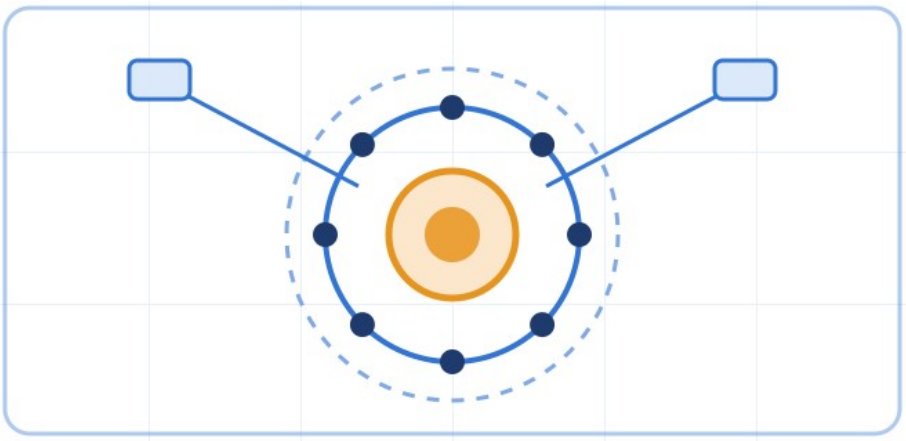
Scrum Manager®
Uncovering better ways

State of the art

Harness Engineering

Mapa de referencia del conocimiento profesional vigente



Actualizado a junio de 2026

Scrum Manager® · Skill Arena

Sobre este documento

Este documento es un mapa curado de los conocimientos, prácticas y modelos empleados profesionalmente a la fecha de su publicación para adoptar agentes de IA con rigor profesional: recubrir un modelo con guías y sensores que conviertan su producción en trabajo fiable, verificable y útil, sea cual sea la disciplina. Para cada uno, este mapa lo sitúa, indica su grado de adopción en la práctica profesional actual y orienta sobre dónde encontrar información de referencia.

Úsalo como observatorio y punto de referencia para contrastar si el conocimiento profesional que empleas está alineado con la práctica actual y en vanguardia.

Se complementa con un documento de desarrollo que profundiza en cada concepto y con la plataforma de entrenamiento y evaluación en Skill Arena. En el área Harness Engineering puedes contrastar tu nivel de conocimiento y, si lo superas, obtener un diploma que acredita curricularmente la solvencia y vanguardia profesional en esta área.

Estado del conocimiento

El conocimiento sobre Harness Engineering evoluciona de forma extremadamente rápida: la disciplina apenas existía hace año y medio y su vocabulario se ha estabilizado a lo largo de 2026 en torno a unas pocas publicaciones de referencia. Su núcleo conceptual (el marco de guías y sensores, los principios de contexto y verificación) ya está asentado, pero los patrones de la frontera — verificación sustantiva, arquitecturas multiagente, plantillas reutilizables— se redefinen casi cada mes. En los distintos apartados del documento, las etiquetas (ESTABLECIDO, EN CONSOLIDACIÓN, EMERGENTE) ayudan a identificar la madurez de cada concepto:

ESTABLECIDO consenso asentado; conocimiento que se da por necesario.

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

EMERGENTE frontera reciente; alta relevancia y alta volatilidad.

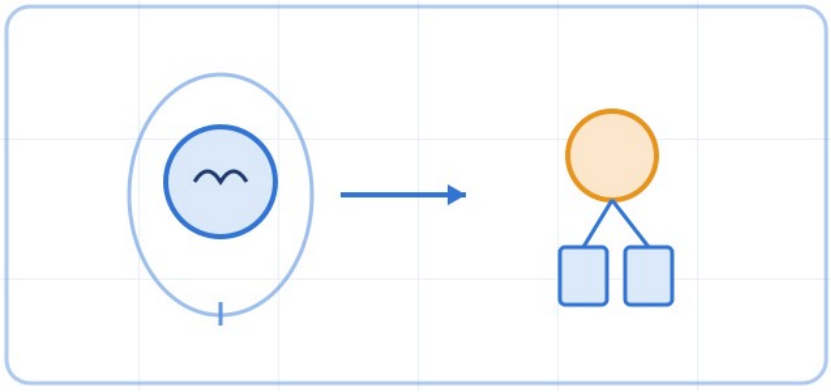
Cómo leer este mapa

El mapa agrupa el conocimiento en seis bloques que van del cambio de paradigma a la práctica de montaje, los patrones de frontera y el criterio profesional que no se delega. Cada puntero indica qué es, por qué importa ahora y dónde acudir para profundizar. Las etiquetas conservan su significado estándar: aquí, un núcleo conceptual ya asentado convive con patrones que aún se están definiendo, de modo que verás los tres estados representados.

El alcance es deliberadamente transversal. Harness Engineering nació en el desarrollo de software, pero su marco aplica a cualquier trabajo profesional asistido por agentes: análisis, consultoría, investigación, asesoría legal, periodismo, escritura, gestión o enseñanza. Los ejemplos alternan entre código y trabajo del conocimiento para no atar el concepto a una sola disciplina.

Bloque A — El cambio de paradigma: Agente = Modelo + Arnés

Antes de construir nada conviene entender por qué un modelo de frontera, por brillante que sea, no basta para entregar trabajo profesional fiable, y qué es exactamente eso que lo recubre.



Sin un entorno operativo, la inteligencia del modelo no puede actuar; el arnés es el cuerpo que convierte razonamiento en trabajo verificable.

Inteligencia frente a agencia

ESTABLECIDO

La inteligencia es una propiedad entrenada del modelo: razonar, conectar conceptos, sintetizar, planificar. La agencia es la capacidad de actuar sobre el mundo: leer documentos, ejecutar acciones, modificar artefactos, mantener coherencia entre sesiones y producir entregables verificables. Un modelo sin entorno operativo razona y opina, pero no puede comprobar nada.

Por qué está aquí ahora. Es la distinción fundacional de la disciplina: separa lo que entrena el fabricante de lo que tú construyes alrededor, y explica por qué dos profesionales con el mismo modelo entregan resultados de calidad muy distinta.

Dónde mirar. Böckeler, Harness engineering (martinfowler.com) · Mollick, One Useful Thing

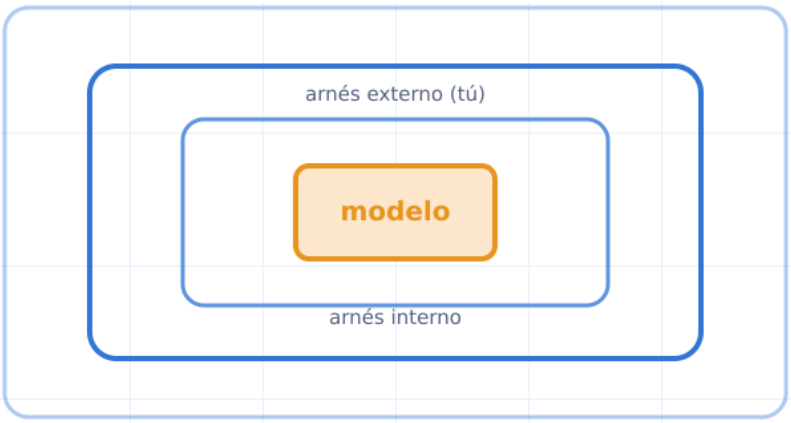
La ecuación Agente = Modelo + Arnés

ESTABLECIDO

La fórmula canónica, popularizada por Birgitta Böckeler (Thoughtworks) y adoptada por la industria en 2026, define el arnés como todo lo que no es el modelo: las instrucciones que lee al empezar, los documentos a su alcance, las herramientas que puede invocar, las políticas de permisos, los bucles de orquestación, la memoria persistente y los sensores de verificación. El modelo es un commodity; el arnés es la ventaja competitiva.

Por qué está aquí ahora. Da el vocabulario común del campo. Sin un modelo mental compartido de qué es y qué no es el arnés, las prácticas que vienen después no tienen dónde anclarse.

Dónde mirar. Böckeler, Harness engineering (martinfowler.com) · Thoughtworks · What is harness engineering



Tres círculos concéntricos: el modelo que entrena el laboratorio, el arnés interno del fabricante y el arnés externo que construyes tú.

Arnés interno frente a arnés externo

ESTABLECIDO

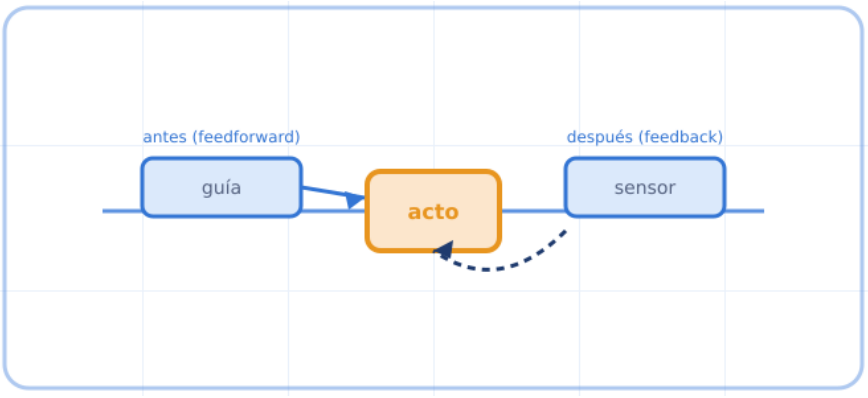
Böckeler dibuja el arnés como círculos concéntricos. El arnés interno lo construye el fabricante alrededor del modelo (interfaz, memoria, búsqueda, bucle de orquestación, ejecutores de herramientas, sistema de permisos): tú lo usas, no lo construyes. El arnés externo lo construyes tú sobre lo anterior: instrucciones, documentos, skills, conectores, verificaciones y plantillas.

Por qué está aquí ahora. Delimita la responsabilidad práctica. El grueso del valor que un profesional extrae de los agentes proviene de hacer bien el arnés externo, no de comprender el interno; saber dónde acaba uno y empieza el otro evita esfuerzo mal dirigido.

Dónde mirar. Böckeler, Harness engineering (martinfowler.com) · Anthropic · Effective harnesses for long-running agents

Bloque B — Las dos columnas: guías y sensores

El marco de Böckeler descompone el arnés externo en dos tipos de control que actúan en momentos distintos del ciclo del agente. Entender ambos, y por qué hacen falta los dos, es el corazón operativo de la disciplina.



Las guías orientan antes de actuar (feedforward); los sensores observan después y permiten autocorrección (feedback).

Guías: controles antes del acto

ESTABLECIDO

Las guías (controles feedforward) anticipan el comportamiento del agente e intentan dirigirlo antes de actuar, aumentando la probabilidad de acertar a la primera. Son lo que el agente lee al empezar y lleva consigo mientras opera: archivos de instrucciones, documentación de método o arquitectura, glosarios, guías de estilo, plantillas estructurales, corpus de referencia y skills reutilizables.

Por qué está aquí ahora. Es la mitad del marco canónico. Un agente sin guías improvisa convenciones en cada sesión; con buenas guías, parte de un estado conocido y trabaja alineado con tus reglas.

Dónde mirar. Böckeler, Harness engineering (martinfowler.com) · AGENTS.md · estándar abierto

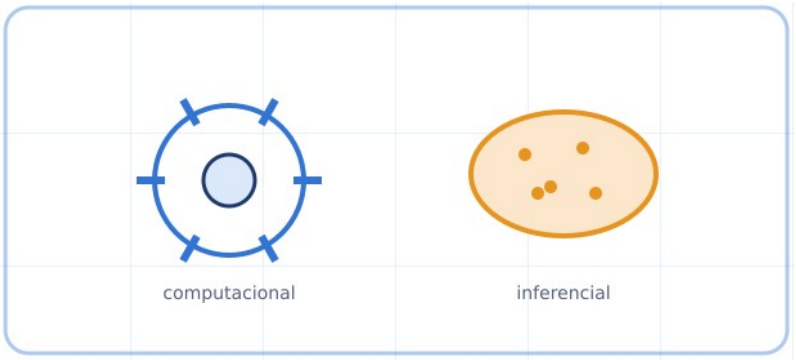
Sensores: controles después del acto

ESTABLECIDO

Los sensores (controles feedback) observan al agente después de producir algo y le permiten autocorregirse. Los buenos sensores devuelven señales accionables, optimizadas para que las consuma otra IA: en lugar de un fallo críptico, explican qué afirmación falla, qué se esperaba, qué se observó y cómo corregirlo. Incluyen suites de verificación, listas de comprobación, fact-checking, verificación de citas, análisis estructural y revisores adversariales.

Por qué está aquí ahora. Es la otra mitad del marco. Sin sensores, el agente codifica reglas pero nunca sabe si su producción las cumple, y repite errores estructuralmente válidos pero funcionalmente rotos.

Dónde mirar. Böckeler, Maintainability sensors for coding agents · Böckeler, Harness engineering (martinfowler.com)



Un control computacional es determinista, rápido y no miente; uno inferencial es potente para semántica pero caro y probabilístico.

Computacional frente a inferencial

ESTABLECIDO

Cualquier control —guía o sensor— se ejecuta de dos maneras. El computacional lo ejecuta una herramienta determinista (un linter, un test, un script, una búsqueda): rápido, barato, fiable dentro de su alcance. El inferencial lo ejecuta otro modelo (LLM como juez): potente para juicios semánticos, pero lento, caro y probabilístico. La regla pragmática: siempre que algo pueda resolverse con un control computacional, no uses uno inferencial.

Por qué está aquí ahora. Distingue al arnés maduro del inmaduro. El inmaduro descansa todo en controles inferenciales porque parecen sofisticados, y termina costando una fortuna sin entregar fiabilidad.

Dónde mirar. Böckeler, Harness engineering (martinfowler.com) · Böckeler, Maintainability sensors for coding agents

Las tres dimensiones de regulación

EN CONSOLIDACIÓN

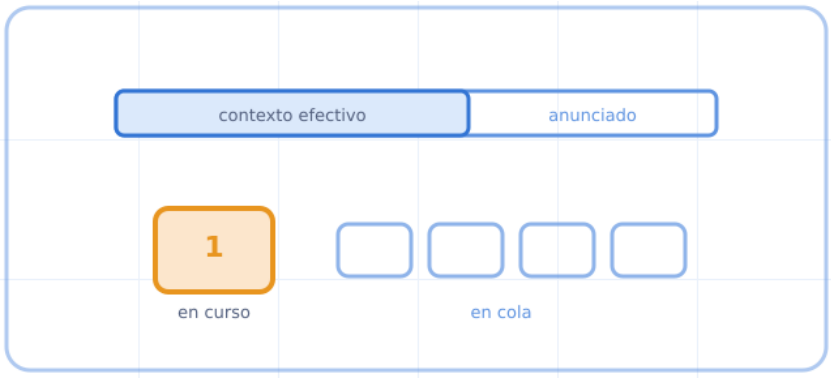
El arnés actúa como un gobernador que regula la producción hacia un estado deseado, en tres dimensiones: calidad estructural (legibilidad, coherencia, mantenibilidad), forma profesional del oficio (rigor metodológico, transparencia sobre fuentes, gestión de incertidumbre, cumplimiento) y comportamiento funcional (¿resuelve realmente el problema?). La primera es la más madura; la tercera es la frontera abierta. Böckeler ha empezado a operacionalizar la dimensión de mantenibilidad con catálogos concretos de sensores.

Por qué está aquí ahora. Ordena dónde poner cada verificación y revela el desequilibrio típico: casi todo el tooling disponible cubre la calidad estructural, mientras la corrección sustantiva queda infraservida.

Dónde mirar. Böckeler, Maintainability sensors for coding agents · Böckeler, Harness engineering (martinfowler.com)

Bloque C — Principios que gobiernan el arnés

Bajo las prácticas concretas hay un puñado de principios que aparecen una y otra vez. Tienen raíz en los límites reales de los modelos y en la cibernética, y explican por qué el arnés se diseña como se diseña.



El contexto efectivo es bastante menor que el anunciado; trabajar con una sola tarea en curso protege ese presupuesto.

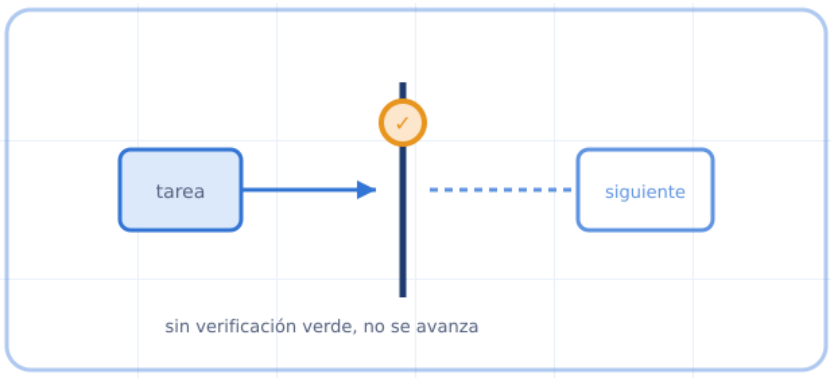
WIP = 1 y el presupuesto de contexto

ESTABLECIDO

Work-in-Progress igual a uno: cada sesión trabaja en una sola tarea o entregable y no avanza hasta que el actual está verificado. El motivo es físico: el contexto utilizable de un modelo es mucho menor que el anunciado —el benchmark RULER de NVIDIA sitúa el contexto efectivo típicamente en torno a la mitad o dos tercios de la ventana— y el efecto Lost in the Middle degrada el procesamiento de información situada en el centro de contextos largos.

Por qué está aquí ahora. Es la regla operativa con mayor impacto empírico en la tasa de éxito de las sesiones. Saturar el contexto con varias tareas a la vez es la causa más común de agentes que dan bandazos.

Dónde mirar. Liu et al., Lost in the Middle (arXiv) · NVIDIA RULER (arXiv)



El pase verificado prohíbe avanzar al siguiente entregable hasta que el actual ha superado un conjunto explícito de verificaciones.

Pass-state gating y la brecha de verificación

ESTABLECIDO

Existe una distancia peligrosa entre el sentimiento de confianza que produce un resultado bien presentado y su corrección real. Los agentes son optimistas estructurales: declaran completo lo que solo es plausible. La regla de oro es que «hecho» no es un adjetivo, sino un resultado de

verificaciones. El pase verificado (pass-state gating) materializa esa regla: el arnés prohíbe avanzar hasta que el entregable actual supera un conjunto explícito de comprobaciones.

Por qué está aquí ahora. Es la defensa directa contra el fallo más caro del trabajo agéntico: aceptar como terminado algo que no lo está y descubrirlo en revisión, o peor, en producción.

Dónde mirar. Anthropic · Effective harnesses for long-running agents · Böckeler, Harness engineering (martinfowler.com)

Cold-start test y legibilidad del entorno

EN CONSOLIDACIÓN

La prueba más reveladora de la calidad de un arnés: abre una sesión nueva, sin historial, y comprueba si el agente, leyendo solo los archivos del proyecto, identifica qué es el sistema, cuál es el progreso y qué falta. Si lo logra en pocos minutos, el arnés es sólido; si no, el conocimiento vive en la cabeza de un humano y no en el proyecto. Su corolario: lo que no está en el proyecto no existe para el agente. El conocimiento tácito debe codificarse en artefactos explícitos.

Por qué está aquí ahora. Convierte una cualidad difusa («buen arnés») en una prueba accionable y repetible, y orienta dónde documentar: no más, sino lo que un recién llegado necesitaría.

Dónde mirar. Lopopolo (OpenAI) · Harness engineering · Anthropic · Effective harnesses for long-running agents

Ley de Ashby y reducción de variedad

EN CONSOLIDACIÓN

La Ley de la Variedad Requisita (Ashby, 1956) dice que un regulador necesita al menos tanta variedad como el sistema que gobierna. Como un modelo puede producir casi cualquier cosa, un arnés que lo iguale sería imposible. La salida práctica es invertir el principio: reducir la variedad del sistema. Atando al agente a una topología concreta —una arquitectura conocida, un stack fijo, una plantilla cerrada, un proceso definido, un corpus acotado— el espacio de lo que puede producir se encoge y un arnés comprehensivo se vuelve viable.

Por qué está aquí ahora. Es la justificación teórica de las plantillas de arnés y de toda la disciplina de acotación. Explica por qué «restringir» no empobrece el resultado, sino que lo hace gobernable.

Dónde mirar. Böckeler, Harness engineering (martinfowler.com) · Ley de la variedad requisita (Ashby)

Bloque D — Preparar el entorno

Aquí la disciplina se vuelve práctica de montaje. Antes de tocar cualquier herramienta concreta hay un cimiento que preparar: el espacio de trabajo, los materiales, las piezas mínimas del arnés y el patrón de memoria entre sesiones.

Con cinco piezas —instrucciones, estado, verificación, alcance y ciclo de vida— ya se está haciendo Harness Engineering.

Pre-flight: espacio, confidencialidad y materiales

ESTABLECIDO

Antes de empezar, tres cosas. Un espacio de trabajo dedicado con versionado, no una conversación suelta. Unas reglas mínimas de confidencialidad: nunca pegar credenciales ni subir datos sensibles a versiones de consumidor, anonimizar cuando se pueda, documentar el nivel de confidencialidad de cada proyecto. Y unos materiales acotados y curados: un corpus de referencia confiable produce trabajo verificable, mientras un agente que «busca lo que encuentre» produce trabajo plausible pero no verificable.

Por qué está aquí ahora. Son los errores más baratos de prevenir y los más caros de arrastrar. La mayoría de los fallos sustantivos del trabajo agéntico se gestan en un entorno mal preparado, no en el modelo.

Dónde mirar. Lopopolo (OpenAI) · Harness engineering · Anthropic · Effective context engineering

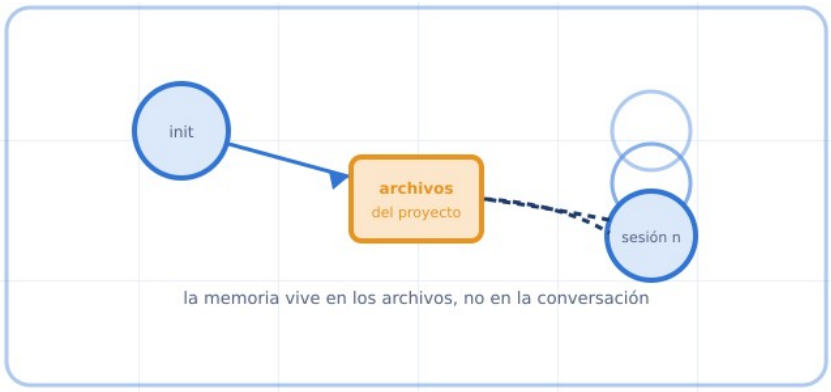
El arnés mínimo viable y AGENTS.md

ESTABLECIDO

Un arnés externo mínimo viable tiene cinco piezas: instrucciones (la guía maestra que lee el agente al empezar), estado (memoria persistente entre sesiones: lista de entregables, registro de decisiones y de progreso, versionado), verificación (los sensores), alcance (reglas que restringen qué hace en cada sesión) y ciclo de vida (bootstrap, persistencia, limpieza). El archivo de instrucciones se ha estandarizado en buena parte del ecosistema como AGENTS.md, un formato abierto mantenido bajo la Linux Foundation.

Por qué está aquí ahora. Da un punto de partida concreto y completo. Si están las cinco piezas, ya se hace Harness Engineering; lo demás —hooks, skills, subagentes— amplifica, pero no sustituye este cimiento.

Dónde mirar. AGENTS.md · estándar abierto · Lopopolo (OpenAI) · Harness engineering



Un agente inicializa el entorno una vez; otro avanza sesión tras sesión leyendo y actualizando los archivos del proyecto.

Initializer + working agent: memoria en archivos

EN CONSOLIDACIÓN

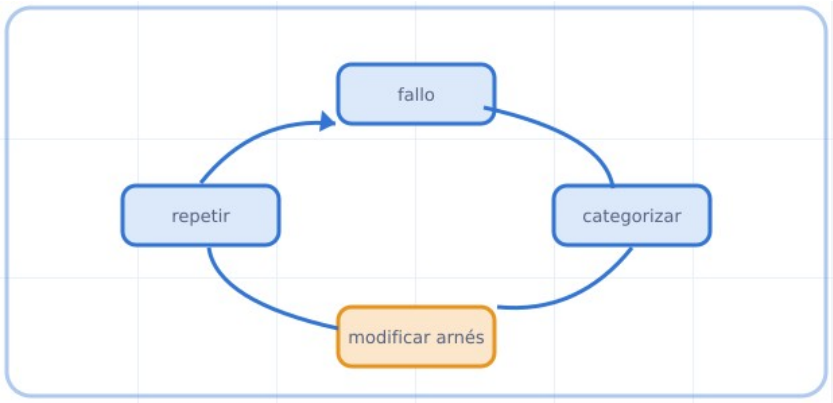
El patrón canónico de Anthropic para tareas largas separa dos roles. Un initializer se ejecuta una sola vez: expande el brief en una lista estructurada de entregables con criterios de aceptación, monta el espacio de trabajo y prepara las verificaciones. Un working agent se despierta sesión tras sesión: se refundamenta leyendo los archivos, lee el registro de progreso, elige una entrada pendiente, la produce y verifica, actualiza el progreso y cierra. La clave: la memoria persistente vive en los archivos del proyecto, no en la conversación.

Por qué está aquí ahora. Resuelve el problema central de las tareas que no caben en una sola ventana de contexto: cómo no perder lo decidido entre sesiones. Anthropic ha publicado además patrones y un repositorio de referencia para implementarlo.

Dónde mirar. Anthropic · Effective harnesses for long-running agents · Anthropic · cwc-long-running-agents (GitHub)

Bloque E — Patrones de frontera

Donde la disciplina aún se está escribiendo. El bucle de mejora del arnés es ya práctica habitual; la verificación sustantiva y las arquitecturas multiagente son terreno en movimiento, con alta relevancia y alta volatilidad.



Cuando algo falla, se categoriza el error y se modifica el arnés; regañar al modelo no cambia su comportamiento.

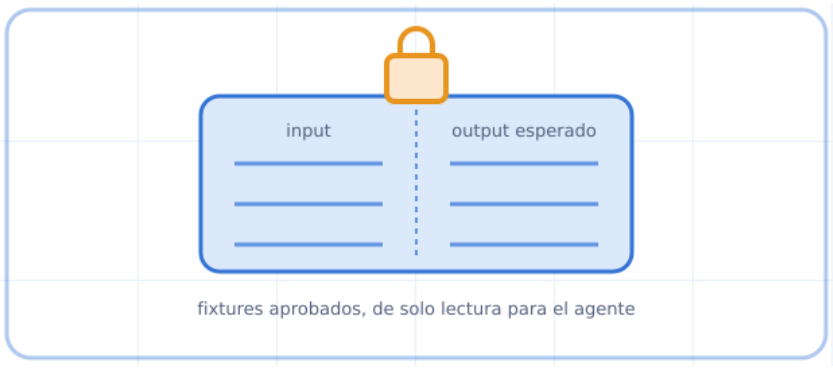
El bucle de dirección y el journal del arnés

EN CONSOLIDACIÓN

El steering loop es la práctica humana de iterar el arnés cuando algo falla, en lugar de iterar el prompt o regañar al modelo. El procedimiento: no reñir, categorizar el error (¿falta una guía? ¿un sensor? ¿acotar el alcance? ¿mejorar la legibilidad?), modificar el arnés en consecuencia, repetir la tarea y comprobar. Funciona porque el modelo no aprende de los regaños —no actualiza sus pesos —, pero sí cambia de comportamiento cuando cambian sus instrucciones, sus señales de error y sus materiales. Un journal del arnés registra cada cambio y su resultado.

Por qué está aquí ahora. Es el mecanismo que convierte el uso de agentes en una disciplina acumulativa en lugar de una sucesión de frustraciones. Cada fallo bien aprovechado mejora el sistema de forma permanente.

Dónde mirar. Böckeler, Harness engineering (martinfowler.com) · Anthropic · Scaling Managed Agents



Pares (input, output esperado) marcados como de solo lectura: el agente debe satisfacerlos, no puede reescribirlos para «pasar».

Behaviour harness, approved fixtures y revisores adversariales

EMERGENTE

Verificar que un entregable realmente hace lo que debía —no solo que está bien formado— es el frente abierto del campo. Las verificaciones que genera el propio agente tienden a ser circulares: validan el malentendido si lo hubo. Los patrones prometedores incluyen los approved fixtures (pares input/output o pregunta/respuesta mínima que el humano fija como read-only y el agente no puede modificar), el property-based testing, los revisores adversariales (un agente cuyo único trabajo es objetar) y el LLM como juez en contexto fresco, siempre como segunda línea tras los controles computacionales.

Por qué está aquí ahora. Es el problema sin resolver de la disciplina y, a la vez, el de mayor consecuencia: cuanto más crítica es la corrección sustantiva, menos autonomía debe darse al agente y más rol humano debe haber. Los patrones se redefinen casi cada mes.

Dónde mirar. Anthropic · Harness Design for Long-Running App Development · Böckeler, Maintainability sensors for coding agents

Arquitecturas multiagente y limpieza agéntica

EMERGENTE

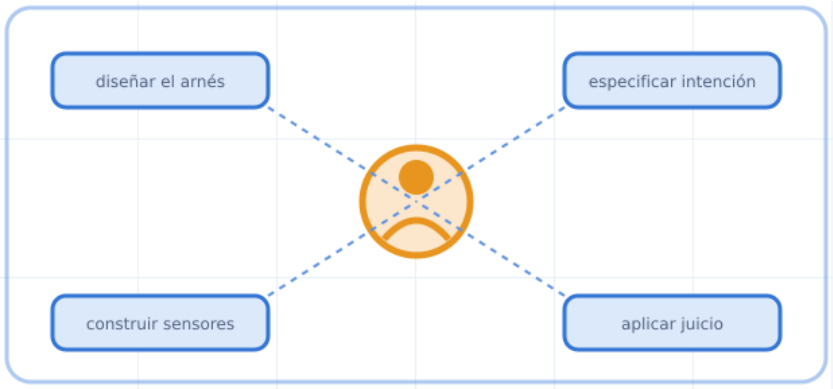
La frontera de la orquestación. Anthropic ha documentado arquitecturas de tres roles (planner, generator, evaluator), inspiradas en redes generativas adversariales, donde un evaluador con contexto fresco y sin permisos de escritura juzga el trabajo del generador en ciclos sucesivos; y la separación Brain/Hands/Session, que desacopla el modelo y su bucle de los sandboxes efímeros y del registro de la sesión. En paralelo, la garbage collection agéntica usa agentes en segundo plano para limpiar la deriva del proyecto: código muerto, fuentes obsoletas, dependencias huérfanas, inconsistencias terminológicas.

Por qué está aquí ahora. Marca hacia dónde evoluciona el arnés en tareas largas y autónomas. Es potente pero volátil: las arquitecturas concretas y las herramientas cambian deprisa, y conviene seguirlas sin adoptarlas por moda.

Dónde mirar. Anthropic · Harness Design for Long-Running App Development · Anthropic · Scaling Managed Agents

Bloque F — Criterio profesional y rol humano

El arnés bien diseñado no elimina al humano: lo concentra donde más vale. Este bloque reúne lo que no se delega —medir, reconocer antipatrones, decidir cuándo no usar un agente y sostener el juicio profesional sobre el entregable.



El humano deja de escribir cada línea para diseñar el entorno, especificar la intención, construir los bucles de feedback y aplicar juicio donde importa.

Medir el arnés, evitar antipatrones y sostener el criterio

EN CONSOLIDACIÓN

Un arnés se gobierna con indicadores: rebuild cost (tiempo de una sesión vacía a trabajo útil), tasa de cierre a la primera, objeciones del destinatario, cobertura de sensores, latencia de mejora del arnés. Frente a ellos, los antipatrones recurrentes: instrucciones sobre-especificadas que el modelo deja de leer, exploración infinita, confianza prematura, bypass por desesperación, proliferación de subagentes y fe ciega en el revisor inferencial. Y el límite del propio método: hay tareas donde lo correcto es no usar un agente —cuando el contexto solo está en tu cabeza, cuando la responsabilidad es intransferible, cuando no sabrías evaluar la respuesta, cuando la confidencialidad es máxima—. El humano diseña el arnés, especifica la intención, construye los bucles de feedback y aplica juicio donde la máquina no debe decidir sola.

Por qué está aquí ahora. Cierra el mapa con la capa que protege la validez de todo lo anterior: sin medición no hay mejora, sin reconocer antipatrones se repiten, y sin criterio humano un arnés impecable puede firmar trabajo que nadie ha pensado de verdad.

Dónde mirar. Böckeler, Harness engineering (martinfowler.com) · Mollick, One Useful Thing

Qué vigilar en la próxima revisión

Señales de cambio que el equipo editorial anticipa para la siguiente edición de este mapa:

- Consolidación de la verificación sustantiva: si los approved fixtures, el property-based testing o los revisores adversariales maduran hasta convertirse en práctica estándar, el behaviour harness pasaría de EMERGENTE a EN CONSOLIDACIÓN.
- Plantillas de arnés reutilizables: la aparición de bundles públicos por topología (informe estratégico, memorando legal, microservicio, paper académico) podría justificar un puntero propio si el movimiento se afianza.
- Estabilización de las arquitecturas multiagente: planner/generator/evaluator y la separación Brain/Hands/Session aún se redefinen; conviene revisar si alguna se asienta como referencia.
- Evolución del estándar de instrucciones: la adopción de AGENTS.md y su gobernanza bajo la Linux Foundation pueden cambiar las recomendaciones prácticas sobre archivos de instrucciones.
- Obsolescencia por mejora del modelo: cada componente del arnés codifica una asunción sobre lo que el modelo no puede hacer solo; al mejorar los modelos, algunas piezas se vuelven peso muerto y deben retirarse.

Nota sobre las referencias

Los enlaces incluidos estaban disponibles y verificados en la fecha de actualización. Dado el ritmo de cambio del área, algunos pueden modificarse; cada revisión del mapa actualiza también sus referencias.