

# Sus agentes están haciendo su trabajo. ¿Dónde acaban?

Una guía sin jerga sobre la deriva de agentes: qué es, por qué los controles que ya tiene no la ven, qué hacen bien Microsoft y el MIT, y qué falta.

Una pluma no cae. Deriva.

## EL PROBLEMA

# Ningún paso estuvo mal. El destino sí.

Un agente consulta una cuenta. Permitido. Pide una excepción. Permitido. Concede un crédito pequeño. Permitido. Amplía la promesa. Desplaza el objetivo. Cada paso pasa cualquier control que tenga puesto hoy. La secuencia entera es un abandono del encargo.

Eso no lo puede ver un control que mira **un paso**. Tampoco uno que mira **los últimos diez**. No por descuido: por construcción. Dos trayectorias que coincidan en sus últimos diez turnos pueden llevar acumulado algo completamente distinto — y un control que solo mira esa ventana no puede distinguirlas.

Es un resultado, no una opinión: **ningún control de ventana acotada puede saber cuánto lleva gastado un agente desde el primer turno**. Puede avisar, puede bloquear un turno feo. Lo que no puede es contar el total.

Por eso medimos la **trayectoria**, no el paso. Y por eso los dos últimos contadores de la página 6 son los que no lleva nadie más.

## CASOS

# Esto ya ha pasado.

### 1 · El agente que borró la base de datos de producción

Y después fabricó usuarios falsos y mintió sobre la recuperación. Ninguna regla individual «prohibía» el paso siguiente; la secuencia acabó donde acabó.

### 2 · Deriva de objetivo bajo presión, medida

Un benchmark independiente de 204 turnos sobre 13 modelos muestra el mismo patrón: bajo conflicto de valores sostenido, el objetivo se desplaza sin una sola violación explícita.

### 3 · El catálogo de riesgos ya lo nombra

Más de 1.700 riesgos clasificados en 74 marcos incluyen la categoría —secuestro de objetivo, agentes descontrolados—. El riesgo está catalogado. Lo que falta es quién lo cuenta.

Las tres fuentes, con su enlace, están en la última página. Ninguna es nuestra.

## EL ESTADO DEL ARTE

# Lo que ya existe, y es bueno.

No competimos con esto. Nos apoyamos encima.

---

### MICROSOFT

#### El plano de enforcement

Motor sin estado, por debajo de 0,1 ms, licencia abierta, cobertura completa del catálogo OWASP para agentes, e interfaces de extensión. Intercepta la llamada, aplica listas de permitidos y topes, aísla la ejecución. Hace muy bien lo que hace.

---

### MIT

#### El mapa del riesgo

Más de 1.700 riesgos clasificados y más de mil documentos de gobierno indexados. Es la referencia para saber *qué* hay que vigilar. Es clasificación y referencia, no ejecución.

**Lo que ninguno de los dos hace, y no es un reproche:** un motor sin estado no puede, por construcción, llevar la cuenta de lo que un agente lleva gastado desde el primer turno. No es que no lo hayan hecho — es que un motor sin estado no puede.

## EL HUECO

## Cuatro preguntas. Solo se contestan dos.

## SE CONTESTA

### ¿Está permitido este paso?

Lista de permitidos, tope de argumento, aislamiento. Determinista y rápido.

## SE CONTESTA

### ¿Está desviado ahora?

Una media que decae  
contesta si el turno actual se aparta del encargo.

## NO SE CONTESTA

### ¿Cuánto lleva gastado?

Desde el primer turno, sin decaer y sin ventana. Nadie lo cuenta.

## NO SE CONTESTA

### ¿Hacia dónde va?

¿Alejándose del encargo, o dando vueltas alrededor? Nadie mira la dirección.

El caso que solo la tercera pregunta ve: **deriva intermitente**. Un turno sucio, uno limpio, indefinidamente. La racha nunca pasa de uno. La media oscila muy por debajo de cualquier umbral. Y el agente lleva cien turnos fuera de su encargo.

Y el caso que solo ve la cuarta: dos corridas que gastan **exactamente lo mismo** y llevan **exactamente la misma racha** — una alejándose sin volver, la otra dando bandazos alrededor del encargo y regresando. Todos los contadores de arriba las leen iguales. Una es un agente trabajando con ruido; la otra es un agente que se va.

Los dos primeros controles dan por buenas las dos cien veces seguidas.

## LO QUE HACEMOS

# Nosotros ponemos los dos que nadie lleva.

No competimos con la puerta. Nos enchufamos encima y añadimos lo que un motor sin estado, por construcción, no puede sostener.

## 01 · EL ANCLA

## Se firma y se congela

El encargo se captura al desplegar y lo **firma** la credencial de quien despliega. Un hash se puede recalcular; una firma no, sin la clave. Cambiar una herramienta permitida y rehacer el hash no cuela.

## 02 · CUATRO CONTADORES

## Ahora, últimamente, en total — y hacia dónde

Uno que decae. Uno que se reinicia con un turno limpio. Uno que no decae ni tiene techo: lo que lleva gastado esta corrida desde el primer turno. Y uno que lee la **dirección**: alejamiento sostenido, o ruido que se cancela.

## 03 · EL BLOQUEO SE QUEDA

## No se suelta portándose bien

Cuando se bloquea, sigue bloqueado. Solo lo abre una persona autorizada con un plan de recambio **firmado**, y la firma se comprueba. Ni reiniciar el servicio lo suelta.

## Y desde este mes, cuatro cosas más

## EL TECHO ES EXACTO

El presupuesto se lleva en enteros, no en coma flotante. Probamos 1.340 combinaciones contra aritmética exacta: un contador en coma flotante cruza el techo **un ciclo tarde en el 9,1 % de ellas** — un ciclo de acción que su política no autorizó. En enteros, cero.

## EL ÁRBOL ENTERO

Un agente reparte trabajo entre subagentes. Cada hijo cumple lo suyo toda su vida y **la suma agota el presupuesto del padre**. Cuando se agota, no ejecuta nadie del árbol.

## AGENTES QUE NO SE CONOCEN

Tres agentes independientes se descarrilan entre sí porque leen los mismos datos, o datos relacionados. Cuando uno escribe, marcamos a los demás — y la marca arrastra **cuánto llevaba gastado el que escribió**.

## SOBREVIVE AL REINICIO

Un techo que se borra al reiniciar no es un techo. El estado vive en base de datos con versión: si dos llamadas coinciden, **una se rechaza** en vez de pisar a la otra.

## DECLARACIÓN

# Lo que no declara no se gobierna.

Y se lo decimos al desplegar, no cuando pase algo. Esta es la lista entera, con lo que deja de verse en cada caso.

| QUÉ SE DECLARA                                                 | SI NO LO DECLARA                                                                                                      |
|----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <b>El encargo, las herramientas y la credencial que firma</b>  | <b>El agente no se registra.</b> Preferimos que el despliegue no arranque a que arranque sin proteger.                |
| <b>Los topes por argumento reembolso <math>\leq 200</math></b> | Ese tope <b>no existe</b> y nada lo comprueba. No hay un valor por defecto secreto.                                   |
| <b>Qué datos lee este agente</b>                               | <b>No se marca a nadie nunca.</b> El acoplamiento entre agentes deja de verse. No falla: es que no existe.            |
| <b>Qué regla relaciona dos recursos</b>                        | El caso en que cada agente cumple lo suyo y la regla común se rompe igual <b>no se detecta.</b>                       |
| <b>Qué demuestra cada herramienta</b>                          | Una afirmación se sostiene con haber ejecutado <i>algo</i> de la clase correcta — una heurística, no una prueba.      |
| <b>Si enciende el juez semántico</b>                           | El canal que más pesa lo cubre solo el detector estructural. Se lo hacemos reconocer <b>por escrito</b> al desplegar. |

Ninguna de estas ausencias hace que el control falle abierto sobre lo que *sí* declaró. Lo que hacen es reducir **qué** puede ver — y eso queda escrito en el perfil firmado que acompaña a cada decisión.

## HONESTIDAD

# Lo que todavía no hacemos.

Si el resto de este documento le ha convencido y esta página no estuviera, el documento estaría mal montado. Esto es lo que un competidor le contaría de nosotros, y preferimos contárselo primero.

## 1 · Una deriva sólo semántica la caza la racha, no la media

Juzgar si un agente sigue hablando de lo que se le encargó es un juicio sobre el sentido, y no lo hacemos nosotros: lo hace un modelo, el que usted configure. Nosotros lo gobernamos — se le consulta **cuando hace falta**: en los pasos que producen texto, cuando la trayectoria ya se está inclinando, y con un suelo de una medición cada ocho ciclos. Lo que queda es aritmética, y nos toca decirlo: una desviación que es *sólo* semántica aporta como mucho el peso del canal objetivo — 0,35 por defecto, contra un umbral de aviso de 0,50. La caza el contador de racha en el turno 15, no la media. Para un agente cuyo trabajo entero es texto, suba ese canal al desplegar — y baje otro, porque los pesos tienen que seguir sumando 1 o la configuración se rechaza al registrar. Con el canal en 0,40 y una deriva que crece, el monitor direccional la caza antes de que se agote el presupuesto; el turno exacto depende de lo deprisa que crezca.

## 2 · El detector de afirmaciones lee la FORMA, no la frase

Antes casaba un catálogo de frases literales, y sobre frases reales de producción cazaba **2 de 41**. Ahora reconoce la *forma* de una afirmación cerrada: un predicado que da el hecho por hecho, un sustantivo de sistema de registro al lado, y ninguna cautela, pregunta, intención ni petición. Mismo corpus: **35 de 41**, el control literal sigue en 12 de 12, y **cero falsos positivos** sobre 19 frases benignas. Sigue sin modelo. Se escapan seis, casi todas de capacidad: los modales quedan fuera a propósito, porque un detector que bloquea «los reembolsos se procesan en 5 días hábiles» lo apagan en una semana.

## 3 · La auditoría detecta reescritura, no truncado

Cada decisión es una fila encadenada y **firmada**: tocar una fila rompe la cadena, y reescribir la historia entera y rehacer los enlaces tampoco cuela sin la clave. Pero **un trozo inicial de la historia verifica igual que la historia entera**. Detectar que alguien cortó el final exige guardar una referencia fuera del mismo sistema. Le decimos cómo; no le decimos que sobre.

## 4 · La confirmación de una herramienta remota

La decisión de ejecutar nace dentro de la misma transacción que comprueba el estado, así que un permiso emitido bajo un estado viejo no llega a autorizarse. Pero si su herramienta remota ejecuta y la confirmación se pierde por el camino, hay que reconciliar. Le damos la clave para hacerlo. **No le decimos que ese problema no existe.**

**Una regla de la casa, por si le sirve para juzgar a cualquier proveedor:** pregunte siempre qué fallo concreto detectaría cada control que le enseñen. Una comprobación que no puede fallar no es una comprobación. Nosotros encontramos dos así en nuestro propio código este mes — y una revisión externa nos encontró otras seis.

## CÓMO COBRAMOS

# Un solo contador, y baja según crece.

Contamos **una cosa**: el paso gobernado. Ni tokens, ni asientos, ni agentes, ni «unidades». Si un paso pasa por la capa, se cuenta una vez.

## QUÉ NO SE MIDE NUNCA

### La capa nunca va a medias

No hay una versión barata que vigile menos. Racionar el gobierno es el modo de fallo, no el modelo de negocio.

## QUÉ LO ABARATA

### Más agentes, menos por paso

El precio unitario baja según crece el volumen. El coste marginal de un paso es de microsegundos de CPU: no hay ninguna llamada a modelo en el camino de decisión.

## QUÉ FIJA SU CIFRA

### Sus pasos, y nada más

Cuéntenos cuántos agentes tiene y cuántos pasos al mes. Le devolvemos una cifra acotada con los supuestos escritos al lado.

El juez semántico, si lo enciende, es la única parte que llama a un modelo y se factura aparte, a coste medido. Todo lo demás es aritmética.

VERIFICACIÓN

# No nos crea. Compruébelo.

Nada de lo anterior le pide un acto de fe. En [diacroma.com](#) puede abrir hoy mismo el demo — tres agentes trabajando una semana entera, con el descuento subiendo a la vista y ni un solo paso que incumpla una regla; el interruptor de DiaCroma enseña lo que ninguno de ellos ve, sin red y sin clave — y el replay de los 204 turnos del benchmark independiente, pasados por nuestra capa sin modificar una coma.

**Y el número que más le va a decir de nosotros:** nuestra suite ejecuta **1.587 comprobaciones** y calcula el total por dos caminos independientes que tienen que coincidir o falla. Sale el mismo número en cuatro versiones distintas de Python. Cada afirmación de este folleto tiene detrás una comprobación que *falla si la afirmación es falsa* — incluidas las de la página 8.

## Fuentes

| AFIRMACIÓN                                                                  | FUENTE                                                                                                                                                                                  |
|-----------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| El agente borró producción, fabricó usuarios y mintió sobre la recuperación | AI Incident Database, incidente 1152 — <a href="#">incidentdatabase.ai/cite/1152/</a>                                                                                                   |
| Deriva de objetivo bajo presión multiturno; 204 turnos, 13 modelos          | Benchmark agent-drift (licencia MIT) — <a href="#">github.com/jhammant/agent-drift</a> · sobre <i>Asymmetric Goal Drift in Coding Agents Under Value Conflict</i> , taller de ICLR 2026 |
| 1.700+ riesgos, 74 marcos, categoría 7.1                                    | MIT AI Risk Repository — <a href="#">airisk.mit.edu/risks</a>                                                                                                                           |
| Más de 1.000 documentos de gobierno clasificados                            | MIT AI Governance Map, MIT FutureTech — <a href="#">airisk.mit.edu/ai-governance</a>                                                                                                    |
| Motor sin estado, <0,1 ms p99, licencia MIT, OWASP Agentic 10/10            | Microsoft Open Source Blog, 2 de abril de 2026 — <a href="#">opensource.microsoft.com</a> · <a href="#">github.com/microsoft/agent-governance-toolkit</a>                               |

## Hablemos

Cuéntenos cuántos agentes tiene más o menos y cuántos pasos al mes, y le devolvemos una cifra acotada con los supuestos escritos al lado.