Título del libro

“Gobernar la inteligencia”

Cómo delegar decisiones sin perder el control

Edición: Septiembre 2026 – Versión 1

ISBN: 9798170791354 / 9798170794614

ASIN: B0HH7BGTSZ

Publicación independiente a través de Amazon KDP.

Diseño y maquetación: Milton Chanes

Portada: Milton Chanes

Impresa y distribuida por Amazon.

«El papel de los humanos como principal factor de producción está condenado a disminuir igual que disminuyó el de los caballos en la agricultura, primero reducidos y luego eliminados por el tractor.»

— 1983, Wassily Leontief

Gobernar la inteligencia

Gobernanza, seguridad y responsabilidad en la era de la inteligencia artificial

Ensayo sobre tecnología

Milton Chanes

Prólogo

Durante buena parte de la historia de la tecnología, el progreso estuvo asociado a nuestra capacidad para hacer más. Construimos máquinas que multiplicaron la fuerza física, sistemas de transporte que redujeron las distancias, ordenadores capaces de realizar cálculos a una velocidad imposible para una persona y redes que terminaron conectando prácticamente cualquier lugar del planeta. La inteligencia artificial pertenece a esa misma historia y, al mismo tiempo, introduce una diferencia considerable. Empieza a intervenir no sólo en la ejecución de nuestras decisiones, sino también en el proceso mediante el cual interpretamos información, elegimos alternativas y determinamos qué hacer a continuación.

En los primeros años de la inteligencia artificial generativa, nuestra atención estuvo concentrada principalmente en descubrir hasta dónde podía llegar la tecnología. Cada nueva generación de modelos parecía superar una frontera anterior y abría posibilidades que poco tiempo antes habrían resultado difíciles de imaginar. Aprendimos a conversar con máquinas capaces de escribir, programar, traducir, analizar documentos, interpretar imágenes y resolver

problemas cada vez más complejos. Esa evolución continuará y seguramente todavía veremos capacidades que hoy resultan difíciles de anticipar.

Sin embargo, existe una diferencia fundamental entre descubrir una capacidad y decidir cómo queremos utilizarla. Esa diferencia se vuelve especialmente importante cuando los sistemas dejan de limitarse a proporcionar información y comienzan a participar en decisiones, utilizar herramientas o ejecutar acciones que producen consecuencias. En ese momento ya no basta con preguntarnos qué puede hacer la inteligencia artificial. Necesitamos decidir qué queremos permitirle hacer, bajo qué condiciones y quién continúa siendo responsable cuando una parte del proceso deja de estar directamente en manos de una persona.

Es precisamente ahí donde comienza la gobernanza.

Gobernar la inteligencia artificial no significa detener su desarrollo ni contemplarla permanentemente como una amenaza. Tampoco significa someter cada innovación a una estructura burocrática incapaz de seguir el ritmo de la tecnología. Significa construir un marco que permita decidir conscientemente qué funciones queremos delegar, qué riesgos estamos dispuestos a asumir, qué decisiones requieren supervisión y qué mecanismos deben existir para comprender, revisar o corregir el comportamiento de los sistemas que incorporamos a nuestras organizaciones.

A medio plazo, esta cuestión será principalmente organizativa. Las empresas dejarán de utilizar una única inteligencia artificial y convivirán con numerosos modelos, copilotos, agentes, automatizaciones y servicios especializados. Algunos tendrán una función modesta, mientras que otros participarán en procesos financieros, jurídicos, comerciales, técnicos o administrativos. Muchos serán creados para resolver necesidades muy concretas y desaparecerán después. La facilidad con la que podrán construirse hará que su número crezca mucho más deprisa que nuestra capacidad tradicional para analizarlos uno por uno.

La gobernanza tendrá entonces que convertirse en una práctica cotidiana. No bastará con aprobar una tecnología en el momento en que entra en la organización. Será necesario comprender por qué existe cada sistema, qué función cumple, qué información utiliza, qué autoridad recibe y bajo qué circunstancias debe revisarse, modificarse o retirarse. La dificultad no consistirá en impedir la innovación, sino en permitir que esa innovación se distribuya sin que la organización pierda la capacidad de comprender lo que está ocurriendo dentro de ella.

A largo plazo, la cuestión será todavía más profunda. Una parte creciente de nuestra relación con empresas, administraciones, sistemas sanitarios, instituciones educativas y servicios financieros podrá estar mediada por sistemas capaces de organizar información, establecer prioridades, formular recomendaciones o participar directamente en

determinadas decisiones. No será necesario que sustituyan completamente a las personas para ejercer una influencia considerable. Bastará con que ocupen una posición intermedia entre la información disponible y quien finalmente debe decidir.

Cuando eso ocurra, conservar la capacidad de comprender por qué se utilizó determinada información, qué papel desempeñó un sistema, qué límites tenía y quién podía intervenir será algo más que una preocupación técnica. Formará parte de la manera en que distribuimos responsabilidad, confianza y poder dentro de nuestras instituciones.

Existe, además, una dificultad inevitable al escribir un libro como éste. La inteligencia artificial avanza a una velocidad que ningún libro impreso puede seguir. Mientras estas páginas se escriben aparecen nuevos modelos, cambian los productos, se incorporan funciones, se modifican nombres comerciales y algunas de las tecnologías que hoy parecen fundamentales pueden ser sustituidas en pocos años o incluso en pocos meses. También cambian las regulaciones, las interpretaciones jurídicas y los estándares que intentan acompañar esa evolución.

He intentado que los datos, referencias, tecnologías y marcos mencionados en estas páginas estén actualizados en el momento de preparar la edición. Sin embargo, sería poco realista prometer al lector que todo continuará exactamente

igual después de la impresión. Este libro debería entenderse, por tanto, de una manera diferente. Las herramientas cambiarán, pero gran parte de los problemas que intentamos resolver permanecerán.

Seguirá siendo necesario saber qué sistemas existen dentro de una organización. Seguirá siendo necesario decidir a qué información pueden acceder. Habrá que determinar qué acciones pueden realizar, qué permisos necesitan, qué riesgos son aceptables, cuándo debe intervenir una persona, qué evidencia debe conservarse y quién responde por las decisiones que la organización haya decidido delegar. Los nombres de las plataformas podrán cambiar, pero esas preguntas seguirán estando presentes.

Por esa razón, el lector debería continuar investigando después de cerrar este libro. Será especialmente importante seguir la evolución de la regulación aplicable a su país y sector, los marcos de gestión de riesgos y los estándares internacionales, las nuevas arquitecturas de agentes, los sistemas de identidad y control de acceso, la seguridad frente a ataques como la inyección de instrucciones, la protección y clasificación de los datos, los mecanismos de evaluación, la observabilidad y la trazabilidad de las acciones. También convendrá seguir atentamente la evolución de los proveedores y plataformas que la propia organización utilice, porque las funciones, permisos y condiciones de un sistema pueden cambiar mucho más deprisa que las políticas internas construidas alrededor de él.

El objetivo, sin embargo, no es convertir al lector en alguien obligado a perseguir cada novedad tecnológica. Precisamente lo contrario. Comprender los fundamentos de la gobernanza permite distinguir entre aquello que constituye un cambio superficial y aquello que modifica realmente el riesgo de un sistema. Un nuevo nombre comercial puede no significar nada importante. Una nueva capacidad para actuar sobre información sensible puede cambiar por completo la forma en que ese sistema debería ser gobernado.

Ésa es una de las ideas centrales de este libro. No existe una estrategia universal de gobernanza que pueda copiarse y aplicarse sin cambios a cualquier organización. Una pequeña empresa, un banco, un hospital, una administración pública o una compañía de software tendrán necesidades, obligaciones y niveles de riesgo diferentes. Lo importante es comprender el proceso mediante el cual se toman esas decisiones para poder construir una estrategia adaptada a cada realidad.

Gobernar significa aprender a delegar de forma consciente. Significa comprender qué puede entregarse a una máquina, qué debería permanecer bajo decisión humana y qué condiciones deben existir entre ambos extremos. Si el lector comprende ese proceso, podrá enfrentarse a tecnologías que todavía no existen, evaluar nuevas herramientas y adaptar los principios de estas páginas a la empresa en la que trabaje sin depender de que un producto determinado siga existiendo.

Ésa es también la razón por la que este libro presta tanta atención a conceptos como identidad, permisos, seguridad, observabilidad, evaluación, responsabilidad y autonomía. No son características de una plataforma concreta, sino distintas formas de responder a una misma necesidad. Saber qué hemos delegado, a quién, con qué autoridad y bajo qué límites.

Las tecnologías descritas en estas páginas cambiarán. Algunas desaparecerán, otras evolucionarán y aparecerán nuevas herramientas que resolverán problemas que hoy todavía estamos intentando comprender. La gobernanza tendrá que adaptarse a ellas, pero sus preguntas fundamentales serán mucho más persistentes.

Durante los últimos años hemos dedicado una enorme cantidad de esfuerzo a descubrir hasta dónde puede llegar la inteligencia artificial. Los próximos exigirán algo diferente y probablemente más difícil. Tendremos que aprender a decidir hasta dónde queremos que llegue, en qué circunstancias queremos utilizarla y qué responsabilidad estamos dispuestos a conservar sobre aquello que delegamos.

Si al terminar este libro el lector no recuerda el nombre de todas las herramientas mencionadas, no será un problema. Si, en cambio, ha aprendido a mirar cualquier sistema de inteligencia artificial y preguntarse qué función desempeña, qué información utiliza, qué autoridad posee, qué riesgos introduce y cómo debería ser gobernado dentro de su organización, entonces este libro habrá cumplido su propósito.

I

El problema

1

Gobernar la inteligencia cuando empieza a actuar

La inteligencia artificial ya no se limita a generar respuestas. También busca información, consulta sistemas, escribe código y utiliza herramientas. El desafío central consiste en decidir qué autoridad le concedemos, cómo comprobamos su comportamiento y quién responde cuando algo sale mal.

1.1. De la pregunta técnica a la pregunta de autoridad

Durante los primeros años de la inteligencia artificial generativa, la conversación sobre gobernanza parecía relativamente manejable. Había que reducir respuestas

discriminatorias, proteger los datos, explicar las limitaciones del modelo y fijar algunas reglas de uso. Todas esas tareas siguen siendo necesarias, pero ya no describen el problema completo.

Los sistemas actuales trabajan con documentos internos, consultan bases de datos, ejecutan código y encadenan acciones hasta completar un objetivo. Cuando una aplicación puede actuar con autonomía creciente, alguien tiene que decidir qué le está permitido, cuál es el ***Apetito de riesgo***[1] de la organización y qué consecuencias deben quedar fuera de su alcance.

Gobernar la inteligencia artificial significa convertir principios como seguridad, privacidad, transparencia y responsabilidad en decisiones aplicables. La organización debe saber qué sistemas utiliza, para qué sirven, qué datos reciben, quién accede a ellos, qué acciones pueden realizar, qué controles necesitan, cuándo debe intervenir una persona y quién tiene autoridad para detenerlos.

"Antes de iniciar un proyecto no basta con preguntar si podemos construirlo. También debemos saber si podemos gobernarlo".

[1] *Apetito de riesgo:* Cantidad y tipo de riesgo que una organización decide aceptar de forma consciente.

1.2. La unidad de gobernanza ya no es el modelo

Durante años, la inteligencia artificial responsable se explicó mediante principios generales como equidad, transparencia, seguridad, privacidad, supervisión humana y responsabilidad. Todos siguen siendo importantes, pero una organización no demuestra que los cumple por haberlos incluido en una política.

Cada principio necesita llegar hasta un criterio que alguien aplica en un momento concreto. La seguridad puede convertirse en permisos mínimos y pruebas antes del despliegue. La privacidad puede materializarse en una selección limitada de datos y un plazo de conservación. La transparencia puede exigir que el usuario sepa cuándo interviene un sistema y pueda entender el papel que desempeñó. La responsabilidad necesita un propietario con autoridad para aceptar el riesgo y detener la operación.

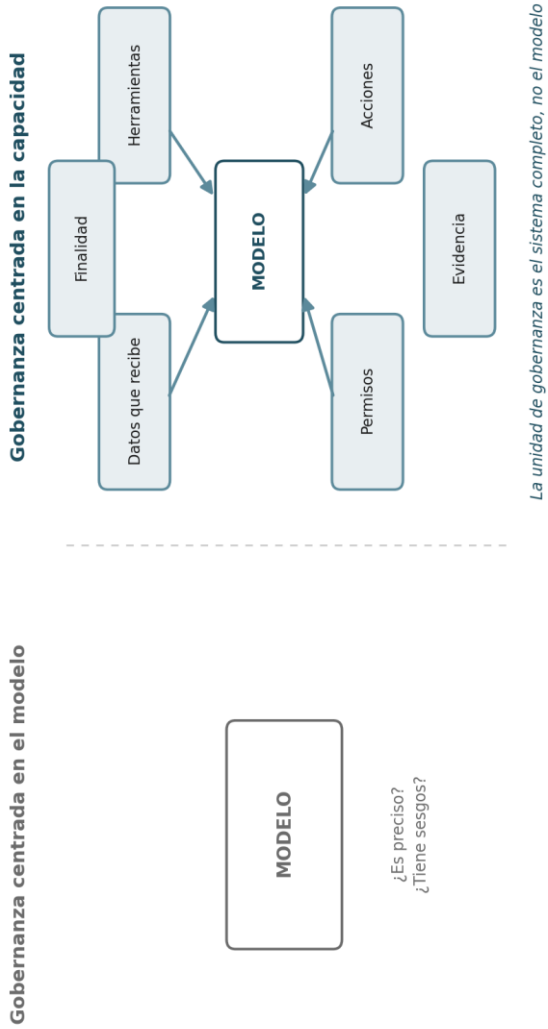

Figura 1.1 — El perímetro de gobernanza se desplaza del modelo a la capacidad completa construida a su alrededor.

***Ejemplo* - El agente de facturas**
Una empresa incorpora un agente que analiza facturas, comprueba contratos y prepara pagos. Saber qué modelo utiliza importa, aunque no permite gobernarlo por sí solo.

- → *¿Qué documentos puede leer?*
- → *¿Qué cuentas puede consultar?*
- → *¿Qué proveedores están autorizados?*
- → *¿Hasta qué importe puede preparar un pago?*
- → *¿Puede ejecutarlo o solo dejarlo preparado?*
- → *¿Quién lo aprueba?*
- → *¿Qué ocurre si se equivoca?*
- → *¿Qué registro queda de cada operación?*

Ninguna de esas ocho preguntas es sobre el modelo. Todas son sobre la capacidad que la organización ha construido a su alrededor — y son exactamente las que determinan si un error acaba en una anotación corregible o en una transferencia irreversible.

1.3. Los principios solo gobiernan cuando se convierten en controles

Durante años, la inteligencia artificial responsable se explicó mediante principios generales como equidad, transparencia, seguridad, privacidad, supervisión humana y responsabilidad. Todos siguen siendo importantes, pero una organización no demuestra que los cumple por haberlos incluido en una política.

Cada principio necesita llegar hasta un criterio que alguien aplica en un momento concreto. La seguridad puede convertirse en permisos mínimos y pruebas antes del despliegue. La privacidad puede materializarse en una selección limitada de datos y un plazo de conservación. La transparencia puede exigir que el usuario sepa cuándo interviene un sistema y pueda entender el papel que desempeñó. La responsabilidad necesita un propietario con autoridad para aceptar el riesgo y detener la operación.

La prueba de realidad

Una política gobierna cuando podemos demostrar cómo cambia el diseño, el despliegue o la operación del sistema. Si no existe ese punto de aplicación, el principio sigue siendo una intención.

1.4. La transparencia útil no exige explicar cada parámetro

La transparencia necesita una definición más precisa que saber cómo piensa el modelo. Un sistema con miles de millones de parámetros no siempre admite una explicación intuitiva de su funcionamiento interno, y una persona responsable no necesita reconstruir matemáticamente cada operación para ejercer una supervisión real.

La información exigible es más concreta y suele estar al alcance de cualquier organización bien administrada.

- → Qué sistema se utiliza, qué versión está desplegada y quién lo proporciona.
- → Con qué finalidad opera, qué datos recibe y durante cuánto tiempo los conserva.
- → Qué evaluaciones ha superado y qué limitaciones conocidas afectan a su uso.
- → Qué herramientas utiliza, qué permisos tiene y qué decisiones puede influir o ejecutar.
- → Quién consume su resultado, quién puede intervenir y qué evidencia queda de cada operación.

Los proveedores publican parte de esta información mediante fichas de modelos, informes de sistemas y documentación de seguridad. Cada uno de estos documentos puede funcionar como un ***Artefacto de transparencia***[2], porque reúne información sobre capacidades, evaluaciones, limitaciones y salvaguardas del sistema. Son insumos útiles, aunque no constituyen garantías. Documentar un riesgo no equivale a controlarlo, y la transparencia del proveedor no sustituye la responsabilidad de quien incorpora el sistema a un proceso real.

[2] *Artefacto de transparencia:* Documento que describe capacidades, evaluaciones, limitaciones y salvaguardas de un modelo o sistema.

1.5. La autonomía se gobierna como una escala

Hablar de sistemas autónomos puede sugerir una elección binaria entre control humano absoluto y libertad completa de la máquina. En la práctica, la autonomía es gradual. Tratarla como una escala permite aumentar los controles a medida que crecen la autoridad delegada, la dificultad de revertir una acción y el daño que puede producirse antes de que alguien intervenga.

Un sistema que consulta y resume información necesita controles distintos de otro que ejecuta pagos dentro de límites o planifica una secuencia completa de acciones.

Así mismo, cuando un sistema puede ejecutar acciones, resulta especialmente importante aplicar el principio de *Mínimo privilegio*[3] que concederle únicamente los permisos que necesita para realizar su función y mantenerlos solo durante el tiempo necesario.

En todos los niveles deben responderse las mismas preguntas sobre quién autorizó la capacidad, quién puede detenerla y qué evidencia permitirá explicar lo ocurrido.

[3] *Mínimo privilegio:* Principio por el que una identidad recibe solo los permisos necesarios durante el tiempo necesario.

La automatización introduce escala porque un error puede repetirse miles de veces. Los agentes añaden velocidad y encadenamiento, ya que deciden qué paso realizar después y continúan trabajando sin que una persona determine manualmente cada movimiento. Por eso la pregunta útil deja de ser si el modelo es bueno y pasa a ser qué consecuencias puede producir el sistema antes de que alguien pueda detenerlo.

Nivel	Capacidad	Control dominante
Consultar	Recupera y resume información	Acceso limitado, procedencia y protección de datos
Recomendar	Propone una decisión	Calidad, explicación, revisión y posibilidad de impugnar
Preparar	Deja una acción lista sin ejecutarla	Aprobación ligada a la operación concreta y con caducidad
Ejecutar acotado	Actúa dentro de límites explícitos	Permisos mínimos, umbrales, registros y parada automática
Actuar ampliado	Planifica y encadena acciones	Supervisión continua, aislamiento, pruebas adversariales y retirada inmediata

Figura 1.2 El control exigible crece con la autoridad delegada.

1.6. La supervisión humana es una capacidad organizativa

La expresión *human in the loop* se ha convertido en una fórmula tranquilizadora que a menudo no compromete a nada. Colocar a una persona al final del proceso no crea supervisión si esa persona carece de contexto, tiempo o autoridad para cambiar la decisión.

Una cola de cientos de aprobaciones diarias puede producir *sesgo de automatización*[4], porque la recomendación termina aceptándose por reflejo. El problema empeora cuando la interfaz oculta la incertidumbre o cuando rechazar la propuesta exige más trabajo que aprobarla. La organización conserva entonces una apariencia documental de control sin que haya existido una revisión sustantiva.

La supervisión efectiva necesita personas designadas, formación vinculada a su función, información comprensible, tiempo suficiente y autoridad real para modificar, escalar o detener. También requiere que la aprobación se refiera a una acción concreta. Autorizar una operación no debería permitir que el sistema ejecute después otra parecida en un contexto distinto.

---

[4] *Sesgo de automatización:* Tendencia a aceptar una recomendación automatizada sin el escrutinio que recibiría una propuesta humana.

1.7. Gobernar el ciclo de vida y no solamente el lanzamiento

Muchas organizaciones concentran la gobernanza en la aprobación previa al despliegue. Sin embargo, los sistemas cambian después de ese momento. Puede actualizarse el modelo, añadirse una herramienta, incorporarse una nueva fuente de datos o aparecer un uso que nadie había previsto. Cada cambio material puede modificar el riesgo aunque el nombre del producto permanezca igual.

El Marco de Gestión de Riesgos de Inteligencia Artificial del NIST, *NIST AI RMF*[5], ofrece una estructura útil basada en cuatro funciones que se alimentan entre sí. Gobernar fija responsabilidades y criterios, mapear comprende el contexto, medir produce evidencia y gestionar decide qué hacer con el riesgo. Es un marco voluntario para organizar el trabajo y no una certificación ni una prueba automática de cumplimiento.

El Reglamento europeo de Inteligencia Artificial refuerza esta transición hacia una gobernanza demostrable y se encuentra en su fase general de aplicación desde el 2 de agosto de 2026. Su calendario y sus obligaciones dependen del tipo de sistema y del papel de cada actor, por lo que el mapa jurídico

[5] *NIST AI RMF:* Marco voluntario de gestión de riesgos organizado en gobernar, mapear, medir y gestionar.

detallado se desarrolla en el capítulo siguiente y deberá verificarse antes de cada edición.

Un ciclo de vida gobernado comienza antes de construir y termina con la retirada. Incluye criterios de entrada, evaluaciones, aprobación, seguimiento de cambios, respuesta a incidentes y eliminación de permisos cuando el sistema deja de utilizarse. Un agente abandonado puede conservar acceso a información y herramientas mucho después de que nadie recuerde quién lo creó.

1.8. La pregunta que revela si existe gobernanza

Hay una forma rápida de averiguar si una organización gobierna realmente una aplicación de inteligencia artificial. Basta con preguntar quién puede detenerla y bajo qué condiciones debe hacerlo.

- → Si nadie conoce la respuesta, falta una responsabilidad esencial.
- → Si solamente el proveedor puede intervenir, existe una dependencia que debe hacerse explícita.
- → Si detener el sistema paraliza el proceso y no hay alternativa probada, existe un problema de resiliencia.
- → Si las personas responsables desconocen las señales de suspensión, falta un procedimiento aplicable.

Una organización madura define por adelantado las Condiciones de parada[6], es decir, las señales que activan la revisión, suspensión o retirada de un sistema. Entre ellas pueden figurar un incidente confirmado, un cambio de modelo sin reevaluación, una discrepancia económica no explicada, una vulnerabilidad crítica o la pérdida del responsable de negocio.

Cada condición necesita un umbral, una persona responsable, una acción por defecto y un plan de continuidad que se haya probado realmente.

1.9. Todo empieza por un inventario de IA útil

Antes de crear comités y paneles conviene responder una pregunta más elemental. La organización debe saber qué inteligencia artificial utiliza. En la práctica, un equipo puede consumir una API, otro probar un copiloto, marketing conectar un modelo a documentos internos y recursos humanos contratar una herramienta que incorpora IA sin presentarse como tal. Ninguna política gobierna aquello que permanece invisible.

[6] *Condición de parada:* Señal predefinida que obliga a revisar, suspender o retirar un sistema.

El *inventario de IA*[7] no necesita describir cada detalle técnico. Debe reunir lo mínimo que permite decidir y mantener la relación entre el sistema y su uso real.

Si alguno de estos campos no puede completarse, la ausencia ya constituye un hallazgo de gobernanza. El capítulo dedicado al inventario explicará cómo mantenerlo vivo cuando los agentes empiecen a multiplicarse y aparezcan dentro de servicios que la organización ya utilizaba.

Campo	Pregunta que debe responder
Finalidad	Qué problema resuelve y qué usos quedan fuera
Responsables	Quién es propietario del proceso y quién mantiene la solución
Datos	Qué categorías recibe, con qué fundamento y durante cuánto tiempo
Autoridad	Qué recomienda, prepara o ejecuta y con qué límites
Dependencias	Qué modelos, proveedores, herramientas y repositorios utiliza
Riesgo	Qué impacto puede producir y qué evaluación lo justifica
Ciclo de vida	Cuándo se revisó, qué cambios obligan a reevaluar y cómo se retira

[7] *Inventario de IA:* Registro de sistemas de IA con finalidad, responsables, datos, autoridad, dependencias, riesgo y ciclo de vida.

1.10. Gobernar acelera la innovación

La gobernanza suele presentarse como una fuerza que compite con la innovación, aunque una organización sin reglas solo experimenta con rapidez durante un tiempo limitado. Después, cada equipo descubre por separado los mismos problemas de privacidad, seguridad, evaluación, contratación y aprobación. Los proyectos vuelven a empezar desde cero y el conocimiento no se acumula.

Una organización con buena gobernanza dispone de modelos y proveedores evaluados, patrones de arquitectura seguros, criterios de riesgo, plantillas reutilizables y rutas de aprobación conocidas. La primera inversión requiere trabajo, pero reduce la fricción posterior y permite que los equipos avancen sin improvisar los mismos controles en cada proyecto.

Gobernar no significa que toda aplicación reciba el procedimiento más pesado. El control debe ser proporcional a la autoridad, el impacto y la dificultad de revertir una decisión. Un asistente que resume documentación pública no necesita el mismo tratamiento que un sistema que preselecciona candidatos, concede crédito o modifica una cuenta de cliente.

1.11. La inteligencia que actúa necesita una autoridad que responda

La inteligencia artificial ya forma parte del software, la documentación, la atención al cliente, el desarrollo, el análisis y las operaciones. Los agentes añaden una capa nueva porque eligen herramientas y continúan trabajando sin que una persona determine manualmente cada paso.

Esto vuelve la gobernanza más concreta. Hay que decidir qué sistema accede a qué información, qué puede recomendar, qué acciones puede preparar, cuáles puede ejecutar, qué límites no debe superar, quién puede intervenir, qué evidencia queda y quién responde por las consecuencias.

La inteligencia artificial puede ser extraordinariamente útil sin tener acceso a todo. Puede acelerar un proceso sin tomar la decisión final, preparar una acción sin ejecutarla y actuar dentro de límites que no controla por sí misma. Gobernar la inteligencia consiste en diseñar esos límites antes de necesitarlos.

Durante años preguntamos hasta dónde podía llegar la inteligencia artificial. La pregunta que abre esta obra es distinta y más importante, porque nos obliga a decidir hasta dónde queremos permitirle llegar y qué responsabilidad estamos dispuestos a conservar.

Notas y referencias

Las referencias se comprobaron el 2 de septiembre de 2026 así como los marcos normativos, los estándares y la documentación técnica.

→ **Comisión Europea AI Act**. *Síntesis oficial del Reglamento y de su aplicación.*

https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai

→ **EUR-Lex Reglamento (UE) 2024/1689.** *Texto oficial del Reglamento europeo de Inteligencia Artificial.*

https://eur-lex.europa.eu/eli/reg/2024/1689/oj

→ **NIST AI Risk Management Framework**. *Marco voluntario para incorporar consideraciones de confianza y riesgo al diseño, uso y evaluación de sistemas de IA.*

https://www.nist.gov/itl/ai-risk-management-framework

→ **OWASP Prompt Injection**. *Referencia técnica sobre inyección directa e indirecta y límites de las medidas de prevención.*

https://genai.owasp.org/llmrisk/llm01-prompt-injection/

Nota: Enlaces comprobados el 2 de Septiembre de 2026

2

¿Quién gobierna todo esto?

Cómo convertir principios en decisiones, controles y evidencia

El capítulo anterior definió qué significa gobernar una inteligencia artificial y por qué la autoridad delegada cambia la naturaleza del problema. Ahora necesitamos convertir esa idea en una forma de trabajo. Una organización gobierna de verdad cuando puede responder quién decide, en qué momento, con qué información, dentro de qué límites y qué ocurre si el sistema falla.

La dificultad no suele estar en redactar principios. Casi todas las organizaciones pueden comprometerse con la equidad, la seguridad, la privacidad y la transparencia. Lo difícil es lograr que esos compromisos modifiquen un diseño, impidan un lanzamiento, limiten un permiso o provoquen la

retirada de un sistema. Este capítulo explica cómo construir ese puente.

2.1. Un programa existe cuando organiza decisiones

La gobernanza de la inteligencia artificial es un sistema de decisiones, responsabilidades y controles que acompaña a cada uso desde su propuesta hasta su retirada. No es un comité, una política ni una herramienta, aunque puede necesitar los tres. Su finalidad es permitir que la organización avance sin perder la capacidad de examinar, limitar y corregir aquello que pone en funcionamiento.

Un programa operativo debe resolver cinco decisiones. Las palabras pueden cambiar entre organizaciones, pero las funciones permanecen.

La prueba más sencilla

Un principio gobierna cuando puede localizarse el momento en que cambia una decisión. Si no puede impedir, limitar o corregir nada, sigue siendo una declaración.

Decisión	Pregunta que debe resolver	Resultado visible
Criterios	Qué usos son aceptables y bajo qué condiciones	Política traducida a reglas verificables
Responsables	Quién propone, aprueba, controla y responde	Derechos de decisión y propietarios identificados
Método	Cómo se clasifica y trata el riesgo	Marco común, escalas y documentación
Ciclo de vida	Qué debe demostrarse antes de avanzar o cambiar	Puertas de control, seguimiento y retirada
Capacidad	Qué deben saber quienes diseñan, usan y supervisan	Formación adaptada al papel y al riesgo

2.2. Separar el núcleo estable de la capa jurídica

Las leyes cambian por país, sector, tipo de sistema y momento. Una empresa que opera en varios mercados no puede construir un programa distinto para cada norma, porque terminaría multiplicando formularios sin comprender mejor sus riesgos. Tampoco puede elegir una sola jurisdicción y suponer que cubre las demás.

La solución consiste en separar dos capas. El núcleo estable contiene inventario, finalidad, evaluación, responsables, permisos, trazabilidad, condiciones de parada, gestión de

cambios y retirada. La *capa jurídica*[8] traduce ese núcleo a obligaciones concretas, identifica excepciones y conserva fechas de entrada en vigor. La primera cambia despacio. La segunda debe estar versionada y tener un responsable que la mantenga.

Núcleo estable	Capa jurídica actualizable
Finalidad, población afectada y propietario	Definiciones legales y alcance territorial
Evaluación de impacto y controles proporcionales	Clasificación normativa y obligaciones aplicables
Registros, supervisión, incidentes y retirada	Plazos, autoridades, guías y mecanismos de reclamación
Gestión de modelos, datos, herramientas y proveedores	Requisitos sectoriales, contractuales y de información

La evolución reciente lo demuestra. El Reglamento europeo de inteligencia artificial pasó a ser aplicable con carácter general el 2 de agosto de 2026 y comenzó su etapa de ejecución, mientras algunas obligaciones para sistemas de alto riesgo quedaron desplazadas a 2027 y 2028. En Colorado, la ley aprobada en 2024 fue sustituida en mayo de 2026 por un marco más estrecho para tecnologías que intervienen en decisiones relevantes, con aplicación prevista desde enero de

[8] *Capa jurídica:* Registro actualizable que traduce los principios y controles generales de gobernanza a las obligaciones, plazos y jurisdicciones concretas aplicables a una organización.

2027. Un programa que hubiera incrustado cada fecha y definición dentro de todos sus procedimientos habría tenido que rehacerse varias veces.

Regla de mantenimiento

Las fechas, jurisdicciones y referencias legales deben vivir en un anexo o registro controlado. El capítulo y el programa conservan los principios duraderos; la organización actualiza su traducción jurídica sin reconstruir todo el sistema.

2.3. Traducir principios a criterios verificables

Los principios sirven para orientar, pero no indican por sí solos qué debe hacer un equipo ante un caso concreto. La traducción exige completar una cadena sencilla. Cada principio se convierte en un criterio, el criterio se aplica en una decisión y la decisión deja evidencia.

Esta cadena evita dos errores frecuentes. El primero es medir algo porque resulta fácil, aunque no guarde relación con el daño que se pretende evitar. El segundo es acumular documentos que describen el sistema sin cambiar su comportamiento. La evidencia tiene valor cuando demuestra

que una decisión se tomó con un criterio conocido y que podía haber producido un resultado distinto.

Principio	Criterio Operativo	Evidencia
Equidad	No se lanza un sistema que afecte a personas sin evaluar resultados para grupos relevantes y justificar los umbrales aceptados	Informe de evaluación, datos utilizados y decisión firmada
Transparencia	La persona conoce cuándo interviene la IA, qué papel cumple y cómo puede pedir una revisión	Texto aprobado, interfaz probada y canal de reclamación
Privacidad	El sistema solo accede a categorías de datos declaradas y necesarias para la finalidad	Ficha de datos, base jurídica, permisos y plazo de conservación
Seguridad	Una salida del modelo no autoriza por sí sola una acción irreversible	Política externa de permisos, límites y aprobaciones
Supervisión humana	Quien revisa dispone de tiempo, información y autoridad para rechazar o detener	Diseño de supervisión, registro de intervenciones y pruebas

2.4. Asignar derechos de decisión antes que reuniones

Un comité multidisciplinario puede ser útil, pero no debe convertirse en propietario de todos los riesgos. Su función principal debería ser establecer criterios comunes, resolver

excepciones y supervisar la calidad del programa, mientras cada *derecho de decisión*[9] permanece asignado de forma explícita a quien corresponda dentro de la organización. Si el comité aprueba cada experimento, retrasa el trabajo y empuja a los equipos hacia canales informales. Si se limita a escuchar presentaciones, crea la apariencia de control sin ejercerlo realmente.

La responsabilidad principal corresponde al propietario del uso, es decir, a quien obtiene el beneficio y acepta las consecuencias dentro del negocio. Tecnología responde por la construcción y la operación. Datos y privacidad controlan procedencia, necesidad y acceso. Seguridad define límites técnicos y respuesta ante incidentes. Jurídico y cumplimiento interpretan obligaciones. Una función independiente comprueba que el sistema declarado coincide con el que funciona en la práctica.

El órgano de gobernanza debe fijar criterios comunes, resolver excepciones y vigilar la calidad del programa. Su tarea no es sustituir a quienes toman decisiones operativas, sino impedir que la responsabilidad se disuelva entre muchas personas hasta que nadie pueda explicar quién aceptó el riesgo.

[9] *Derecho de decisión*. Autoridad explícita para proponer, aprobar, rechazar, escalar, interrumpir o retirar un sistema.

Decisión	Quién debe responder	Cuando se eleva
Aceptar la finalidad y el riesgo residual	Propietario del uso	Cuando el impacto supera su autoridad o afecta derechos sensibles
Aprobar diseño y despliegue	Propietario del uso con responsables técnicos y de control	Cuando existe una excepción, falta evidencia o el riesgo no puede reducirse
Autorizar cambios relevantes	Responsable del sistema	Cuando cambian finalidad, modelo, datos, herramientas, permisos o población afectada
Suspender o retirar	Propietario y función con autoridad de interrupción	De inmediato cuando se supera una *condición de parada*[10]

2.5. Utilizar un marco sin convertirlo en religión

Empezar desde una hoja en blanco resulta innecesario. El Marco de Gestión de Riesgos de Inteligencia Artificial del Instituto Nacional de Estándares y Tecnología de Estados Unidos, conocido como NIST AI RMF, ofrece una estructura voluntaria para incorporar la confiabilidad al diseño, desarrollo,

[10] *Condición de parada:* Hecho observable previamente definido que obliga a suspender, limitar o escalar la operación de un sistema.

uso y evaluación. Organiza el trabajo mediante cuatro funciones relacionadas, gobernar, mapear, medir y gestionar.

ISO/IEC 42001 adopta otra perspectiva. Define los requisitos para establecer, aplicar, mantener y mejorar continuamente un *sistema de gestión*[11] de inteligencia artificial. Su lógica de planificar, hacer, comprobar y actuar resulta apropiada cuando la organización necesita integrar la IA en controles corporativos ya existentes y demostrar que el sistema se mantiene en el tiempo.

[11] *Sistema de gestión de IA:* Conjunto organizado de políticas, responsabilidades, procesos y mecanismos de mejora continua mediante los cuales una organización dirige y controla sus usos de inteligencia artificial.

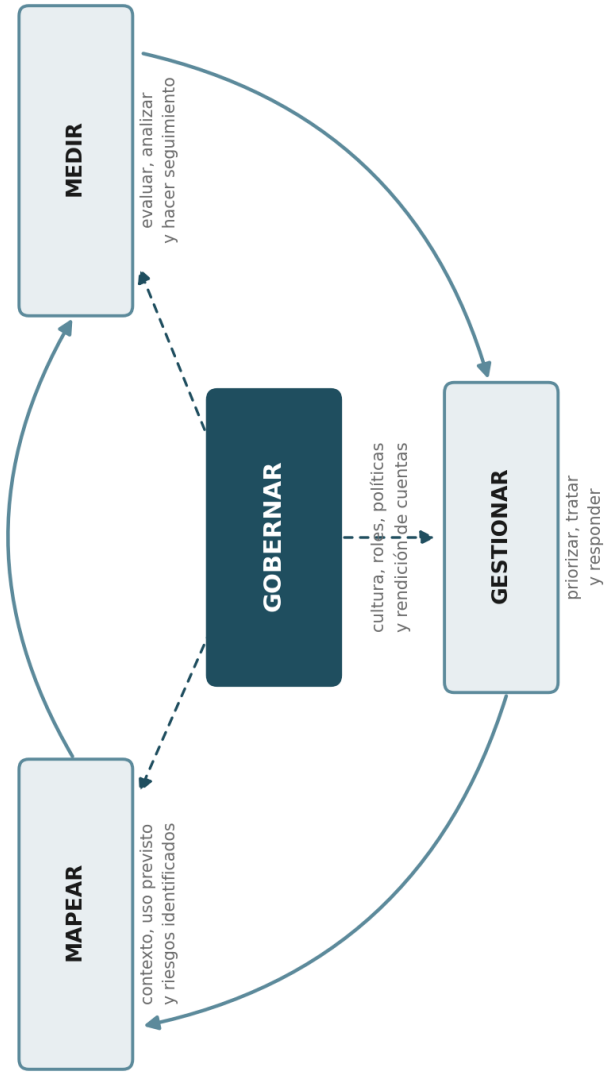

Figura 2.1 Gobernar atraviesa las demás funciones y no constituye una etapa aislada.

Ambas referencias pueden complementarse. NIST ayuda a ordenar el razonamiento sobre riesgos y resultados. ISO/IEC 42001 ayuda a convertir ese razonamiento en un sistema de gestión sostenido. Ninguna demuestra por sí sola que un caso de uso sea seguro, legal o justo. El marco organiza las preguntas; la organización sigue siendo responsable de las respuestas.

2.6. Gobernar cada cambio durante el ciclo de vida

Un proyecto no necesita la misma evidencia en todas sus etapas. Antes de construir importa la finalidad y la necesidad. Antes de desplegar importan las pruebas, los límites, la supervisión y la responsabilidad. Durante la operación importan el comportamiento real, los incidentes y los cambios. Al retirar importan las credenciales, los datos, las dependencias y la conservación de evidencia.

La *puerta de control*[12] de entrada comprueba finalidad, personas afectadas, datos necesarios y alternativas menos intrusivas. La puerta de lanzamiento exige resultados de evaluación, controles de seguridad, límites de autonomía,

[12] *Puerta de control*. Punto del ciclo de vida en el que un sistema debe demostrar que cumple determinados criterios antes de poder avanzar a la siguiente etapa.

supervisión y un propietario identificado. La puerta de cambio se activa cuando se modifica un modelo, una instrucción, una fuente de datos, una herramienta, un permiso o un proveedor. La revisión de operación analiza incidentes, reclamaciones, desviaciones, costes y señales de uso no previsto.

No todos los sistemas necesitan el mismo trámite. Un asistente que redacta un borrador con información pública puede recorrer un camino ligero. Un sistema que influye sobre crédito, empleo, salud o prestaciones requiere análisis y evidencia mucho más exigentes. La proporcionalidad reduce burocracia porque concentra el esfuerzo donde una decisión incorrecta puede afectar de manera seria a una persona o a la organización.

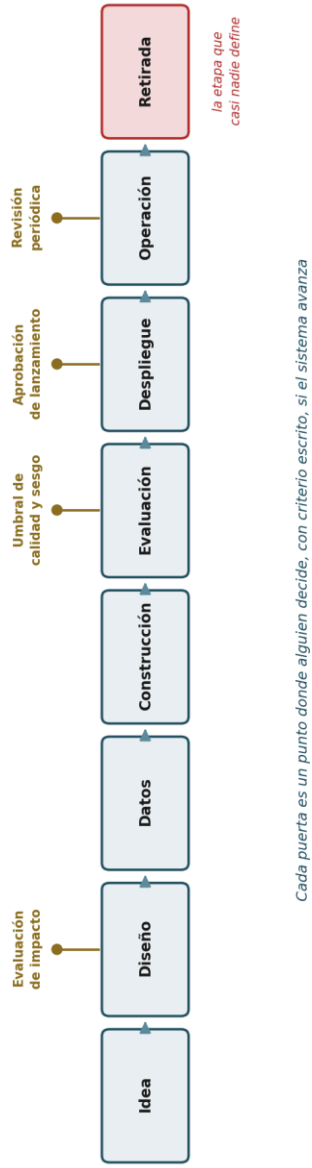

Figura 2.2 Cada puerta necesita un criterio escrito, un responsable y una consecuencia si no se supera.

2.7. Prever excepciones, incidentes y retirada

Los programas suelen describir la vía normal y olvidar aquello que ocurre cuando la realidad se aparta del procedimiento. Una *excepción*[13] no debe resolverse mediante un correo informal. Necesita motivo, alcance, propietario, controles compensatorios y fecha de vencimiento. Si no caduca, termina convirtiéndose en una política paralela que nadie recuerda haber aprobado.

La respuesta a incidentes también debe empezar antes del incidente. La organización necesita saber qué señales obligan a detener el sistema, quién tiene autoridad para hacerlo, cómo se preserva la evidencia, a quién se informa y qué condiciones permiten reanudar la operación. En un agente, desactivar la interfaz puede no ser suficiente. También puede ser necesario revocar credenciales, cancelar tareas pendientes, desconectar herramientas y comprobar que no queda actividad residual.

La retirada es una etapa del ciclo de vida, no una tarea administrativa posterior. Un sistema abandonado puede conservar datos, permisos y conexiones mientras pierde a la

[13] *Excepción*. Desviación temporal y expresamente aprobada respecto de un criterio de gobernanza, con alcance, responsable, controles compensatorios y fecha de vencimiento.

persona que comprendía su propósito. Darlo de baja exige revocar accesos, conservar la evidencia necesaria, eliminar o archivar datos según la obligación aplicable, informar a quienes corresponda y marcar el activo como retirado en el inventario.

La condición de parada

Debe describir un hecho observable, indicar quién puede interrumpir la operación y producir una acción concreta. «Detener si el sistema se comporta mal» no es un control. «Bloquear pagos cuando el importe acumulado supera el límite autorizado» sí lo es.

2.8. Formar según la responsabilidad y el riesgo

Una gobernanza que vive en un solo departamento no puede controlar el uso real. Quien compra una aplicación, diseña un proceso, prepara datos, configura permisos, supervisa resultados o atiende una reclamación necesita comprender riesgos diferentes. Por eso la formación debe seguir al papel y al sistema, no al organigrama.

El artículo 4 del Reglamento europeo exige que proveedores y responsables del despliegue adopten medidas de alfabetización en inteligencia artificial para las personas que operan o utilizan estos sistemas en su nombre. Tras la reforma de 2026 continúa la obligación de promover esa alfabetización,

aunque no se exige garantizar un nivel individual específico. La Comisión recomienda considerar el conocimiento previo, la función de la organización, el riesgo del sistema y el contexto de uso.

La señal de una formación útil no es la cantidad de asistentes. Aparece cuando una persona reconoce que está utilizando información no autorizada, cuestiona una salida convincente, identifica un cambio que necesita aprobación o sabe cómo detener una operación. El aprendizaje debe conectarse con decisiones reales, ejercicios y mecanismos de ayuda.

2.9. El sistema mínimo que permite empezar

Una organización no necesita esperar a tener una oficina especializada, una certificación o una plataforma completa. Puede empezar con un conjunto reducido de elementos que produzcan visibilidad y disciplina.

- → Un inventario vivo de sistemas y usos, con finalidad, propietario, datos, proveedores, herramientas y estado.
- → Una clasificación proporcional basada en impacto, autonomía, reversibilidad, alcance y sensibilidad del dominio.

- → Criterios de admisión, escalamiento, excepción y prohibición que los equipos puedan aplicar sin interpretar cada vez la política.
- → Puertas de control para propuesta, despliegue, cambios relevantes, operación y retirada.
- → Derechos de decisión que indiquen quién recomienda, quién controla, quién acepta el riesgo y quién puede detener.
- → Un paquete mínimo de evidencia con evaluaciones, versiones, autorizaciones, incidentes y revisiones.
- → Formación ajustada a los papeles y un canal para consultar dudas antes de que se conviertan en incidentes.

Este sistema inicial no resuelve todos los problemas, pero cambia la posición de la organización. La IA deja de aparecer como una suma de experimentos aislados y empieza a convertirse en una capacidad conocida, limitada y revisable. A partir de ahí se pueden añadir automatización, herramientas y controles más sofisticados sin perder la lógica común.

2.10. Gobernar es conservar la capacidad de decidir

La gobernanza no pretende anticipar cada error ni obligar a que todas las decisiones pasen por un comité. Su función es más concreta. Debe conseguir que la organización

sepa qué ha autorizado, por qué lo hizo, qué límites impuso, qué evidencia conserva y quién puede cambiar el rumbo.

Cuando estas respuestas existen, el control deja de ser una barrera añadida al final. Se convierte en parte del diseño y permite avanzar con mayor rapidez porque los equipos conocen el camino. Cuando no existen, la velocidad inicial se paga después mediante sistemas invisibles, excepciones permanentes, permisos huérfanos y decisiones que nadie recuerda haber aceptado.

El siguiente capítulo entra en la materia que este programa debe ordenar. Un modelo, una aplicación, una fuente de datos y un agente no son la misma cosa, aunque puedan formar parte de un solo sistema. Antes de clasificar riesgos o asignar controles, necesitamos reconocer exactamente qué componentes producen la consecuencia que queremos gobernar.

Notas y referencias

Las referencias se comprobaron el 2 de septiembre de 2026 así como los marcos normativos, los estándares y la documentación técnica.

→ **Comisión Europea.** *Reglamento de Inteligencia Artificial, aplicación, gobernanza y calendario actualizado.*

https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai

→ **Comisión Europea.** *Preguntas y respuestas sobre alfabetización en inteligencia artificial y artículo 4.*

https://digital-strategy.ec.europa.eu/en/faqs/ai-literacy-questions-answers

→ **Unión Europea.** *Reglamento (UE) 2024/1689 por el que se establecen normas armonizadas en materia de inteligencia artificial.*

https://eur-lex.europa.eu/eli/reg/2024/1689/oj

→ **NIST.** *Artificial Intelligence Risk Management Framework y recursos de aplicación.*

https://eur-lex.europa.eu/eli/reg/2024/1689/ojhttps://www.nist.gov/itl/ai-risk-management-framework

→ **ISO. ISO/IEC 42001:2023,** *sistemas de gestión de inteligencia artificial.*

https://www.iso.org/standard/42001

→ **Asamblea General de Colorado.** *SB 26-189, Automated Decision-Making Technology, texto y resumen promulgados.*

https://leg.colorado.gov/bills/sb26-189

Nota: Enlaces comprobados el 2 de Septiembre de 2026

II

La arquitectura

3

¿Qué estamos gobernando?

El capítulo anterior explicó cómo convertir principios en decisiones, responsables, controles y evidencia. Sin embargo, ese programa no puede funcionar si la organización no sabe cuál es el objeto que pretende gobernar. Decir que utiliza un modelo concreto aporta muy poca información. El mismo modelo puede limitarse a resumir un documento o intervenir en un proceso con datos personales, herramientas, memoria y capacidad de actuar.

La diferencia no está dentro del modelo, sino en el sistema construido a su alrededor y en el uso que la organización le asigna. Por eso la unidad adecuada de gobierno no es una marca, una versión ni una interfaz. Es una capacidad

concreta puesta en *operación*[14], con una finalidad, unas personas afectadas, unos datos, unas conexiones, unos límites y un responsable.

3.1. El modelo es solo una pieza

Durante la primera expansión de la inteligencia artificial generativa, la conversación pública se concentró en comparar modelos. Se preguntaba cuál escribía mejor, cuál razonaba con mayor precisión o cuál admitía más información en una consulta. Esas diferencias siguen importando, pero no describen el comportamiento de una aplicación completa.

Un *modelo*[15] recibe una entrada y produce una salida a partir de patrones aprendidos. Para convertirlo en un producto útil, alguien añade instrucciones, una interfaz, fuentes de información, reglas, filtros y mecanismos de evaluación. Si

[14] *Capacidad en operación*. Uso específico de un sistema dentro de un proceso, con finalidad, personas, datos, responsables y consecuencias definidas. En este libro se utiliza como unidad principal de gobierno.

[15] *Modelo:* En este contexto, modelo se refiere al componente aprendido de un sistema de IA que interpreta entradas y produce predicciones, clasificaciones o contenidos. No debe confundirse con la aplicación o el sistema completo.

además puede seleccionar herramientas y encadenar pasos para cumplir un objetivo, hablamos de un *agente*[16].

En todos los casos, aquello que llega al usuario es una composición, no el modelo aislado.

Idea principal de este capítulo

El riesgo y el valor pertenecen a la capacidad completa que la organización pone en funcionamiento. Evaluar solamente el modelo equivale a examinar un motor sin preguntar en qué vehículo se instaló, quién lo conduce ni por qué carretera circula.

La documentación técnica refleja esta separación. OpenAI describe un agente mediante modelo, herramientas e instrucciones.

Anthropic distingue los flujos con caminos programados de los agentes que deciden dinámicamente cómo avanzar y señala que el modelo puede ampliarse con

[16] *Agente*: Sistema que utiliza un modelo para decidir pasos y emplear herramientas con el fin de completar una tarea. El grado de autonomía puede variar según las herramientas disponibles, los permisos y las reglas de ejecución.

recuperación de información[17], herramientas y *memoria*[18]. Las definiciones no son idénticas, pero coinciden en lo esencial. La capacidad surge de la relación entre varias piezas.

3.2. Cuatro objetos que conviene distinguir

Parte de la confusión nace de utilizar modelo, producto, agente y sistema como si fueran sinónimos. Distinguirlos no es un ejercicio académico. Cada palabra cambia qué debe documentarse, evaluarse y asignarse a un responsable.

Una misma aplicación puede contener varias capacidades en operación. Un asistente corporativo puede resumir documentos públicos, consultar expedientes internos y preparar una respuesta para un cliente. Aunque la interfaz sea la misma, los tres usos no comparten finalidad, datos ni consecuencias. Registrarlos como un único activo ocultaría las diferencias que importan.

También puede ocurrir lo contrario. Una capacidad empresarial puede depender de varios modelos y aplicaciones.

---

[17] *Recuperación de información:* Mecanismo mediante el cual un sistema consulta fuentes externas —por ejemplo, documentos, bases de datos o repositorios— para incorporar información relevante al contexto utilizado por el modelo.

[18] *Memoria*: En este contexto sería el mecanismo mediante el cual una aplicación o agente conserva o recupera información de interacciones anteriores o de un estado persistente para utilizarla en pasos posteriores.

Un proceso de atención al cliente quizá clasifique la consulta con un modelo pequeño, recupere información de una base documental y redacte la respuesta con otro modelo. Desde la perspectiva de gobierno, lo relevante es el resultado que produce el conjunto.

Decisión	Qué es	Qué pregunta permite responder
Modelo	Componente aprendido que interpreta entradas y genera predicciones o contenidos	Qué capacidades y límites tiene la pieza central
Producto o aplicación	Interfaz y lógica que combinan el modelo con instrucciones, datos y funciones	Qué experiencia y opciones recibe el usuario
Agente	Sistema que utiliza un modelo para decidir pasos y emplear herramientas en busca de un objetivo	Qué puede hacer con cierto grado de independencia
Capacidad en operación	Uso específico del sistema dentro de un proceso, con personas, datos, responsables y consecuencias	Qué resultado real debe gobernar la organización

3.3. El mismo modelo puede crear dos riesgos distintos

Imaginemos que dos equipos utilizan exactamente el mismo modelo de lenguaje. El primero lo emplea para resumir informes públicos y obliga a una persona a revisar el borrador. El segundo lo conecta al correo, al gestor de clientes y a una

herramienta capaz de modificar registros. Ambos pueden anunciar que usan la misma inteligencia artificial, pero la afirmación es casi irrelevante para comprender su riesgo.

Elemento	Sistema de resumen	Sistema conectado
Finalidad	Preparar un borrador informativo	Resolver solicitudes de clientes
Información	Documentos públicos seleccionados	Mensajes, perfiles y expedientes internos
Herramientas	Ninguna	Correo y gestor de clientes
Acción	Produce texto para revisión	Puede escribir y actualizar registros
Daño posible	Un resumen incompleto o incorrecto	Una comunicación indebida, un cambio erróneo o una exposición de datos
Gobierno necesario	Calidad, fuentes y revisión	Además, permisos, identidad, límites, supervisión y registro de acciones

La tabla muestra por qué una clasificación basada solo en el modelo fracasa. El aumento de riesgo no procede necesariamente de una inteligencia más potente. Aparece cuando el sistema incorpora información sensible, autoridad de ejecución, mayor alcance o menos oportunidades de corrección.

Esta relación también funciona en sentido inverso. Un modelo muy capaz puede utilizarse con poca autoridad para investigar, comparar alternativas o preparar decisiones. La

organización obtiene parte de su valor sin permitirle ejecutar la consecuencia final. La separación entre capacidad intelectual y autoridad se desarrollará más adelante, junto con identidad, permisos y controles de ejecución.

3.4. La unidad de gobierno es una capacidad situada

Un modelo no tiene por sí solo una finalidad empresarial. Tampoco conoce quién sufrirá las consecuencias de un error ni qué procedimientos rodean su salida. Esas propiedades aparecen cuando se incorpora a un proceso. Por eso conviene gobernar una capacidad situada, es decir, un uso concreto dentro de un entorno concreto.

La expresión *sistema sociotécnico*[19] ayuda a recordar que el resultado no lo produce únicamente el software. Intervienen las personas que formulan la petición, quienes interpretan la respuesta, las reglas del proceso, los incentivos, los datos disponibles y las posibilidades de reclamar o corregir. Un sistema puede mostrar buenas métricas técnicas y generar malos resultados si se introduce en un proceso que empuja a aceptar sus recomendaciones sin tiempo ni autoridad para revisarlas.

[19] *Sistema sociotécnico*. Conjunto formado por tecnología, personas, reglas, procesos e incentivos que participa en la producción de un resultado. Analizar solamente el componente tecnológico deja fuera una parte del sistema.

Una prueba práctica

Si dos usos comparten proveedor e interfaz, pero cambian la finalidad, las personas afectadas, la información, las acciones permitidas o la posibilidad de corregir, deben tratarse como capacidades diferentes o como variantes gobernadas por separado.

El NIST AI Risk Management Framework adopta una perspectiva amplia al referirse al diseño, desarrollo, uso y evaluación de productos, servicios y sistemas de inteligencia artificial. Esa amplitud es útil porque impide concentrar toda la evaluación en el momento de elegir un modelo. La pregunta de gobierno debe seguir al sistema durante su ciclo de vida y dentro del contexto donde produce efectos.

3.5. Dibujar la frontera a partir de la consecuencia

La *frontera del sistema*[20] no debe trazarse siguiendo únicamente el contrato del proveedor o el diagrama técnico. Debe incluir todo aquello que pueda cambiar la consecuencia que se pretende controlar. Si una recomendación influye en la concesión de crédito, forman parte del análisis el modelo, los datos enviados, las reglas que combinan la salida con otras

[20] *Frontera del sistema:* Límite analítico que incluye los componentes y relaciones capaces de modificar la consecuencia que se pretende gobernar.

señales, la pantalla que ve el analista, el tiempo disponible para revisar y el mecanismo de reclamación.

Esta forma de mirar evita dos extremos. El primero consiste en reducir el sistema al componente de inteligencia artificial y atribuirle decisiones que en realidad surgen del proceso completo. El segundo consiste en ampliar tanto la frontera que nadie puede trabajar con ella. La delimitación debe ser suficientemente amplia para explicar el resultado y suficientemente concreta para asignar responsabilidades.

Capa	Qué puede cambiar	Por qué pertenece a la frontera
Finalidad y proceso	La tarea, el momento y la consecuencia de la salida	Determinan para qué se usa la capacidad
Modelo e instrucciones	Comportamiento, formato y criterios de respuesta	Condicionan lo que el sistema propone
Datos y contexto	Información disponible, actualidad y sensibilidad	Condicionan lo que el sistema conoce
Herramientas y conexiones	Sistemas que puede consultar o modificar	Transforman una respuesta en una acción
Personas y supervisión	Quién pide, revisa, acepta, corrige o reclama	Determinan cómo se convierte la salida en una decisión
Proveedor e infraestructura	Versiones, disponibilidad, localización y cambios externos	Pueden alterar el sistema sin modificar el proceso visible

Estas capas no sustituyen al inventario ni a la evaluación de riesgo que se desarrollarán en capítulos posteriores. Su función aquí es más básica. Permiten reconocer qué pertenece al sistema antes de decidir cómo documentarlo o controlarlo.

3.6. Las relaciones importan tanto como las piezas

Una lista de componentes puede ofrecer una falsa sensación de conocimiento. Dos sistemas con los mismos elementos pueden comportarse de forma distinta según cómo se conecten. Una base de datos accesible solo para consulta no equivale a otra que permite escritura. Una revisión humana anterior a una acción no cumple la misma función que una comprobación posterior. Una herramienta disponible bajo petición no concede la misma autonomía que otra elegida por el agente durante un proceso prolongado.

Para comprender un sistema hay que seguir tres recorridos. El primero muestra cómo entra y se transforma la información. El segundo indica cómo una salida llega a convertirse en acción. El tercero revela dónde se detiene el proceso, quién puede intervenir y qué queda registrado. Estos recorridos exponen dependencias que una ficha de proveedor no muestra.

- → El recorrido de la información identifica fuentes, transformaciones, destinos y conservación.

- → El recorrido de la autoridad muestra qué componente propone, qué regla permite y quién responde por la acción.
- → El recorrido de la evidencia permite reconstruir versiones, decisiones, intervenciones y resultados.

En febrero de 2026, NIST lanzó una iniciativa específica sobre estándares para agentes y situó entre sus prioridades la interoperabilidad, la seguridad, la identidad y la autorización. La señal es importante porque reconoce que los agentes obtienen utilidad al relacionarse con datos y sistemas externos. También confirma que el problema ya no puede describirse solamente mediante las propiedades del modelo.

3.7. Un proveedor entrega componentes, no transfiere la responsabilidad

La mayoría de las organizaciones no construirá todas las piezas que utiliza. Comprará modelos, aplicaciones, conectores, servicios de datos y plataformas de operación. Esa dependencia no elimina la responsabilidad sobre el uso. El proveedor puede explicar cómo funciona su componente y aportar evaluaciones, pero no conoce por completo la finalidad local, las personas afectadas ni la manera en que una salida se incorpora a una decisión.

Por eso una evaluación del proveedor y una evaluación del sistema responden preguntas distintas. La primera examina capacidades, seguridad, condiciones contractuales,

cambios, soporte y evidencia disponible. La segunda observa qué sucede cuando ese componente se combina con datos, procesos y autoridad dentro de la organización. Un certificado o una prueba general puede apoyar la segunda evaluación, pero no reemplazarla.

También importa la capacidad de detectar cambios externos. Un servicio puede actualizar el modelo, modificar una política de conservación o retirar una función sin que el usuario final perciba una migración formal. Si la organización no conoce sus dependencias ni las condiciones de notificación, puede terminar operando una versión distinta de la que evaluó.

3.8. Cada cambio relevante crea una nueva pregunta

Gobernar una capacidad completa obliga a revisar qué significa cambiarla. Sustituir el modelo es una modificación evidente, pero no es la única. Añadir una fuente documental, ampliar una población, activar memoria, conectar una herramienta, reducir la revisión humana o modificar el propósito puede alterar más el riesgo que una actualización del modelo.

No todos los cambios requieren repetir todo el proceso. La respuesta debe ser proporcional y puede ir desde una comprobación limitada hasta una nueva aprobación. Lo indispensable es disponer de criterios que distingan una

modificación rutinaria de otra capaz de alterar la finalidad, el impacto o la autoridad del sistema.

Cambio	Pregunta que debe reabrirse
Modelo, versión o configuración	Siguen siendo válidas las pruebas y los límites conocidos
Datos o fuentes de contexto	Cambian la calidad, la sensibilidad, los derechos o la posibilidad de manipulación
Herramientas o permisos	Aparecen nuevas acciones, destinos o consecuencias
Finalidad, usuarios o población afectada	La evaluación anterior cubre realmente el nuevo uso
Supervisión o integración en el proceso	Una persona conserva tiempo, información y autoridad para intervenir
Proveedor o infraestructura	Cambian contratos, localización, dependencias, continuidad o capacidad de auditoría

3.9. La ficha mínima de una capacidad

Antes de aplicar marcos complejos, una organización debería poder describir cada capacidad con un conjunto estable de preguntas. La ficha no necesita convertirse en un informe extenso. Debe permitir que otra persona comprenda qué se ha puesto en funcionamiento y localice la evidencia correspondiente.

- → Qué resultado persigue y qué necesidad justifica el uso de inteligencia artificial.
- → Quién es el propietario de la capacidad y quiénes participan en su operación y control.
- → A qué personas afecta y qué consecuencia puede producir una salida incorrecta.
- → Qué modelos, instrucciones, datos, herramientas, proveedores e infrastructura intervienen.
- → Qué acciones puede realizar, cuáles requieren aprobación y cuáles están prohibidas.
- → Cómo intervienen las personas, cómo pueden corregir el resultado y qué canal existe para reclamar.
- → Qué versiones se evaluaron, qué cambios obligan a revisar y qué evidencia permite reconstruir la operación.

Esta ficha establece el objeto común sobre el que trabajarán tecnología, negocio, seguridad, datos, jurídico y auditoría. Cada función podrá profundizar en su ámbito sin describir un sistema diferente. Cuando las áreas parten de fronteras incompatibles, la responsabilidad se fragmenta y los controles dejan huecos entre componentes.

3.10. De preguntar por el modelo a comprender la capacidad

La calidad del modelo continúa siendo importante, pero ya no basta para explicar el valor ni el riesgo de una aplicación. Una organización puede cambiar profundamente un sistema sin sustituirlo y puede utilizar el mismo modelo en capacidades que necesitan niveles de control muy distintos. La pregunta correcta no es cuál es el mejor modelo, sino qué resultado produce el conjunto, sobre quién, con qué información y con qué autoridad.

Esta mirada también mejora las decisiones de inversión. Obliga a identificar dónde reside realmente la ventaja. A veces estará en el modelo; otras, en los datos, la integración, el diseño del proceso, la experiencia humana o la capacidad de supervisar. Comprar una tecnología potente no crea por sí sola una capacidad útil. Esa capacidad aparece cuando las piezas funcionan juntas dentro de límites comprensibles.

Los próximos capítulos podrán examinar esas piezas sin perder de vista el conjunto. Las conexiones y la cadena de suministro explicarán de dónde proceden las capacidades. La seguridad mostrará cómo limitar herramientas, memoria y acciones. La identidad y los permisos aclararán quién puede

hacer qué. La observabilidad[21] aportará la evidencia necesaria para reconstruir lo ocurrido. Este capítulo conserva una sola función dentro de esa arquitectura. Recordarnos que ninguna de esas cuestiones puede gobernarse correctamente mientras el modelo siga confundido con el sistema.

[21] *Observabilidad:* Capacidad de comprender y reconstruir el comportamiento de un sistema a partir de registros, eventos, métricas, versiones y otras evidencias de su operación.

Notas y referencias

Las referencias técnicas se comprobaron el 2 de septiembre de 2026. Las páginas de producto y la documentación de agentes cambian con rapidez, por lo que deben revisarse antes del cierre editorial y en cada nueva edición.

→ **NIST.** *Artificial Intelligence Risk Management Framework y recursos de aplicación.*

https://www.nist.gov/itl/ai-risk-management-framework

→ **NIST.** *AI Agent Standards Initiative for Interoperable and Secure Innovation, 17 de febrero de 2026.*

https://www.nist.gov/news-events/news/2026/02/announcing-ai-agent-standards-initiative-interoperable-and-secure

→ **OpenAI.** A *practical guide to building agents.*

https://openai.com/business/guides-and-resources/a-practical-guide-to-building-ai-agents/

→ **Anthropic.** *Building effective agents, 19 de diciembre de 2024, con aviso de actualización posterior.*

https://www.anthropic.com/engineering/building-effective-agents

Nota: Enlaces comprobados el 2 de Septiembre de 2026

→

4

Cuando los permisos se combinan

Lo que enseñó la primera gran crisis pública de seguridad agéntica

El capítulo anterior fijó la unidad de gobernanza. Una organización no gobierna un modelo aislado, sino una capacidad completa formada por datos, instrucciones, herramientas, permisos, memoria, personas y proveedores. Este capítulo observa qué ocurre cuando esas piezas se conectan con rapidez y sin una frontera de seguridad equivalente a la autoridad concedida.

El caso de OpenClaw resulta útil porque no presenta un peligro hipotético. El proyecto mostró en pocos meses la utilidad de un agente persistente que trabaja en una computadora y, a la vez, la facilidad con la que una configuración apresurada puede convertir esa utilidad en exposición. La lección no consiste en declarar inseguro un

producto para siempre. Consiste en entender que el riesgo de un agente nace de la combinación de capacidades y cambia cada vez que añadimos una nueva conexión.

4.1. El mismo diseño que crea valor amplía el daño posible

OpenClaw se popularizó como una plataforma abierta para ejecutar asistentes en una máquina elegida por el usuario. Podía recibir encargos desde aplicaciones de mensajería, conservar contexto y utilizar herramientas para organizar el correo, enviar mensajes, gestionar el calendario o trabajar con archivos. Esa proximidad a los datos y a las aplicaciones explicaba buena parte de su atractivo.

La idea central

La capacidad de un agente no se mide solo por la calidad del modelo. También depende de los datos que puede leer, las herramientas que puede invocar, los destinos a los que puede escribir y el tiempo durante el que puede actuar sin revisión.

También explicaba el riesgo. Un chatbot encerrado en una conversación puede producir una respuesta incorrecta. Un agente con acceso al correo, al sistema de archivos, a una terminal y a credenciales puede convertir una interpretación incorrecta o una instrucción maliciosa en una acción. La diferencia no está en que el modelo se vuelva hostil. Está en que

el sistema dispone de medios para transformar texto en consecuencias.

4.2. Una adopción rápida reveló varias fronteras débiles

A comienzos de 2026 el proyecto creció con una velocidad poco habitual. Miles de personas instalaron un asistente permanente y lo conectaron a cuentas, repositorios, canales de mensajería y dispositivos. Esa expansión convirtió un experimento de entusiastas en una prueba pública de arquitectura. En pocas semanas aparecieron vulnerabilidades en la pasarela de control, instalaciones expuestas a internet y habilidades maliciosas distribuidas mediante el catálogo comunitario.

Una de las vulnerabilidades documentadas como *CVE-2026-25253*[22] permitía robar el *token de autenticación*[23] de la pasarela mediante una página maliciosa y utilizar después ese acceso para ejecutar acciones. Al mismo tiempo, distintas mediciones encontraron decenas de miles de instalaciones visibles desde internet. Los recuentos variaron mucho porque

[22] *CVE:* Identificador público de una vulnerabilidad conocida, formado por el prefijo CVE, el año y un número.

[23] *Token de autenticación:* Valor digital que permite demostrar ante un sistema que una identidad o sesión ha sido autenticada y está autorizada para acceder a determinados recursos.

cada investigador utilizó fechas y técnicas de detección diferentes.

Momento	Hecho comprobado	Qué demuestra
Finales de enero de 2026	Se corrige una vulnerabilidad que permitía robar el token de la pasarela y *encadenar ejecución remota de código*[24]	El *plano de control*[25] necesita una frontera propia y no puede confiar solo en el navegador o la red local
Principios de febrero	Investigadores documentan 341 habilidades maliciosas entre 2.632 revisadas	Una habilidad debe tratarse como una dependencia con capacidad de actuar, no como simple documentación
Febrero de 2026	Distintos servicios detectan grandes cantidades de instalaciones accesibles desde internet	La exposición depende tanto de la configuración como del código
Febrero a junio	El proyecto añade análisis de habilidades, controles de exposición, políticas de herramientas y guías de endurecimiento	La seguridad del runtime[26] es una disciplina continua y no una función terminada

Tabla 4.1 Secuencia de descubrimiento, respuesta y aprendizaje.

[24] *Ejecución remota de código:* Capacidad de ejecutar instrucciones en una máquina ajena a través de una conexión o vulnerabilidad.

[25] *Plano de control:* Interfaz y servicios desde los que se configura, autoriza, supervisa o administra un sistema.

La *cadena de suministro de software*[27] añadió otro problema. Una investigación temprana identificó 341 habilidades maliciosas entre 2.632 paquetes revisados en ClawHub. Meses después, Unit 42 encontró habilidades evasivas que continuaban eludiendo algunos controles incorporados tras la primera campaña. El dato más importante no es el porcentaje exacto. Es que una habilidad escrita como instrucciones puede heredar la autoridad del agente que la instala y utilizar herramientas legítimas para fines distintos de los esperados.

4.3. Una cifra de exposición necesita método y fecha

Los informes publicados durante la crisis citaron cifras muy distintas. Algunos contaban direcciones IP. Otros identificaban servicios por cabeceras, certificados o patrones de respuesta. Unos medían instalaciones visibles y otros intentaban comprobar si además eran vulnerables. También observaban momentos diferentes, mientras los usuarios instalaban parches, cerraban puertos o modificaban cortafuegos.

---

[27] *Cadena de suministro de software:* Conjunto de componentes, bibliotecas, herramientas, paquetes y proveedores externos que intervienen en la creación o funcionamiento de un sistema y que pueden introducir riesgos propios.

Por eso una reducción entre dos mediciones no demuestra por sí sola que el riesgo haya disminuido en la misma proporción. Dejar de ser visible para una técnica de búsqueda puede significar que se corrigió la exposición, que cambió la huella o que el servicio se movió detrás de otra capa. La cifra solo permite una conclusión cuando conocemos qué se buscó, cuándo y con qué criterio se consideró un resultado válido.

Regla para informes de seguridad

Ninguna cifra de exposición debería llegar a un comité sin fecha, método y responsable de la medición. Si falta uno de esos elementos, el número sirve como señal, pero no como medida comparable.

4.4. No hubo un único fallo

Agrupar todo bajo la etiqueta de inseguridad oculta las decisiones que deben corregirse. El episodio reunió al menos cuatro familias de fallo. Algunas pertenecen a la seguridad clásica del software y otras adquieren una forma nueva cuando el sistema interpreta lenguaje y elige herramientas. Cada familia necesita un control diferente.

Ninguna de estas familias se resuelve pidiendo al modelo que se comporte bien. Una instrucción puede orientar, pero no sustituye la autenticación, el aislamiento, la política de herramientas ni la gestión de dependencias. Un sistema comprometido puede ignorar el consejo. El control efectivo

debe existir en una capa que el modelo no pueda modificar mediante lenguaje.

Familia	Qué ocurrió	Control dominante
Exposición del plano de control	Pasarelas accesibles desde internet o confiadas a límites de red insuficientes	Acceso local por defecto, autenticación obligatoria, segmentación y comprobación de configuración
Instrucciones no confiables	Contenido externo podía influir en el uso de herramientas o en la selección del siguiente paso	Separación entre datos e instrucciones, herramientas limitadas y autorización fuera del modelo
Cadena de suministro	Habilidades maliciosas o evasivas se presentaban como ampliaciones útiles	Procedencia, revisión, firma, análisis, catálogo aprobado y retirada rápida
Agencia excesiva[28]	El agente recibía acceso a archivos, comandos, cuentas o redes más amplios de lo necesario	Aislamiento, mínimo privilegio, límites de salida y permisos ligados a la tarea

4.5. El permiso cambia de significado cuando se combina con otros

La lección más transferible del caso puede expresarse de forma sencilla. Lo que una organización puede permitir depende de todo lo que ya ha permitido. Leer correo, consultar

[28] *Agencia excesiva:* Concesión de más capacidad de acción de la que requiere una tarea, de modo que un fallo de interpretación puede producir un daño mayor.

documentos internos, navegar por internet y publicar en un repositorio pueden parecer capacidades razonables por separado. Reunidas en un mismo agente, crean una ruta desde información privada hasta un canal externo.

Capacidad	Aislada	Combinada con las demás
Leer correo corporativo	Permite resumir y clasificar mensajes	Introduce contenido externo que puede contener instrucciones ocultas
Consultar documentos internos	Aporta contexto para responder	Pone información privada al alcance de cualquier acción posterior
Navegar o llamar a servicios externos	Permite buscar y completar tareas	Crea un canal por el que pueden salir datos
Agencia excesiva	El agente recibía acceso a archivos, comandos, cuentas o redes más amplios de lo necesario	Aislamiento, mínimo privilegio, límites de salida y permisos ligados a la tarea

El investigador Simon Willison describió esta combinación como la *trifecta letal*[29]. Un sistema reúne contenido no confiable, acceso a datos privados y una forma de comunicarse con el exterior. El contenido malicioso orienta la acción, los datos proporcionan aquello que interesa robar y el

[29] *Trifecta letal:* Combinación de datos privados, contenido no confiable y comunicación externa que permite construir una ruta de exfiltración.

canal de salida permite extraerlo. Retirar una de las tres condiciones rompe la cadena completa, aunque no elimine todos los riesgos.

Esta propiedad vuelve insuficiente un catálogo estático de permisos. Una autorización inocua puede dejar de serlo cuando se incorpora una nueva fuente de datos o una herramienta de salida. El cambio material no está en el permiso añadido de manera aislada, sino en la ruta que aparece al conectarlo con los anteriores.

Control propietario del capítulo

Cada vez que se añada una fuente de información, una herramienta o un destino, debe reevaluarse el conjunto completo de permisos del agente. La revisión no pregunta solo qué aporta la nueva conexión. Pregunta qué rutas crea.

4.6. Los límites efectivos viven fuera del modelo

Un *runtime* [30] gobernado convierte esa revisión en controles aplicables. Puede decidir qué herramientas ve el modelo, qué argumentos acepta cada una, qué recursos quedan dentro del entorno aislado y qué destinos de red están

[30] *Runtime:* Entorno de ejecución que aplica las reglas y controles bajo los que opera un agente, incluidos el acceso a herramientas, recursos, credenciales y destinos de red.

permitidos. También puede bloquear una acción, pedir aprobación y registrar la decisión. El modelo propone el siguiente paso, mientras otra capa decide si ese paso puede ejecutarse.

Las credenciales merecen un tratamiento especial. La *intermediación de credenciales*[31] permite mantener las claves fuera del contexto del modelo y utilizarlas únicamente cuando una operación autorizada las necesita.

Introducir claves directamente en el contexto del modelo o guardarlas en archivos que el[32] agente puede leer amplía innecesariamente el perímetro de ataque. Un diseño más sólido conserva el secreto fuera del entorno del agente y entrega una referencia o lo incorpora únicamente al realizar la llamada autorizada. Esto reduce la posibilidad de que una instrucción maliciosa lo copie, aunque no corrige una herramienta que devuelva el secreto ni una máquina anfitriona ya comprometida.

[31] *Intermediación de credenciales:* Mecanismo que mantiene los secretos fuera del contexto del modelo y los incorpora solo al realizar una operación autorizada.

[32] *Exfiltración:* Extracción no autorizada de información desde un sistema hacia un destino externo.

La *autorización progresiva*[33] aplica el mismo principio a las capacidades. El agente comienza con lo necesario para una tarea conocida. Si intenta utilizar un recurso no previsto, el sistema bloquea la acción y genera una solicitud concreta. La persona responsable puede aprobarla, rechazarla o concederla durante un plazo limitado. Registrar el motivo de una denegación resulta tan importante como registrar la concesión, porque explica qué combinación se intentaba evitar.

Control	Qué decide	Qué no resuelve por sí solo
Política de herramientas	Qué funciones están disponibles y con qué parámetros	No corrige una herramienta autorizada que esté mal diseñada
Aislamiento	Qué archivos, procesos y recursos puede alcanzar una ejecución	No determina si la finalidad del uso es legítima
Intermediación de credenciales	Cuándo se incorpora un secreto sin mostrarlo al modelo	No protege una máquina anfitriona comprometida
Control de salida	A qué dominios, servicios o canales puede comunicarse el agente	No garantiza que un destino permitido trate los datos correctamente
Aprobación humana	Qué acción excepcional puede continuar	No funciona si la persona carece de contexto o aprueba por rutina

[33] *Autorización progresiva:* Modelo que concede capacidades de forma incremental y registra cada aprobación, denegación, alcance y caducidad.

4.7. El runtime contiene consecuencias, pero no corrige una mala arquitectura

La reacción a una crisis puede producir una confianza excesiva en la capa que se presenta como solución. Un entorno aislado reduce el radio de daño, pero sigue ejecutando aquello que se le permite. Si un mismo agente conserva datos privados, contenido no confiable y comunicación externa, el diseño continúa ofreciendo una ruta de *exfiltración*, aunque resulte más difícil explotarla.

Tampoco existe un runtime que se gobierne a sí mismo. Las políticas necesitan propietarios, pruebas y revisión. Las excepciones temporales deben caducar. Las credenciales deben retirarse cuando cambia la función. Las imágenes, bibliotecas y habilidades requieren actualización. El registro debe llegar a alguien capaz de interpretar una anomalía. Llamar seguro a un entorno no acredita ninguna de esas tareas.

- → No expone por sí mismo quién decidió aceptar un riesgo ni por qué.
- → No sustituye la evaluación de procedencia de una habilidad o dependencia.
- → No convierte una aprobación masiva en supervisión humana efectiva.
- → No evita que una configuración antigua quede abierta después de cambiar el proceso.

La documentación actual de OpenClaw reconoce esta distinción. Sus guías separan la política de herramientas, el *sandbox*[34], la seguridad del sistema anfitrión y la frontera de confianza de la pasarela. También advierten que una biblioteca de operaciones seguras sobre archivos no es un sandbox. Esta precisión importa porque impide atribuir a un control una protección que nunca prometió ofrecer.

4.8. La respuesta madura no es prohibir, sino diseñar la autoridad

OpenClaw no fue importante únicamente por los incidentes. Su crecimiento mostró una demanda real de asistentes persistentes que trabajan con aplicaciones cotidianas. La respuesta útil no consiste en negar esa demanda ni en conectar el agente a todo para comprobar hasta dónde llega. Consiste en diseñar una versión de la utilidad cuyo daño posible permanezca acotado.

Una organización puede empezar por una tarea estrecha, una identidad propia, datos seleccionados y herramientas de lectura. Después puede ampliar capacidades cuando exista evidencia de que los controles funcionan. Cada ampliación necesita una revisión de rutas, no solo una casilla

[34] *Sandbox:* Entorno aislado que limita los archivos, procesos, recursos y otras capacidades a los que puede acceder una ejecución, reduciendo las consecuencias de un fallo o comportamiento no previsto.

añadida al inventario. Si la nueva herramienta permite escribir, comunicar o ejecutar, cambia la naturaleza del sistema y puede exigir otra aprobación.

El caso también demuestra que la seguridad evoluciona después del lanzamiento. El proyecto incorporó análisis de habilidades, cooperación con proveedores de seguridad, auditorías de configuración y políticas más detalladas. Meses después seguían apareciendo técnicas evasivas. No es una contradicción, sino la dinámica normal de una plataforma que se convierte en infraestructura. La madurez no se demuestra afirmando que ya no existen fallos, sino reduciendo el tiempo entre descubrimiento, contención, corrección y aprendizaje.

4.9. La pregunta cambió de sitio

Durante la primera etapa de la inteligencia generativa preguntábamos si un modelo era capaz de responder con suficiente calidad. Cuando ese modelo puede usar herramientas, la pregunta decisiva se desplaza. Necesitamos saber dónde se ejecuta, qué identidad utiliza, qué datos alcanza, qué acciones puede realizar, qué destinos tiene disponibles y quién autorizó esa combinación.

El episodio de OpenClaw hizo visible que muchos incidentes atribuidos a la inteligencia artificial siguen naciendo de decisiones conocidas. Servicios expuestos, autenticación débil, dependencias no verificadas, secretos accesibles y

privilegios excesivos no aparecieron con los modelos de lenguaje. Lo nuevo es que un agente conecta esas debilidades y puede recorrerlas mediante instrucciones expresadas en lenguaje natural.

Por eso el concepto propietario de este capítulo no es OpenClaw ni el runtime. Es la combinación de permisos. Un inventario puede enumerar capacidades y una política puede aprobarlas, pero solo la revisión de sus relaciones revela qué rutas de daño se han creado. Gobernar un agente exige observar el conjunto cada vez que una de sus piezas cambia.

El siguiente paso será estudiar la seguridad como una disciplina más amplia. Allí importarán la inyección de instrucciones, la cadena de suministro, la identidad, la supervisión y la evidencia. Este capítulo deja una regla para todos ellos. Ningún control debe evaluarse solo por lo que bloquea de manera aislada, sino por la autoridad que permite cuando se combina con el resto del sistema.

Notas y referencias

Las referencias se comprobaron el 2 de septiembre de 2026. OpenClaw y su ecosistema evolucionan con rapidez, por lo que las cifras, las versiones y las medidas técnicas deberán revisarse antes de cerrar una edición.

→ **OpenClaw** *Presentación del proyecto. Descripción oficial de la plataforma y de su ejecución en infraestructura elegida por el usuario.*

https://openclaw.ai/blog/introducing-openclaw

→ **NIST** *NVD CVE-2026-25253. Registro público de la vulnerabilidad de robo de token de la pasarela y su gravedad.*

https://nvd.nist.gov/vuln/detail/cve-2026-25253

→ **Palo Alto Networks Unit 42** *OpenClaw's Skill Marketplace and the Emerging AI Supply Chain Threat. Investigación sobre campañas tempranas y habilidades evasivas observadas entre febrero y mayo de 2026.*

https://unit42.paloaltonetworks.com/openclaw-ai-supply-chain-risk/

→ **OpenClaw Security.** *Modelo de confianza, configuración segura, auditoría y separación de fronteras.*

https://docs.openclaw.ai/gateway/security

→ **OpenClaw Where OpenClaw Security Is Heading.** *Hoja de ruta sobre límites de archivos, salida de red, procedencia de plugins y aprobaciones.*

https://openclaw.ai/blog/where-openclaw-security-is-heading

→ **OWASP Top 10 for Agentic Applications for 2026.** *Taxonomía de riesgos para aplicaciones agénticas.*

https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/

→ **Simon Willison The lethal trifecta for AI agents.** *Formulación del patrón que combina datos privados, contenido no confiable y comunicación externa.*

https://simonwillison.net/2025/Jun/16/the-lethal-trifecta/

Nota: Enlaces comprobados el 2 de Septiembre de 2026

5

La seguridad no cabe en un solo control

Cómo proteger el recorrido completo de una aplicación de inteligencia artificial

El capítulo anterior mostró que un permiso cambia de significado cuando se combina con otros. Esa regla permite reconocer una ruta peligrosa, pero todavía falta responder una pregunta más amplia. ¿Dónde debe intervenir la seguridad para que un error del modelo, una instrucción maliciosa o una herramienta defectuosa no recorran el sistema hasta producir daño?

La respuesta no es un filtro situado delante del modelo. Una aplicación de inteligencia artificial recibe información, recupera documentos, forma un contexto, genera una propuesta, elige herramientas, ejecuta operaciones y deja registros. Cada transición abre una frontera distinta. Si la defensa se concentra en una sola, el ataque buscará las demás.

La tesis del capítulo

La seguridad de una aplicación de inteligencia artificial se construye a lo largo de todo su recorrido. Cada capa debe limitar el daño que puede causar el fallo de otra.

5.1. La seguridad clásica sigue debajo de todo

La autenticación, la autorización, el cifrado, la gestión de secretos, la validación de datos, la protección de interfaces y la actualización de dependencias no pierden valor cuando aparece un modelo. Siguen siendo el suelo de la aplicación. Si una persona accede a documentos que no debería ver o una herramienta utiliza una cuenta administrativa global, ningún detector semántico corregirá ese privilegio.

Lo nuevo surge porque el lenguaje participa en la decisión. Un texto introducido por el usuario puede contener una petición, pero un correo, una página web o un documento recuperado también puede contener frases que el modelo interprete como instrucciones. Además, la salida generada puede pasar a otra aplicación que la publique, la convierta en una consulta o la ejecute. Los controles tradicionales permanecen, aunque ahora deben rodear un componente probabilístico que interpreta contenido y propone acciones.

Esto cambia la unidad de análisis. Ya no basta con preguntar si el modelo responde de forma segura. Hay que comprobar quién inició la operación, qué información llegó al contexto, qué herramienta fue elegida, con qué credencial actuó, qué sistema aplicó la autorización y qué evidencia quedó después.

5.2. La superficie de ataque es un recorrido

Imaginemos un asistente de atención al cliente que lee un mensaje, consulta el historial de pedidos y puede preparar un reembolso. La misma tarea atraviesa varias capas. El mensaje puede estar manipulado, la búsqueda puede recuperar datos de otro cliente, el modelo puede proponer un importe incorrecto, la herramienta puede admitir más operaciones de las necesarias y la aprobación puede ocultar el destinatario real. El incidente no pertenece a un único componente, porque se forma en las conexiones entre ellos.

Este mapa permite evaluar una arquitectura y también una plataforma comercial. Lo importante no es que un producto se presente como cortafuegos de inteligencia artificial, pasarela o sistema de guardas. Lo decisivo es saber en qué transiciones puede bloquear una operación, en cuáles solo observa y qué ocurre cuando no está disponible.

Punto del recorrido	Fallo posible	Control que debe existir
Identidad	La aplicación no sabe en nombre de quién actúa	Autenticación y contexto de identidad verificable
Entrada	El usuario o una fuente externa introduce contenido malicioso	Clasificación, políticas de uso y señales de ataque
Recuperación	La búsqueda entrega información no autorizada o envenenada	Permisos antes de recuperar, separación y procedencia
Contexto	Se mezclan reglas, datos y contenido externo sin fronteras claras	Jerarquía de instrucciones y aislamiento de contenido
Salida	La respuesta contiene datos sensibles o código peligroso	Esquemas, validación y tratamiento como entrada no confiable
Herramienta	El agente elige una función innecesaria o argumentos dañinos	Lista permitida, alcance mínimo y política por operación
Ejecución	Una propuesta se ejecuta sin autorización independiente	Comprobación determinista, límites y aprobación selectiva
Registro	No hay evidencia o se conserva información sensible de más	Trazas útiles, acceso restringido, enmascarado y retención

Tabla 5.1. Un control situado en un punto no protege automáticamente los siguientes.

5.3. La entrada es una señal, no una frontera suficiente

Inspeccionar la petición del usuario reduce ataques triviales, detecta credenciales y puede activar una revisión. Sin embargo, la manipulación de instrucciones es un problema de

significado. La misma intención puede expresarse de innumerables formas, repartirse entre fragmentos o aparecer en un archivo que el usuario nunca escribió.

Por eso un detector debe tratarse como una capa imperfecta. Puede elevar o reducir la confianza, pero no demostrar que todo texto permitido sea seguro. La consecuencia práctica es importante. El sistema necesita funcionar de forma contenida incluso cuando una instrucción maliciosa atraviesa el filtro.

Los capítulos siguientes estudiarán *la inyección de instrucciones*[35] y sus variantes. Aquí basta con fijar su posición en el recorrido. Puede entrar por una conversación, un documento, una página, un correo, la memoria o la respuesta de una herramienta. La defensa no termina al clasificar el texto, porque el objetivo real consiste en impedir que ese texto adquiera autoridad.

5.4. Recuperar información exige permiso y procedencia

La recuperación de documentos plantea dos problemas independientes. El primero consiste en decidir qué puede leer

[35] *Inyección de instrucciones:* Ataque en el que contenido no confiable es interpretado por el modelo como instrucciones capaces de alterar su comportamiento o influir en las acciones del sistema.

la persona que formula la pregunta. *Si una búsqueda semántica*[36] localiza un informe de recursos humanos fuera de sus permisos y lo entrega al modelo, la aplicación puede responder con gran precisión y seguir siendo insegura.

La autorización debe aplicarse antes de que el documento entre en el contexto. Este principio puede describirse como *recuperación consciente de permisos*[37]: la búsqueda solo debe considerar desde el principio los documentos que la identidad solicitante está autorizada a consultar. Filtrar la respuesta después llega tarde, porque el modelo ya recibió la información. En entornos *multiinquilino*[38], compartidos por departamentos o clientes, esta regla exige mantener la separación durante la indexación, la búsqueda y la generación, no solo en la interfaz visible.

---

[36] *Búsqueda semántica:* Método de recuperación que localiza información por proximidad de significado y no únicamente por coincidencia literal de palabras.

[37] *Recuperación consciente de permisos:* Búsqueda que restringe desde el inicio los documentos recuperables a aquellos que la persona o identidad solicitante está autorizada a consultar.

[38] *Multiinquilino:* Entorno en el que varios clientes, usuarios o áreas comparten una misma infraestructura, manteniendo separados sus datos, permisos y recursos.

El segundo problema es la *procedencia* [39] . Un documento autorizado puede contener instrucciones ocultas, datos alterados o referencias externas preparadas para desviar la tarea. Los permisos responden quién puede ver el documento; la procedencia ayuda a saber de dónde salió y si ha cambiado. Ninguno sustituye al otro.

Dos preguntas distintas

¿Tiene esta persona derecho a recuperar el documento? ¿Puede este documento influir en las decisiones del sistema y con qué nivel de confianza?

5.5. La salida del modelo también es contenido no confiable

Una de las inversiones conceptuales más útiles consiste en desconfiar de lo que sale del modelo. Una respuesta no es una autorización y tampoco es código seguro por haber sido generada dentro de la organización. Si termina en una página, una consulta a una base de datos, una terminal o una función privilegiada, debe pasar por los mismos controles que cualquier otra entrada.

El asistente del ejemplo puede proponer un reembolso de 120 euros. Antes de ejecutarlo, un *componente*

[39] *Procedencia:* Información que permite conocer el origen, la transformación y la versión de un dato, documento o componente.

determinista[40] debe comprobar el tipo de operación, el pedido, el límite permitido, la moneda, el destinatario y la identidad que autoriza. El modelo puede explicar por qué cree que procede, pero no debe decidir por sí mismo si posee autoridad.

Esta separación también mejora la fiabilidad. Un esquema limita la forma de los argumentos, una regla impide importes negativos o superiores al máximo y una operación *idempotente* [41]evita que un reintento cobre o pague dos veces. Son defensas de software conocidas que contienen errores aunque no exista un atacante.

Principio operativo

La salida del modelo es una propuesta. El sistema final valida los argumentos, aplica la autorización y decide si la operación puede continuar.

[40] *Componente determinista:* Parte de un sistema que aplica reglas explícitas y produce resultados previsibles para las mismas condiciones, sin delegar la decisión en la interpretación probabilística del modelo.

[41] *Idempotencia:* Propiedad de una operación que permite repetirla sin producir efectos adicionales después de la primera ejecución válida.

5.6. Herramientas, aprobaciones y reversibilidad

Cada herramienta amplía el perímetro, aunque el riesgo no depende solo de cuántas existan. También importan las operaciones expuestas, los recursos alcanzables y los límites de uso. Un asistente que resume correo necesita leer mensajes; no necesita enviarlos, borrarlos ni crear reglas de reenvío. Ofrecer una interfaz completa porque ya está disponible convierte comodidad de integración en *agencia excesiva*[42].

Las aprobaciones humanas aportan control cuando se reservan para acciones de impacto y muestran información suficiente. Una persona puede revisar unos pocos reembolsos excepcionales si ve el importe, el pedido, el destinatario y el motivo. Si recibe miles de cuadros de diálogo genéricos, se convierte en un botón humano y la aprobación pierde significado.

La reversibilidad añade otra barrera. Preparar un borrador antes de publicar, mover a la papelera antes de borrar, reservar fondos antes de transferirlos o mostrar una simulación antes de modificar una base de datos reduce el coste de un error. La seguridad madura no depende de acertar siempre, sino de evitar que un fallo aislado produzca de inmediato una consecuencia irreversible.

---

[42] *Agencia excesiva:* Concesión de más funciones, permisos o autonomía de los que una tarea necesita.

5.7. Observar también crea una responsabilidad

Las *trazas*[43] permiten reconstruir qué modelo intervino, qué documentos se recuperaron, qué herramienta se llamó y qué resultado devolvió. Sin esa evidencia, una organización puede descubrir el daño sin comprender el recorrido. Las convenciones emergentes de *OpenTelemetry* [44] para inteligencia generativa ayudan a representar estas operaciones de forma coherente, aunque registrar un campo no garantiza que alguien lo supervise ni que el contenido sea seguro.

Guardar peticiones, respuestas y argumentos de herramientas crea un repositorio especialmente sensible. Puede contener datos personales, secretos comerciales, historiales médicos o credenciales introducidas por error. Por eso el registro necesita enmascarado, control de acceso, retención justificada y separación entre los metadatos útiles para investigar y el contenido completo.

---

[43] *Traza:* Registro estructurado del recorrido de una operación a través de un sistema, que permite reconstruir qué componentes intervinieron, qué acciones realizaron y qué resultados produjeron.

[44] *OpenTelemetry* es un proyecto abierto que proporciona estándares y herramientas para generar y transmitir señales de observabilidad, como trazas, métricas y registros.

El capítulo dedicado a la *observabilidad* [45]decidirá qué conservar y cómo convertir las trazas en evidencia. En este punto solo necesitamos recordar que observar no equivale a proteger. Una alerta posterior no sustituye al control que debía bloquear una transferencia antes de ejecutarse.

5.8. Las taxonomías de 2026 muestran el cambio de escala

OWASP mantiene listas diferentes para aplicaciones con modelos de lenguaje y para aplicaciones agénticas. No son normas ni inventarios exhaustivos, pero sirven para observar cómo cambia la pregunta de seguridad. La edición de agosto de 2026 para modelos conserva la atención sobre entradas, datos, consumo y salidas. La lista agéntica publicada en diciembre de 2025 para el ciclo 2026 sigue el daño a través de objetivos, herramientas, identidades, memoria y relaciones entre agentes.

La diferencia revela un desplazamiento. Cuando el sistema solo genera, preocupa lo que dice y lo que revela. Cuando recupera, importa qué puede ver. Cuando utiliza herramientas, importa qué puede hacer. Cuando conserva memoria o coordina otros agentes, también hay que vigilar

[45] *Observabilidad:* Capacidad de comprender el estado y el comportamiento interno de un sistema a partir de señales como trazas, registros, métricas y eventos.

cómo se propaga un error y cuánto tiempo puede permanecer activo.

Cambio	Pregunta que debe reabrirse
Inyección de instrucciones	Secuestro del objetivo
Divulgación de información sensible	Uso indebido y explotación de herramientas
Agencia excesiva	Abuso de identidad y privilegios
Cadena de suministro	Vulnerabilidades de la cadena de suministro agéntica
Envenenamiento de datos y modelos	Ejecución inesperada de código
Consumo sin límites	Envenenamiento de memoria y contexto
Desinformación	Comunicación insegura entre agentes
Exposición de contexto oculto	Fallos en cascada
Debilidades de vectores y representaciones	Explotación de la confianza entre personas y agentes
Tratamiento inadecuado de la salida[46]	Agentes fuera de control

Tabla 5.2. Comparación orientativa de las dos listas de OWASP para 2026. Las filas no implican equivalencia directa.

[46] *Tratamiento inadecuado de la salida:* Uso de una respuesta del modelo sin validarla antes de publicarla o entregarla a otro componente.

La tabla no debería usarse como lista de comprobación cerrada. Una misma ruta puede combinar varias categorías y un control puede reducir más de una. Su valor está en obligar a ampliar el análisis desde la respuesta hasta la trayectoria completa.

5.9. Cómo evaluar una plataforma de seguridad

Las plataformas especializadas pueden aportar detección, políticas, autorización, evaluación y observabilidad. Para compararlas conviene apartar por un momento su etiqueta comercial y dibujar el flujo real de la aplicación. Después se pregunta en qué puntos existe capacidad de bloqueo y en cuáles solo se recibe una señal.

Una plataforma no puede reparar por sí sola una arquitectura que concede privilegios administrativos, mezcla datos de clientes o ejecuta cualquier salida del modelo. Puede reforzar controles y aplicarlos de forma consistente, pero no sustituye la identidad, la autorización, la segmentación ni la responsabilidad de quienes diseñan el sistema.

Área	Pregunta que revela control efectivo
Identidad	¿Sabe qué persona, aplicación o agente inició la operación?
Entrada	¿Inspecciona también documentos recuperados y respuestas de herramientas?
Recuperación	¿Respeta permisos y separación antes de entregar contenido al modelo?
Salida	¿Valida datos estructurados y detecta información sensible antes del siguiente componente?
Herramientas	¿Puede bloquear una llamada y sus argumentos antes de la ejecución?
Aprobación	¿Muestra la operación exacta y permite aplicar reglas por impacto?
Límites	¿Impone techos de coste, tiempo, pasos, frecuencia y destinos?
Evidencia	¿Conserva la trayectoria sin duplicar innecesariamente contenido sensible?
Cobertura	¿Protege todos los modelos y herramientas o deja rutas alternativas sin control?

5.10. Defender por capas y probar el recorrido

La *defensa en profundidad*[47] parte de una suposición incómoda y útil. Cualquier capa puede fallar. El detector puede no reconocer una instrucción, la búsqueda puede recuperar un

[47] *Defensa en profundidad:* Seguridad construida mediante capas independientes para que el fallo de una no comprometa el conjunto.

documento inesperado, el modelo puede elegir mal y la persona puede aprobar con prisa. El sistema se considera robusto cuando uno de esos fallos no basta para comprometer el conjunto.

Capa	Responsabilidad dominante	Prueba que debería superar
Seguridad del software	Identidad, permisos, secretos, dependencias, interfaces y segmentación	Una cuenta o componente comprometido no obtiene acceso global
Controles de IA	Entradas, contexto, recuperación, salidas, herramientas y límites de autonomía	Un texto manipulado no adquiere autoridad para actuar
Gobernanza	Usos permitidos, propietarios, aprobaciones, proveedores y respuesta a incidentes	Cada excepción tiene responsable, alcance y caducidad
Evaluación continua	Casos normales, errores, ataques, fallos de red y cambios de configuración	El sistema se comporta de forma contenida cuando una dependencia falla

Los ejercicios de ataque también deben cambiar. Conseguir que un asistente pronuncie una frase prohibida dice poco sobre un agente empresarial. Las pruebas útiles intentan que lea datos de otro cliente, utilice una herramienta inesperada, repita una operación, revele un secreto en un registro, entre en un ciclo o siga actuando cuando recibe información parcial.

La pregunta final de cada prueba no es solo qué respondió el modelo. Es qué recorrido siguió el sistema, qué

barrera debía detenerlo, por qué no lo hizo y cuánto daño habría sido posible. Esa información convierte un fallo en una mejora verificable.

5.11. El verdadero límite está fuera del modelo

Una aplicación segura no necesita confiar en que el modelo sea imposible de engañar. Necesita conseguir que engañarlo no baste. El modelo no debería acceder a documentos fuera de los permisos del usuario, disponer de credenciales administrativas, elegir herramientas innecesarias, ejecutar directamente una operación crítica ni comunicarse con cualquier destino.

Esta es la función propietaria del capítulo dentro del libro. La seguridad es una propiedad del recorrido completo y se construye mediante defensas distribuidas. Los capítulos siguientes podrán profundizar en la inyección de instrucciones, la identidad, la cadena de suministro y la observabilidad sin volver a explicar el mapa entero.

Gobernar la inteligencia exige aceptar que una respuesta generada puede ser brillante y seguir siendo una propuesta. La autoridad pertenece a los sistemas, las políticas y las personas que deciden qué información puede entrar, qué operación puede ejecutarse y qué consecuencias deben permanecer bajo control.

Notas y referencias

Las referencias se comprobaron el 2 de septiembre de 2026. Las taxonomías de seguridad cambian con rapidez y conviene revisar sus ediciones antes de cerrar una nueva versión del libro.

→ **OWASP Top 10 for LLM Applications 2026.** *Edición publicada el 3 de agosto de 2026 y fuente de la primera columna de la tabla 5.2.*

https://genai.owasp.org/resource/owasp-genai-llm-top-10-2026/

→ **OWASP Top 10 for Agentic Applications for 2026.** *Taxonomía para agentes que planifican y actúan mediante herramientas.*

https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/

→ **NIST Adversarial Machine Learning:** *A Taxonomy and Terminology of Attacks and Mitigations. Taxonomía de ataques y mitigaciones publicada en marzo de 2025.*

https://csrc.nist.gov/pubs/ai/100/2/e2025/final

→ **NIST Generative Artificial Intelligence Profile.** *Perfil del AI Risk Management Framework dedicado a riesgos de la inteligencia generativa.*

https://doi.org/10.6028/NIST.AI.600-1

→ **OpenTelemetry Semantic conventions for generative AI.** *Convenciones abiertas para representar llamadas a modelos, agentes y herramientas en trazas.*

https://opentelemetry.io/docs/specs/semconv/gen-ai/

→ **Simon Willison The lethal trifecta for AI agents.** *Formulación del patrón que combina datos privados, contenido no confiable y comunicación externa.*

https://simonwillison.net/2025/Jun/16/the-lethal-trifecta/

Nota: Enlaces comprobados el 2 de Septiembre de 2026

Estas son las primeras 100 páginas de mi libro: "**Gobernar la inteligencia**."

Este adelanto te permitirá descubrir el enfoque de la obra y comprender los principales temas que desarrolla. El libro ha sido pensado especialmente para empresas, organizaciones y profesionales que necesitan afrontar uno de los grandes desafíos actuales: cómo integrar, organizar y gobernar el uso de la inteligencia artificial dentro de sus equipos, procesos y entornos de trabajo.

¿Te está gustando? Puedes encontrar **Gobernar la inteligencia** en Amazon, disponible en diferentes idiomas y formatos.

Si quieres conocer mis próximos libros, nuevas publicaciones y otros proyectos, visita mi web y conoce alguno de los temas relacionados con este libro y otros, a través de mi blog en diferentes idiomas.

www.miltonchanes.de

Y recuerda: si tienes una suscripción a *Kindle Unlimited*, podrás leer este y otros títulos incluidos en el programa sin coste adicional desde la aplicación de *Kindle*.

En *amazon* encontrarás otros de mis libros, por ejemplo “La conquista de la inteligencia”, un ensayo sobre tecnología y sociedad.