

Dónde puede fallar la gobernanza de la IA entre la aprobación y la ejecución

Por Stephan Pochet

La gobernanza de la IA requiere controles eficaces desde la información de origen hasta la aprobación y la ejecución. Este ensayo examina la revisión humana, los datos, los cambios del sistema, los permisos y la respuesta a incidentes.

Un asistente prepara un pago a un proveedor. Alguien comprueba la factura y lo aprueba. Antes de ejecutarse el pago, cambian los datos bancarios del proveedor.

¿Sigue siendo válida la aprobación?

Esa pregunta debe resolverse al diseñar el proceso. Si surge por primera vez durante una investigación, la organización ya está intentando reconstruir una decisión que debería haber controlado.

La gobernanza de la IA suele hacerse tangible en situaciones como esta. Una política designa a un responsable. Una evaluación describe los riesgos. Sin embargo, un cambio corriente entre la revisión y la ejecución puede dejar al descubierto un fallo que ninguno de esos documentos resuelve.

Cinco áreas de trabajo ayudan a reducir ese riesgo: la revisión humana, la gobernanza de los datos, los registros del ciclo de vida, los permisos que se hacen cumplir y la monitorización. Son asuntos conocidos. Lo difícil es conseguir que funcionen juntos en una misma operación.

Empezar por la acción

El ejemplo del pago a un proveedor es hipotético, pero ofrece un punto de partida útil. Tiene una finalidad reconocible y un resultado que importa fuera de la aplicación. El dinero llega a la cuenta prevista o no llega. La organización debe responder por ese resultado aunque hayan intervenido varios sistemas.

Empezad por delimitar el trabajo del asistente. Preparar una recomendación es una actividad. Enviar instrucciones a un servicio de pagos es otra. Un equipo que llama asistencia de pagos a ambas puede pasar por alto el momento en que la aplicación adquiere capacidad para afectar a otra persona.

Esta distinción también cambia las pruebas necesarias para aprobar el sistema. Un asistente de redacción útil puede admitir errores que una persona corrija antes de que ocurra nada. Un sistema autorizado para ejecutar necesita controles sobre la propia acción. Una respuesta correcta en un conjunto de pruebas no demuestra que respetará los

permisos del usuario cuando se conecte a una cuenta real.

Seguid también el trabajo entre departamentos. El equipo de aplicaciones puede mantener el asistente mientras finanzas controla las fichas de proveedores. Otro equipo puede gestionar las identidades. Todos pueden cumplir correctamente su cometido y dejar, aun así, un fallo entre servicios. La revisión sirve para localizar esos fallos mientras todavía se pueden abordar de manera deliberada.

Un inventario resulta útil cuando permite encontrar a quien puede responder a la siguiente pregunta. Si solo nombra un departamento, averiguad quién puede autorizar un cambio o detener una operación. Alguien debe estar localizable cuando el proceso encuentre una situación que el diseño no había previsto.

Delimitar qué se está aprobando

En el ejemplo del pago, quien revisa necesita saber exactamente qué aprueba: el proveedor, el importe, la cuenta de destino y la documentación que justifica el pago. El sistema debe establecer qué cambios invalidan esa aprobación.

De lo contrario, una persona puede aprobar una acción y acabar figurando como responsable de otra.

También necesita tiempo e información suficientes para cuestionar la recomendación. Una cola de solicitudes urgentes puede convertir la revisión en una costumbre de hacer clic. Los equipos deberían comprobar si quienes revisan detectan un importe incorrecto o un cambio de proveedor sin justificar en condiciones de trabajo realistas.

Después, probad un rechazo. Comprobad que la ejecución se detiene de verdad, incluidas las operaciones que ya estén en cola en un sistema conectado. De poco sirve registrar un rechazo si el pago se ejecuta igualmente.

Pedir la intervención de una persona tiene un coste

La revisión humana tiene un coste que el equipo debe reconocer. Cada solicitud de aprobación consume atención. Si la aplicación pide confirmación para pasos rutinarios y fácilmente reversibles, quienes revisan pueden prestar menos atención cuando llegue la petición importante. El diseño debe distinguir esas situaciones según el riesgo de la acción.

Considerad qué puede comprobar la persona por su cuenta. Mostrar una explicación generada por IA junto a una recomendación generada por IA puede dar una apariencia de corroboración cuando ambas dependen de la misma fuente defectuosa. Facilitad el acceso al registro original y haced visibles las discrepancias sin resolver. Nadie debería descubrir después de aprobar que la documentación estaba incompleta.

La responsabilidad también necesita un límite práctico. Quien revisa no puede aceptar de forma consciente todas las consecuencias posibles de un sistema que no puede inspeccionar. Definid la decisión que le corresponde y las comprobaciones que se realizan

en otros puntos. Así, la aprobación resulta útil tanto para quien la da como para quien examine el expediente después.

Cuando esa persona no esté disponible, una solicitud importante debe seguir una alternativa definida. Puede quedar pendiente o pasar a otra persona autorizada. La elección depende del proceso, pero el silencio no debe convertirse discretamente en consentimiento. La organización necesita decidir qué retraso puede tolerar antes de que la presión por terminar imponga esa decisión informalmente.

La fuente importa tanto como la respuesta

Un asistente puede encontrar los datos de un proveedor en una factura o recuperarlos de un documento compartido. Esas fuentes no tienen necesariamente la misma autoridad.

Alguien debe decidir qué registro es el válido para el pago y quién puede modificarlo. Es una decisión de gobernanza de los datos con una consecuencia operativa inmediata. Si una factura contradice la ficha aprobada del proveedor, el proceso necesita una forma definida de resolver la discrepancia.

El mismo problema aparece en aplicaciones cuyas consecuencias parecen menos importantes. Un asistente interno puede responder con seguridad a partir de un manual que se sustituyó el mes pasado. Borrar el archivo antiguo de una carpeta compartida puede dejar una copia en el índice de recuperación. La corrección solo está terminada cuando el equipo comprueba que la aplicación ha dejado de utilizarlo.

La responsabilidad, el acceso, la vigencia y los procedimientos de corrección importan. Determinan si quien revisa puede confiar en la información que recibe.

La corrección tiene que llegar a la aplicación

Responsabilizarse de los datos incluye poder corregirlos. Supongamos que finanzas confirma que la ficha de un proveedor es incorrecta. El equipo debe averiguar dónde la obtuvo la aplicación y si otras recomendaciones pendientes dependen de la misma información. Corregir el registro autorizado es el comienzo del trabajo.

Las copias complican la respuesta. Una caché puede seguir proporcionando un valor anterior. Un índice de recuperación puede actualizarse según un calendario. Una aprobación pendiente puede contener una recomendación anterior a la corrección. El equipo debe decidir qué elementos invalidar y cómo comprobar que la corrección les ha llegado.

El acceso plantea un problema parecido. Un empleado puede perder el permiso para consultar un documento mientras el índice del asistente sigue mostrando su contenido. Comprobar el acceso solo al abrir el chat no permite establecer si cada elemento recuperado está autorizado. Probad con cuentas que tengan responsabilidades diferentes e incluid un cambio de puesto.

Mantened la investigación proporcionada. No hace falta reunir toda la información disponible sobre un usuario para establecer qué documento respaldó una recomendación. Registrad lo necesario para entender y corregir la decisión, con restricciones adecuadas. Una pista de auditoría que difunde información confidencial sin necesidad crea otro problema de gobernanza.

La aprobación corresponde a un sistema concreto

El nombre de un modelo solo cuenta parte de la historia al auditor. Las instrucciones de la aplicación, las fuentes consultadas, las herramientas conectadas y los permisos también influyen en lo que puede hacer.

Supongamos que se aprobó un asistente para preparar instrucciones de pago. Una versión posterior incorpora una herramienta que permite enviarlas para su ejecución. El modelo puede seguir siendo el mismo, pero la aplicación tiene ahora otra capacidad de actuación. Ese cambio debe llegar a quien se encarga de evaluarlo y aprobarlo.

Un inventario identifica la aplicación y a su responsable. Un registro ayuda a seguir las versiones de los modelos. Los registros de despliegue vinculan esos datos con la configuración que está funcionando. En conjunto, deberían permitir al equipo establecer qué se aprobó y si el sistema respetó esas condiciones.

Hay límites. Un proveedor externo puede no facilitar suficiente información sobre las versiones para reconstruir todos los aspectos de un resultado anterior. Hay que dejar constancia de esa limitación. No se debe prometer una pista de auditoría que el servicio no puede ofrecer.

Decidir qué cambios exigen una nueva revisión

El proceso de publicación de versiones necesita criterios para volver a evaluar la gobernanza. Sustituir un modelo fundacional es un caso evidente. Ampliar el grupo de usuarios puede importar tanto como eso. También conectar un nuevo repositorio documental o permitir que la aplicación funcione sin una persona presente.

Evalúad esos cambios por lo que permiten hacer al sistema y por las personas a las que afectan. Una versión menor de software puede introducir un cambio importante de autoridad. En cambio, una actualización de mantenimiento puede mantener intactos el uso aprobado y los límites de control. Tratar todos los cambios como si fueran iguales puede saturar a quienes revisan sin mejorar su atención a los que importan.

Dejad constancia de la decisión y de su fundamento. Una investigación posterior debería poder distinguir un cambio evaluado y aceptado de otro que pasó inadvertido. Esto también ayuda al equipo que herede la aplicación. Necesita entender por qué existe una restricción antes de decidir si la elimina.

La vuelta a una versión anterior merece un ensayo propio. Restablecer un modelo anterior no restaura necesariamente la aplicación previa si han cambiado la base de proveedores, los permisos o el servicio conectado. El plan de recuperación debe identificar qué se puede

restaurar con seguridad y qué requiere un proceso manual. Un botón para volver atrás es una promesa que hay que probar.

Hacer cumplir la regla donde puede incumplirse

Si el asistente nunca debe cambiar la cuenta bancaria de un proveedor, hay que restringir esa acción en el sistema conectado. Una frase en sus instrucciones no es un control de autorización suficiente.

Las recomendaciones de OWASP sobre autonomía excesiva son prácticas: limitar las funciones y los permisos disponibles y hacer cumplir las autorizaciones en los sistemas posteriores. La aprobación humana es otro control, especialmente para acciones de gran impacto. Cada medida cumple una función diferente. [1]

En el proceso de pago, probad las vías que permiten eludir el procedimiento previsto. ¿Puede otra herramienta realizar el mismo cambio prohibido? ¿Una cuenta de administrador evita la restricción? ¿Qué ocurre si el servicio de aprobación no está disponible?

Las excepciones también requieren atención. Cada permiso temporal debe tener un responsable que deje constancia de por qué se concedió. Fijad su fecha de caducidad al otorgarlo. De lo contrario, la siguiente revisión puede descubrir que una solución de emergencia se ha convertido en el funcionamiento habitual.

Las excepciones muestran cómo se trabaja de verdad

Imaginad que un pago legítimo se bloquea repetidamente. Finanzas necesita completarlo, el equipo de aplicaciones está bajo presión y un administrador puede ampliar el acceso en unos minutos. El atajo técnico es fácil. La cuestión de gobernanza es qué ocurre con ese acceso cuando se ha resuelto el problema inmediato.

El registro de la excepción debe permitir a quien lo revise después entender su alcance y la condición para darla por terminada. Quien la concede también debe saber qué protecciones siguen vigentes. Si se elimina una restricción por completo, llamar temporal al cambio no reduce la exposición mientras esté activo.

Revisad también las excepciones recurrentes. Las peticiones repetidas del mismo apaño pueden indicar que el proceso aprobado no responde a una necesidad legítima. Puede que haya que rediseñarlo y evaluarlo correctamente. Mantener a los usuarios dependientes de excepciones continuas hace que la política formal describa peor cómo se realiza el trabajo.

Algunas excepciones deben rechazarse. Si el equipo no puede establecer quién recibirá un pago o hacer cumplir la autorización necesaria, quizá no proceda completar la operación mediante el asistente. Una vía manual puede mantener la actividad mientras se resuelve el problema técnico. Esa vía también debe tener un responsable.

Dar a la alerta un destino útil

La monitorización debe ayudar a alguien a tomar una decisión. Una alerta sobre un intento de uso no autorizado de una herramienta necesita un responsable que pueda investigarlo y, cuando sea necesario, restringir o detener el proceso.

Los registros de actividad y las trazas pueden ayudar a establecer qué herramientas se utilizaron, qué devolvieron y qué configuración estaba activa. No revelan automáticamente el razonamiento interno del modelo. También pueden contener información confidencial, por lo que las pruebas necesitan sus propios controles de acceso y conservación.

La guía sobre observabilidad de la IA, en inglés ofrece un punto de partida. La prueba operativa es sencilla: introducir un fallo controlado y seguirlo desde la detección hasta el escalado, la contención y la recuperación. Incluid a quien cubre una ausencia o atiende una incidencia fuera del horario habitual.

El marco de gestión de riesgos de la IA del NIST incluye la monitorización continua, la respuesta y la recuperación en su función MANAGE. Esos objetivos deben convertirse en responsabilidades que las personas puedan ejercer. [2]

Decidir qué puede demostrar una señal

Una medida de monitorización necesita interpretación. Un aumento de llamadas a herramientas rechazadas puede indicar intentos de uso indebido. También puede deberse a un cambio legítimo que las reglas de permisos no contemplan. La señal da al equipo un motivo para investigar; no determina la explicación.

Un panel sin alertas puede resultar engañoso por otro motivo. La aplicación puede haber dejado de enviar eventos. Comprobad también la vía de monitorización para no confundir la ausencia de alertas con pruebas de que todo ha funcionado. Cuando sea posible, comparad los registros del asistente con los de la acción en el sistema conectado.

La respuesta debe tener en cuenta el trabajo terminado. Detener el asistente impide parte de la actividad futura, pero no deshace un mensaje externo ni recupera un pago ya enviado. El proceso de incidentes debe distinguir la contención de la corrección e identificar quién puede evaluar las consecuencias para las personas o los registros afectados.

Aquí la gobernanza se convierte en una responsabilidad continua. El equipo aprende algo sobre un fallo y debe decidir qué cambia en el proceso. Puede ser un límite de permisos, la documentación presentada a quien revisa o las condiciones que exigen una nueva aprobación. Cerrar el incidente sin considerar esos cambios deja abierta la misma vía para el siguiente fallo.

Hacer que otra persona pueda entender las pruebas

Un registro de gobernanza debe resultar comprensible para alguien que no estaba presente cuando se tomó la decisión. Es una prueba exigente. Los equipos suelen apoyarse en un conocimiento compartido que nunca llega al expediente: una limitación mencionada en una reunión, una suposición sobre quién tendría acceso o la promesa de mantener desactivada una función.

Anotad las condiciones que importan para la aprobación. Si el asistente puede redactar pero no ejecutar, expresadlo de forma que el equipo de despliegue pueda comprobarlo. Identificad las pruebas de ese límite y qué obligaría a reconsiderarlo. El registro debe ayudar a formular una pregunta precisa sobre el sistema activo.

Evitad convertir el expediente en un archivo de todo lo que produjo el proyecto. Una carpeta grande puede ocultar la ausencia de una decisión clara. Organizad las pruebas en torno al uso aprobado y haced que sus limitaciones sean fáciles de encontrar. Conservad los resultados de las pruebas con contexto suficiente para entender qué se comprobó y qué quedó fuera del alcance.

La misma disciplina ayuda cuando cambia el personal. Un nuevo responsable debería poder averiguar por qué existe un control sin reconstruir la historia del proyecto a partir de mensajes. También debería poder identificar una suposición que ya no se cumple. La gobernanza se vuelve frágil cuando mantener un sistema depende de recordar quién sabía la respuesta el año pasado.

Seguir una decisión hasta el final

Una revisión útil puede empezar con una sola operación. Seguidla desde el documento de origen hasta la recomendación, la aprobación, la ejecución y el registro final. Cambiad un detalle importante durante el proceso. Dejad un componente fuera de servicio. Pedid a quien revisa que rechace la acción.

El ejercicio suele revelar preguntas que las revisiones por separado pasan por alto: quién resuelve los registros contradictorios, qué cambios requieren una nueva aprobación y quién tiene autoridad para intervenir cuando llega una alerta.

El pago es un ejemplo. La misma revisión puede aplicarse a un asistente que cierra un expediente, modifica una ficha de cliente o envía un mensaje externo, con controles proporcionados a las consecuencias.

La organización debería poder explicar cómo se produjo la acción y por qué estaba permitida. Sobre todo, las personas que gestionan el proceso deberían poder impedir una acción que han decidido que no debe realizarse.

Fuentes

[1] OWASP: Excessive Agency.

[2] NIST: AI Risk Management Framework Core.

Los supuestos y las prácticas de revisión son ejemplos de aplicación propuestos en este artículo.

Preguntas y respuestas

¿Qué hace eficaz la aprobación humana?

Quien revisa necesita una acción concreta, documentación fiable, tiempo suficiente y autoridad para rechazarla. Los cambios importantes deben activar una nueva evaluación y el rechazo debe impedir la ejecución.

¿Qué diferencia hay entre un inventario de IA y un registro de modelos?

El inventario identifica las aplicaciones, sus finalidades y sus responsables. El registro sigue las versiones de los modelos. Los registros de despliegue los vinculan con la configuración activa.

¿Dónde se deben aplicar los permisos de una aplicación de IA?

El sistema conectado debe hacer cumplir la autorización de la acción solicitada. Las instrucciones al modelo no bastan como límite de permisos.

¿Los registros explican el razonamiento interno del modelo?

No. Los registros y las trazas pueden recoger acciones observables y datos de configuración, pero no revelan automáticamente el razonamiento interno. Las pruebas también necesitan controles de acceso y conservación.