramientas y datos; AGENTS.md da instrucciones reutilizables.

Tomar el estado actual como definitivo. Es un tema emergente; conviene revisar su evolución en cada actualización del mapa.

Ignorar la portabilidad. Adoptar herramientas sin considerar la interoperabilidad puede generar dependencias difíciles de revertir.

Para profundizar

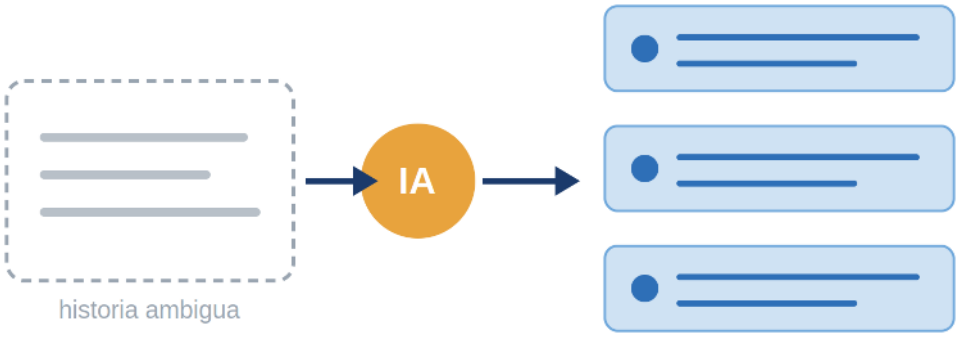
· Model Context Protocol

· AGENTS.md

BLOQUE C · APLICACIÓN AL CICLO DE TRABAJO ÁGIL

Capítulo 8. IA en el flujo de producto y backlog · ESTABLECIDO

Apoyo de la IA para refinar el backlog, redactar criterios de aceptación y plantear experimentos.



La IA reduce la ambigüedad del backlog: de la historia difusa a elementos con criterios verificables.

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 ambigüedad y retrabajo sin introducir burocracia, manteniendo siempre la decisión en el equipo. Este capítulo desarrolla las aplicaciones concretas y el límite que las hace útiles.

8.1 Refinamiento y detección de ambigüedad

La IA es eficaz detectando ambigüedad en historias de usuario y generando preguntas de refinamiento que el equipo quizá no se había planteado. No sustituye la conversación de refinamiento: la alimenta. El patrón correcto es usar las preguntas que genera como punto de partida del diálogo del equipo, no como un cuestionario que se responde mecánicamente.

8.2 Slicing con valor demostrable

Dividir historias grandes manteniendo valor en cada porción es una habilidad difícil. La IA puede proponer divisiones (slicing) y alternativas que el equipo no había considerado. El criterio humano decide cuál mantiene valor real y demostrable: la propuesta de la IA es material para la decisión, no la decisión.

8.3 Criterios de aceptación verificables

La IA ayuda a redactar criterios de aceptación más completos, incluyendo escenarios negativos y casos borde que es fácil olvidar. Un criterio de aceptación útil cubre no solo el camino feliz, sino lo que debe ocurrir cuando algo va mal. La IA es buena recordando esos casos; el equipo valida que sean los relevantes para su contexto.

8.4 Hipótesis y experimentos

En el espacio del descubrimiento, la IA apoya la formulación de hipótesis y el diseño de experimentos con métricas y guardarraíles. Ayuda a articular qué se quiere aprender, cómo se medirá y qué señales indicarían que conviene detenerse. La decisión de qué experimentar y cómo interpretar los resultados sigue siendo del equipo.

El límite que lo hace útil. En todos estos usos, la IA produce material para la conversación del

equipo, no requisitos cerrados. El momento en que las sugerencias de la IA se aceptan sin discusión es el momento en que esta 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:

- Usar la IA para detectar ambigüedad en historias y generar preguntas de refinamiento que alimenten la conversación del equipo.
- Apoyarse en la IA para proponer divisiones de historias, decidiendo con criterio cuál mantiene valor demostrable.
- Redactar con apoyo de IA criterios de aceptación que cubran escenarios negativos y casos borde.
- Formular hipótesis y diseñar experimentos con métricas y guardarraíles con asistencia de IA.
- Mantener la decisión en el equipo, tratando las salidas de la IA como punto de partida y no como requisito.

Errores y antipatrones frecuentes

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; no lo reemplazan.

Criterios de aceptación solo del camino feliz. Olvidar los escenarios negativos y casos borde produce criterios incompletos; la IA ayuda a recordarlos.

Introducir burocracia con la excusa de la IA. Si el apoyo de la IA añade pasos sin reducir ambigüedad ni retrabajo, está estorbando, no ayudando.

Para profundizar

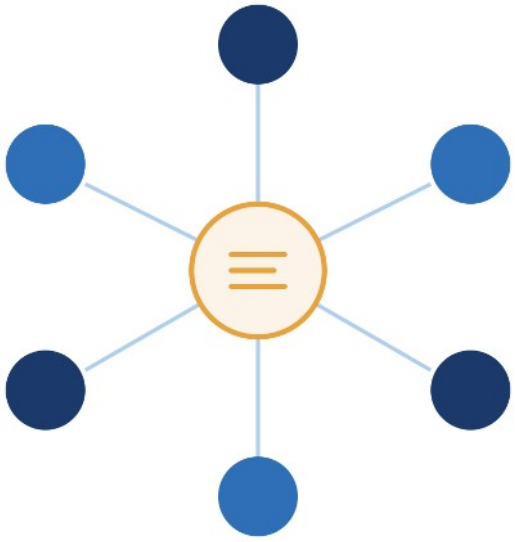
· Scrum.org — El Scrum Master y la inteligencia artificial

· Scrum Manager — Scrum en la era de la IA

BLOQUE C · APLICACIÓN AL CICLO DE TRABAJO ÁGIL

Capítulo 9. IA en facilitación y dinámicas de equipo · EN CONSOLIDACIÓN

Apoyo de la IA para preparar facilitación y analizar dinámicas, sin sustituir la interacción humana.



La IA sintetiza y prepara, sin sustituir la interacción humana que sostiene la facilitación.

Introducción

Es el uso que más se echa en falta en los materiales sobre IA y agilidad, y el 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. Este capítulo desarrolla los usos legítimos y la frontera que no conviene cruzar.

9.1 Preparación de facilitación

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

9.2 Síntesis de descubrimiento y análisis cualitativo

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 competencia del facilitador, que conoce el contexto que la IA no ve.

9.3 La frontera que no conviene cruzar

Hay un límite nítido: 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, la respuesta a la emoción en la sala —todo eso es competencia exclusiva 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 el trabajo de facilitación en sus extremos —la preparación previa y el análisis posterior— pero no en su centro, que es la interacción humana en tiempo real. Ese centro

es, precisamente, donde reside el valor profesional del facilitador y lo que la IA no puede aportar.

Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Usar la IA para preparar sesiones: formatos, preguntas, estructura de dinámicas, abordaje de conversaciones difíciles.
- Apoyarse en la IA para sintetizar y analizar cualitativamente el material de descubrimiento y retrospectivas.
- Interpretar los patrones que la IA destaca 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.
- Situar el valor profesional del facilitador en la interacción en tiempo real, no en la preparación ni el análisis.

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 competencia humana insustituible.

Aceptar la síntesis de la IA sin interpretarla. Los patrones que la IA agrupa carecen del contexto que solo el facilitador aporta.

Despersonalizar la facilitación. Usar la IA para mediar la relación entre personas vacía la dinámica de su propósito.

Confundir preparación con conducción. La IA prepara la sesión; conducirla, con todo lo que implica de presencia y juicio, es del facilitador.

Para profundizar

· Scrum Manager — Scrum en la era de la IA

· Anthropic — Building effective agents

BLOQUE C · APLICACIÓN AL CICLO DE TRABAJO ÁGIL

Capítulo 10. IA en calidad y entrega — perfiles de desarrollo · ESTABLECIDO

Apoyo de la IA en pruebas y revisión de código, manteniendo el criterio técnico del profesional.

La IA amplía la cobertura de pruebas y la revisión, incluidos los casos borde más allá del happy path.

Introducción

Este capítulo recoge un uso de alto valor para equipos técnicos. Se marca explícitamente como aplicable a perfiles de desarrollo porque su aplicación directa requiere trabajar con código, a diferencia de los usos transversales de los capítulos anteriores. Para quienes no programan, el capítulo aporta comprensión del alcance; para quienes lo hacen, prácticas concretas.

10.1 Generación y revisión de pruebas

La IA amplía el alcance de las pruebas: genera casos alineados con criterios de aceptación y con los riesgos identificados, y es especialmente útil para asegurar la cobertura de escenarios negativos y casos borde más allá del «camino feliz». El profesional valida que los casos generados sean los relevantes y que cubran lo que de verdad importa; la cantidad de pruebas no es señal de calidad si no atacan los riesgos reales.

10.2 Apoyo en la revisión de código

Como par en la revisión de código, la IA ayuda a detectar problemas de legibilidad, mantenibilidad, documentación y, sobre todo, riesgos de seguridad: validación de entradas, manejo de errores, gestión de permisos, efectos colaterales. Es un apoyo que amplía la cobertura del revisor humano, no un sustituto de su criterio. La decisión final sobre un cambio sigue siendo del profesional técnico.

El criterio técnico no se delega. La IA en calidad amplía lo que un revisor puede abarcar, pero la responsabilidad sobre el código y la decisión de aceptarlo permanecen en el profesional. Un hallazgo de la IA es una señal para examinar, no un veredicto que aplicar sin juicio.

10.3 El riesgo específico del código generado

Conviene recordar el contexto que da urgencia a esta competencia: una proporción significativa del código generado por IA introduce vulnerabilidades, y la sensación de velocidad no siempre corresponde a una mejora real de productividad. Por eso la verificación (capítulo 3) y la frontera de confianza (capítulo 4) aplican con especial fuerza al código: lo generado se revisa con el mismo o mayor rigor que lo escrito a mano.

Lo que debes saber hacer

Al dominar este tema, un profesional es capaz de:

- Generar con IA casos de prueba alineados con criterios de aceptación y riesgos, cubriendo escenarios negativos y casos borde.
- Validar que las pruebas generadas atacan los riesgos reales, sin confundir cantidad con calidad.
- Usar la IA como apoyo en revisión de código con foco en seguridad: entradas, errores, permisos, efectos colaterales.
- Mantener el criterio técnico y la decisión final sobre el código en el profesional, no en la IA.
- Aplicar verificación y frontera de confianza con especial rigor al código generado por IA.

Errores y antipatrones frecuentes

Confundir cantidad de pruebas con calidad. Muchas pruebas que no atacan los riesgos reales dan falsa seguridad.

Aceptar el código generado sin revisión rigurosa. Una proporción relevante introduce vulnerabilidades; la revisión debe ser igual o más exigente que para 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, no suponer.

Para profundizar

· OWASP Top 10 for LLM Applications

· Anthropic — Building effective agents