Universidad del Desarrollo
Facultad de Ingeniería

dataScience UDD

CRISSLAB
DECODING BEHAVIOR, ADVANCING SOCIETY

NICO NORTHWESTERN INSTITUTE ON COMPLEX SYSTEMS

TRABAJO CON AGENTES

Cuando «listo» necesita pruebas

Una arquitectura para revisar, continuar y defender el trabajo

Cristian Candia, Ph.D.

Universidad del Desarrollo (UDD), Chile
Director at the Computational Research in Social Sciences Lab
Associate Professor, Data Science Institute, School of Engineering

Northwestern University, United States.
External Faculty, Northwestern Institute on Complex Systems (NICO)
Kellogg School of Management.

Capybara Spa (AI & Network Science for Preventive, Traceable School Coexistence Compliance)
Founder & Chief Scientific and Technological Officer (CSTO)

Video Credit: Mauro Martino

El agente termina la tarea y responde:

LISTO.

¿LISTO PARA QUÉ?

¿El archivo abre?

¿Las pruebas cubren lo importante?

¿Podemos reconstruir cómo se llegó al resultado?

«Listo» alcanza para seguir trabajando.

Para confiar, necesitamos ver qué se hizo y qué se comprobó.

ACCESO NO ES AUTORIDAD

ACCESO

La respuesta está disponible

AUTORIDAD

La respuesta tiene respaldo suficiente para orientar una creencia, un análisis o una acción.

El riesgo aparece cuando una respuesta orienta una decisión antes de que revisemos la evidencia que la respalda

Candia (2026), When Answers Are Everywhere: The New Scarcity in the Age of AI. Libro el proceso.

EL ACCESO CAMBIA LA CONDUCTA

CUANDO PODÍAN CONSULTAR A LA IA, CASI NADIE DIJO «NO SÉ»

Los participantes respondían seis preguntas difíciles sobre detalles visuales en películas y podían decir «no sé». En algunas condiciones experimentales consultaban una IA cuyo consejo era habitualmente incorrecto.

MENOS PERSONAS DIJERON «NO SÉ»

SIN IA → CON ACCESO A IA

SIN INCENTIVOS - ESTUDIO 1a

36% → 6%

CON INCENTIVOS - PROMEDIO ESTUDIOS 2–4

33.7% → 5.7%

MENOS ACIERTO · MÁS CONFIANZA

SIN IA → CON ACCESO A IA

PERFORMANCE AGREGADO

27,5% → 9,2% SIN INCENTIVOS MONETARIOS

29% → 16% CON INCENTIVOS MONETARIOS

CONFIANZA MEDIA - ESTUDIO 2 - ESCALA 0–100

29,6 → 75,9

Con acceso a IA, las personas dijeron mucho menos « no sé », acertaron menos y, al mismo tiempo, se sintieron mucho más seguras. La respuesta no solo estuvo disponible: recibió autoridad antes de haber demostrado que la merecía.

Marcoccia, Quattrociocchi & Capraro (2026). *AI advice suppresses people's willingness to say "I don't know"…* arXiv:2607.13562v1, 15 July 2026.

CUATRO PRUEBAS ANTES DE ACEPTAR «LISTO»

¿Qué tendría que existir para aceptar que un trabajo está realmente listo?

CORRECTO	El resultado cumple lo que la tarea pedía, según los criterios que pudimos verificar.
REPRODUCIBLE	Otra persona puede repetir el proceso y obtener resultados consistentes.
TRAZABLE	Podemos reconstruir las entradas, herramientas, versiones y decisiones.
CIERRE DEFENDIBLE	Cumple condiciones explícitas de aceptación y deja claros sus límites y pendientes.

La arquitectura no garantiza la verdad del resultado: reduce el riesgo de aceptar como «listo» un trabajo sin respaldo suficiente

DÓNDE APARECE EL VALOR

Una arquitectura puede aportar valor aunque el modelo no acierte más

RENDIMIENTO

¿Resuelve más tareas o produce mejores resultados?

EFICIENCIA

¿Reduce tiempo, costo y retrabajo?

CONFIABILIDAD

¿Produce resultados consistentes y evita que los errores se propaguen?

AUDITABILIDAD

¿Deja evidencia para revisar, continuar y responder por el trabajo?

En investigación y análisis, reducir retrabajo y dejar evidencia puede valer tanto como mejorar el rendimiento.

CAPACIDADES DESIGUALES ENTRE TAREAS

La IA mejora algunas tareas y perjudica otras dentro del mismo trabajo

Experimento aleatorizado con 758 consultores de BCG, asignados a trabajar sin IA, con GPT-4 o con GPT-4 más una introducción al uso de prompts.

18 TAREAS QUE GPT-4 PODÍA APOYAR

+12,2% **más tareas completadas**

25,1% más rápido, con mejor calidad evaluada por expertos

1 TAREA FUERA DE LAS CAPACIDADES DE GPT-4

RECOMENDACIÓN CORRECTA

84,5% → **≈65,5%**

SIN IA — CON IA

Caída aproximada: 19 puntos porcentuales

GPT-4: 70,6% · GPT-4 + introducción: 60,0%*

La unidad de adopción no debería ser el cargo completo ni el proyecto completo. Debe ser cada tarea, junto con una forma externa de comprobar si salió bien. En este caso, usar GPT-4 con mayor fluidez pudo hacer más convincente y más rápida una conclusión equivocada.

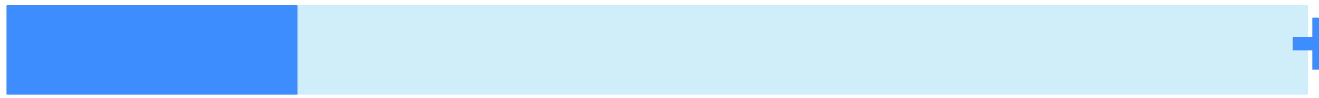
Fuente: Dell'Acqua et al. (2026), Organization Science, doi:10.1287/orsc.2025.21838. Experimento preregistrado; la evidencia fuera de la frontera proviene de una tarea.

SKILLS EN EVALUACIÓN

Las skills también ayudan de forma desigual según la tarea

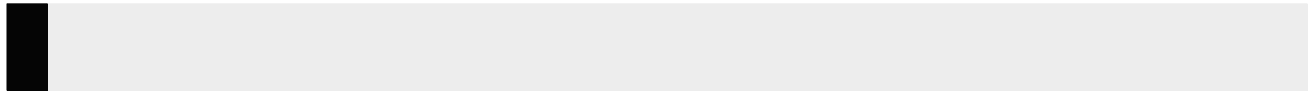
Cambio promedio en tareas resueltas (pass rate) al añadir skills curadas

SkillsBench
86 tareas · 11 dominios



+16,2 pp

SWE-Skills-Bench
Tareas de ingeniería de software



+1,2 pp

Los resultados no son directamente comparables, pero muestran el mismo principio: una skill no tiene valor por sí sola. Depende de su ajuste con la tarea y con la forma de verificarla.

Li et al. (2026), SkillsBench, arXiv:2602.12670 · Han et al. (2026), SWE-Skills-Bench, arXiv:2603.15401.

QUÉ DETERMINA LA UTILIDAD

Una skill ayuda solo cuando encaja con la tarea y el entorno

2–3

módulos pertinentes

rindieron mejor que cargar toda la documentación

16/84

tareas empeoraron

al añadir una skill curada

39/49

skills no mejoraron

el pass rate promedio en ingeniería de software

El costo del mal ajuste

Una skill incompatible con la versión del proyecto redujo el pass rate hasta 10 puntos porcentuales. Más contexto también aumentó el costo sin mejorar el éxito. La utilidad depende de entregar el procedimiento correcto, para la versión correcta y en la tarea correcta.

Li et al., SkillsBench, arXiv:2602.12670 · Han et al., SWE-Skills-Bench, arXiv:2603.15401.

BIBLIOTECAS REFINADAS

Una biblioteca mejora cuando aprende de sus fallos

SPREADSHEETBENCH · CONSTRUIR UNA BIBLIOTECA

SIN SKILLS	BIBLIOTECA CONSTRUIDA POR SKILLAXE · 22 SKILLS
16,0%	52,0%

La biblioteca redujo fallos de ejecución.

Otra biblioteca autogenerada también alcanzó 52%, pero necesitó 69 skills. Aquí el aporte fue una biblioteca más compacta y mejor dirigida.

SKILLSBENCH · REFINAR SKILLS GENERADAS POR IA

SIN REFINAR	DESPUÉS DE SKILLAXE
32,5%	41,6%

+9,1 puntos porcentuales de pass rate. El refinamiento permitió completar correctamente más tareas.

La mejora vino sobre todo de completar más tareas, no de acertar más en las que ya completaba.

Mejorar una skill exige observar su ejecución, corregir el procedimiento y volver a probarlo fuera de las tareas usadas para ajustarlo

Fuente: Gautam, Radhakrishna & Gulwani (2026), SkillAxe, arXiv:2606.10546v2. Pass rate evaluado con verificadores propios de cada benchmark.

Una skill convierte experiencia en un procedimiento reutilizable

El sistema extraía o construía una **skill/workflow** a partir de ejecuciones anteriores

AWM (Agent Workflow Memory) · MIND2WEB

Evalúa si el agente reconoce acciones correctas en trayectorias web previamente registradas

36,2% → 45,1%

Agente base → Agente + AWM

AWM · WEBARENA

Evalúa si puede completar autónomamente una tarea dentro de un sitio funcional

23,5% → 35,5%

Agente base → Agente + AWM

AFTER · TRANSFERENCIA ENTRE MODELOS

¿El procedimiento sigue funcionando cuando cambia el modelo que lo ejecuta?

SKILL APRENDIDA DESDE TRAZAS DE UN SOLO MODELO

59,4%

SKILL APRENDIDA DESDE TRAZAS DE VARIOS MODELOS

73,1%

de exactitud bajo cambio de modelo *de skill aprendida desde un solo modelo* vs *skill aprendida desde varios modelos*

Una skill es memoria procedural cuando el procedimiento se reutiliza en tareas nuevas y sigue funcionando cuando cambia el ejecutor

Wang et al., Agent Workflow Memory 2024, arXiv:2409.07429 · Belikova et al., AFTER 2026, arXiv:2606.23127.

Lo anterior deja una tesis sencilla: el agente solo no basta.

DISEÑO

Conservar · conectar · verificar · bloquear

Lo anterior no pide simplemente “más IA”, sino una forma de trabajo que permita verificar antes de cerrar

1	**ACCESO NO ES AUTORIDAD**	Una respuesta disponible puede llegar a la decisión antes que la evidencia.
2	**LA FRONTERA ES POR TAREA**	La misma IA puede ayudar en una parte del trabajo y empeorar otra.
3	**LAS SKILLS NO SON MAGIA**	Sirven cuando encajan con la tarea, el entorno y la forma de verificación.
4	**MEMORIA PROCEDURAL**	Guardar procedimientos vale si permiten continuar, revisar y transferir el trabajo.

La arquitectura no garantiza el resultado. Hace visible qué se hizo, qué se comprobó y qué queda pendiente

CINCO PIEZAS

Una arquitectura confiable separa cinco funciones distintas

SKILL	Procedimiento reutilizable que reúne instrucciones, referencias, scripts y archivos.
AGENTE	Ejecutor que interpreta un objetivo, elige acciones, usa herramientas y ajusta el plan dentro de un rol delimitado.
HANDOFF	Artefacto estructurado que transfiere hechos, decisiones, evidencia, bloqueos y próximas acciones entre fases o ejecutores.
HARNESS	Infraestructura que registra el estado, las trazas, los artefactos y las validaciones para reconstruir una ejecución.
GATE	Control que impide avanzar o cerrar cuando falta evidencia o una condición de aceptación acordada.

El agente ejecuta. La arquitectura conserva, conecta y controla el trabajo

Referencias de definición: OpenAI Developers, Build skills & Customization (consultado 17-07-2026) · Anthropic (2024), Building effective agents.

CUÁNTA ESTRUCTURA Y AUTONOMÍA USAR

Empieza con la forma de ejecución más simple que permita hacer y verificar el trabajo

Separar cinco funciones no significa desplegarlas todas: añade estructura y autonomía solo cuando la tarea lo exige.

1 LLAMADA DIRECTA — Una respuesta o cambio puntual

2 EJECUCIÓN CON SKILL — Un procedimiento reutilizable guía el trabajo

HARNESS

3 WORKFLOW TRAZADO — Fases, estado, handoffs, validación y gates

4 AGENTE — El ejecutor decide la ruta dentro de un rol delimitado

5 MULTIAGENTE — Roles separados se delegan a distintos ejecutores

Añade complejidad solo cuando mejore la ejecución, la verificación o la separación del juicio

Fuente: Anthropic (2024), Building effective agents — workflows previsibles, agentes flexibles; aumentar complejidad solo cuando el resultado lo justifique.

FLUJO TRAZABLE

Cuando la tarea necesita un workflow trazado, cada fase deja un estado verificable

HARNESS · manifiesto · estados de fase · trazas · handoffs · artefactos · validaciones

El workflow no es solo una secuencia: conserva transiciones y puede rechazar el cierre

¿Dónde queda todo eso cuando termina el turno o se pierde el contexto de la conversación?

HARNESS

El harness convierte la ejecución en una corrida persistente y verificable

En esta arquitectura vive como scripts Python y una carpeta local que registra estados, trazas, handoffs, artefactos y validaciones. La skill workflow-state-handoff-policy le indica a codex cuándo una tarea debe usar un workflow trazado y contiene los comandos que debe ejecutar.

1 · CÓDIGO

Scripts Python

crean, actualizan, consumen y validan la corrida

scripts/init_workflow_run.py
scripts/update_workflow_state.py
scripts/create_workflow_handoff.py
scripts/validate_workflow_run.py
scripts/workflow_status.py

2 · CORRIDA

Carpeta local

donde vive la evidencia de una ejecución concreta

workflow_runs/<run_id>/
  run_manifest.json
  trace.jsonl
  state/*.json
  handoffs/*.json
  outputs/
  validation/

3 · CONTROL

Reglas de cierre

bloquean falsos “listo” cuando falta evidencia

- fases completadas o justificadas
- handoffs consumidos o cancelados
- outputs y validaciones existentes
- evento terminal coherente

El agente ejecuta; el harness deja evidencia verificable de cómo ocurrió y hace que ese avance quede persistente, reconstruible y verificable fuera del chat

El harness conserva la corrida completa. Pero una fase no necesita recibir toda la carpeta para continuar. Necesita un subconjunto más acotado: qué sabemos, qué se decidió, qué artefactos existen, qué está bloqueado y qué debe hacerse ahora

Fuente: WORKFLOW_RUN_HARNESS.md; scripts del repositorio Codex_Skills. Snapshot 16–17/07/2026.

El handoff transfiere el estado accionable entre fases

La regla vive en una skill; scripts Python crean y validan el archivo; y cada corrida conserva el handoff como un JSON que la fase siguiente debe consumir.

1 PROCEDIMIENTO

Skill / Markdown

```
collections/.../workflow-handoff-
briefing/SKILL.md
```

- Cuándo usar handoff
- Qué contenido debe preservar
- Cuándo bloquear una transferencia

2 HERRAMIENTAS

Scripts + esquema

```
scripts/create_workflow_handoff.py
scripts/consume_workflow_handoff.py
schemas/workflow_handoff.schema.json
```

- Crean el artefacto JSON
- Marcan ready / consumed / cancelled
- Validan campos y referencias

3 EVIDENCIA DE CORRIDA

Archivo JSON

```
workflow_runs/<run_id>/handoffs/
h0001_explore_to_execute.json
```

- Hechos, decisiones y restricciones
- Artefactos y validación
- Bloqueos y próximas acciones

El harness conserva la corrida. El handoff transfiere hechos, decisiones, evidencia, bloqueos y próximas acciones

Una vez que el estado puede transferirse de esta manera, podemos usar agentes distintos para ejecutar roles separados sin perder continuidad ni evidencia.

ARQUITECTURA ACTUAL

Una fuente canónica conserva los procedimientos. Cada corrida conserva su evidencia

36	11	29 / 7	6	5
skills canónicas	colecciones	globales / de proyecto	agentes opcionales	esquemas de estado y evidencia

MEMORIA ENTRE CORRIDAS · PROCEDIMIENTOS

CANON
collections/*/skills/*
contenido editable de cada skill

→

REGISTRO DE ALCANCE (MANIFEST) + INSTALADOR
qué skills se exponen · dónde · validación

→

PUNTOS DE DESCUBRIMIENTO
usuario · repositorio · proyecto
symlinks hacia el canon

→

CODEX
descubre la skill sin copiarla

MEMORIA DENTRO DE UNA CORRIDA · ESTADO Y EVIDENCIA: python3 scripts/init_workflow_run.py

WORKFLOW RUN · `workflow_runs/<run_id>/`

manifiesto · estados de fase · handoffs · trazas · artefactos · validación

Una fuente por procedimiento. Una carpeta verificable por ejecución

Fuente: Arquitectura local, snapshot auditado del repositorio Codex_Skills, 17-07-2026.

SKILL O AGENTE

La estabilidad de la secuencia de trabajo decide cuánto debe decidir el agente

No son piezas excluyentes: la skill conserva el procedimiento, el agente decide cómo recorrerlo

BASTA UNA SKILL

cuando los pasos son estables, reutilizables y tienen una definición clara de “bien hecho”

- lista de control reproducible
- comandos o scripts conocidos
- formato y criterios fijos

NECESITAS UN AGENTE

cuando los siguientes pasos deben decidirse a partir de lo que ocurre durante la ejecución

- selecciona y aplica skills según el contexto
- exploración abierta
- respuesta del entorno
- recuperación ante errores

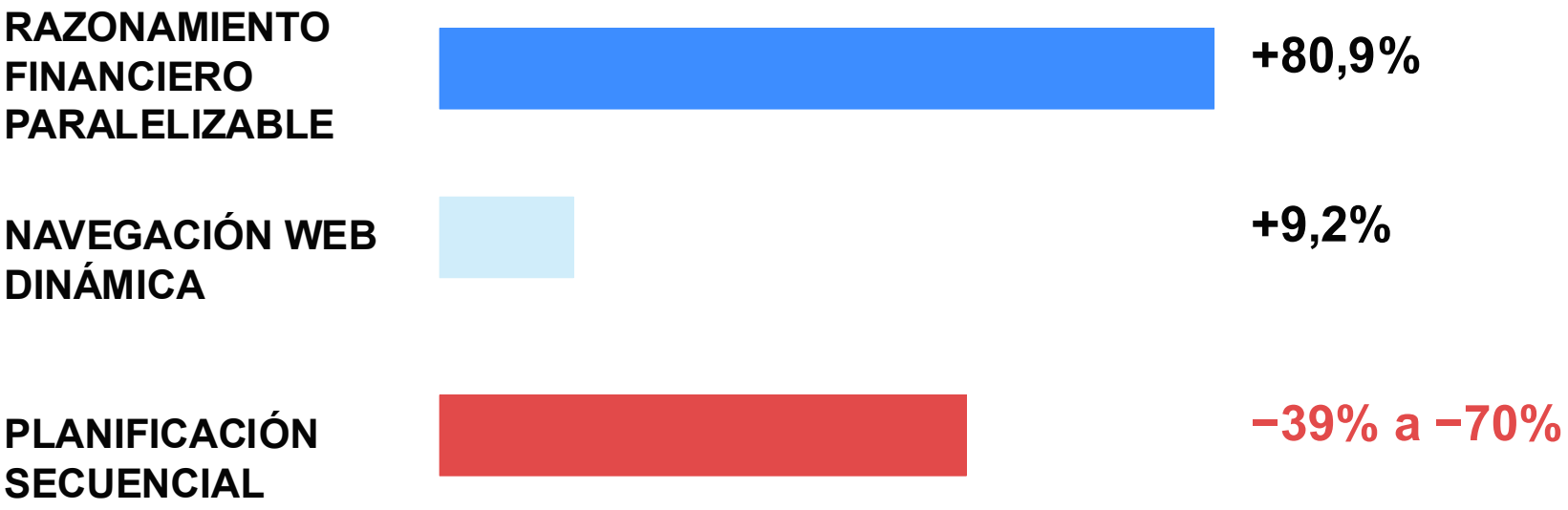
Si los pasos pueden definirse por adelantado, la autonomía adicional debe justificar su costo

Anthropic (2024), Building effective agents · OpenAI Developers, Build skills & Subagents (17-07-2026).

MULTIAGENTES

Los multiagentes ayudan cuando el trabajo puede dividirse de verdad

Cambio relativo frente a un agente único, según la estructura de la tarea y la forma de coordinación



PROPAGACIÓN DEL ERROR · AGENTE ÚNICO = 1

17,2×

agentes independientes sin coordinación central

4,4×

coordinación central, el orquestador contiene la propagación

Más agentes no implican más evidencia independiente.

Otros estudios muestran que muchos fallos multiagente no vienen del modelo aislado, sino del diseño, la coordinación y la verificación del sistema.

Fuentes: Kim et al. (2025/26), arXiv:2512.08296 · Cemri et al. (2025), Why Do Multi-Agent LLM Systems Fail?, arXiv:2503.13657.

Escalar agentes no basta. También debe escalar la capacidad de revisarlos y responder por sus resultados

La ciencia con agentes necesita instituciones que crezcan con ellos

Si aumenta la capacidad de producir resultados científicos, debe crecer también la capacidad de verificarlos, autorizarlos y responder por ellos.

LO QUE LOS AGENTES PUEDEN ESCALAR

- búsqueda y síntesis de literatura
- generación y ejecución de código
- análisis de datos
- propuesta de hipótesis
- crítica y exploración de modelos

LO QUE DEBE ESCALAR AL MISMO TIEMPO

- verificación independiente
- trazabilidad del origen y de cada paso
- responsabilidad humana asignada
- resultados interpretables
- autorización para acciones de alto impacto
- diversidad metodológica

Escalar la producción sin escalar la capacidad de revisarla, autorizarla y responder por ella solo acelera errores y falsos cierres

Jimenez et al. (2026), AI Scientists as Engines of Discovery: A Case for Development within Reformed Institutions. arXiv:2606.22859, v1.

El agente puede encontrar una solución que funciona sin entender por qué funciona

CausalGame convierte el descubrimiento causal en un juego de diseño de drones. Cada agente puede probar 200 diseños, entrega una única solución final (evaluada en 1.000 drones nuevos) y debe explicar el mecanismo que cree haber descubierto.

CONTEXTO DEL EXPERIMENTO

30
modelos de frontera evaluados individualmente: GPT, Claude, Gemini, DeepSeek, etc.

14
escenarios con un mecanismo causal oculto

TRAMPAS OBSERVACIONALES

- sesgo de selección
- Variables de confusión ocultas
- error de medición

RESULTADO CENTRAL

68,0%
mejor promedio de supervivencia final (claude opus 4.5)

78–85%
óptimos analíticos

5–7%
de las sesiones recibió crédito por razonamiento causal

Un resultado exitoso no demuestra comprensión causal. Sin ella, puede no generalizar y conducir a la intervención equivocada

Chen et al. (2026), CausalGame: Benchmarking Causal Thinking of LLM Agents in Games. arXiv:2607.04293v1; short version accepted as ICML 2026 Oral.

Cada resultado debe poder reconstruirse desde sus datos y su código

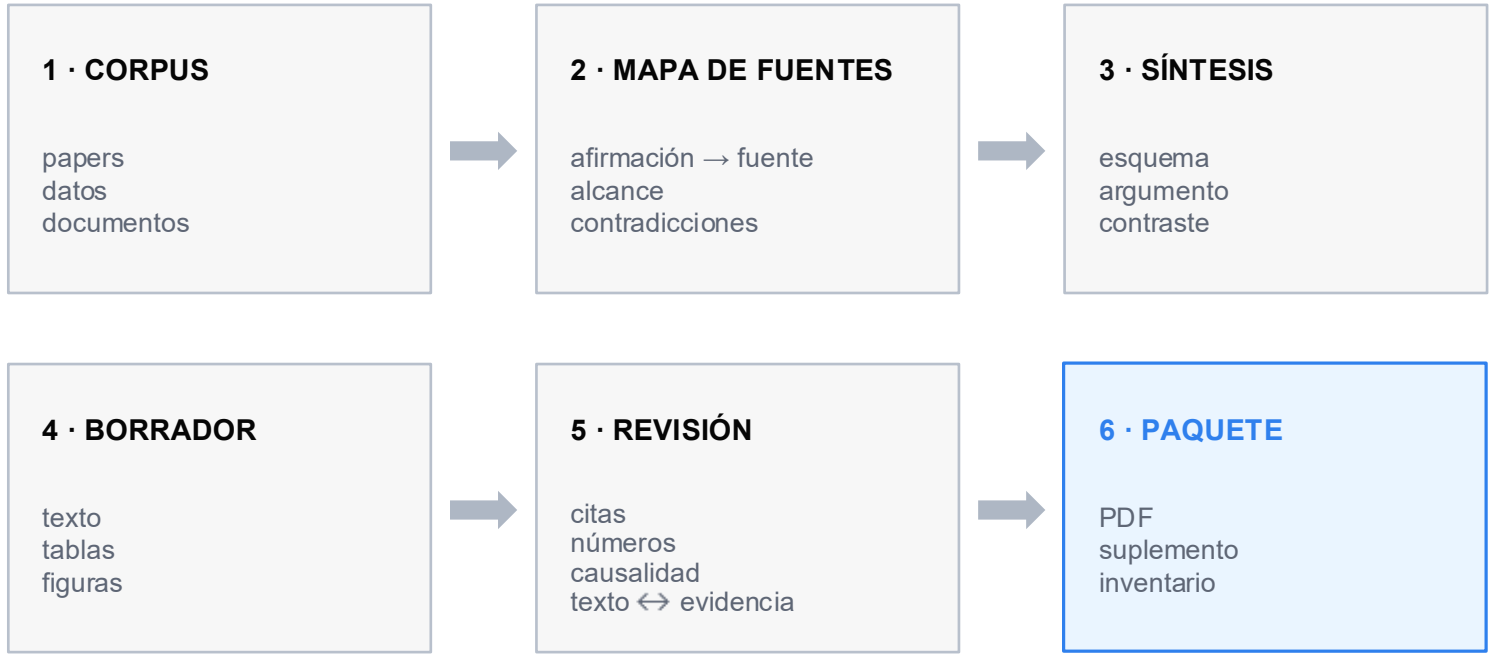
La unidad verificable no es solo la tabla o la figura: es la cadena que conecta origen, transformación, análisis y revisión.

Contrato de evidencia

Cada entrega debe indicar qué datos entraron, qué código los transformó, qué resultado salió y qué validación permite confiar o corregir.

Cada afirmación debe conservar el camino hasta la evidencia que la respalda

En escritura científica, la unidad trazable no es solo el archivo final: es cada afirmación relevante y su soporte.



Contrato común

En ambos flujos, la arquitectura conserva el origen, la ejecución, la revisión y la evidencia de cierre.

Poder reconstruir el camino no demuestra que el resultado sea correcto

EVALUACIÓN

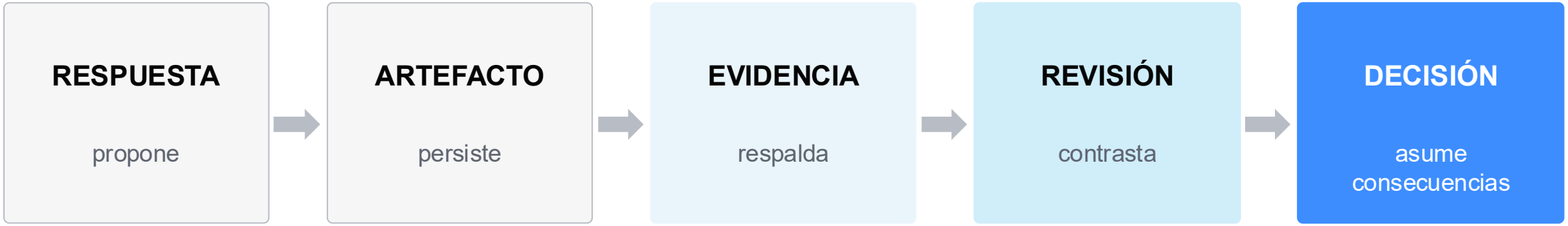
La evaluación debe verificar el resultado, no premiar una explicación convincente

01	**CRITERIO EXTERNO**	tests, estado final o condiciones de aceptación independientes del agente.
02	**VARIAS CORRIDAS**	debe funcionar de manera consistente, no solo acertar una vez.
03	**TRAZABILIDAD**	entradas, herramientas, versiones, resultados y evidencia vinculados

Una explicación convincente no compensa un estado final incorrecto.

Fuentes: Yao et al. (2024), tau-bench, arXiv:2406.12045 · Jimenez et al. (2024), SWE-bench, ICLR 2024 · Han et al. (2026), arXiv:2603.15401.

La arquitectura crea distancia entre una respuesta y una decisión



ATAJO: AUTORIDAD PREMATURA

Aceptación antes de reunir y contrastar el respaldo necesario.

La arquitectura no reemplaza el juicio: evita que una respuesta, aunque suene convincente, reciba autoridad sin evidencia suficiente

Cuando las respuestas abundan

el juicio se vuelve poder.

El juicio es la capacidad disciplinada de conceder o negar autoridad a una respuesta.

Esta arquitectura no automatiza el juicio. Crea mejores condiciones para ejercerlo.

EVIDENCIA Y BENCHMARKS

1. Dell'Acqua et al. (2026). Navigating the Jagged Technological Frontier. Organization Science

2. Marcoccia, Quattrociocchi & Capraro (2026). AI advice suppresses people's willingness to say 'I don't know'… arXiv:2607.13562

3. Li et al. (2026). SkillsBench. arXiv:2602.12670

4. Han et al. (2026). SWE-Skills-Bench. arXiv:2603.15401

5. Zhong et al. (2026). SkillLearnBench. arXiv:2604.20087

6. Gautam et al. (2026). SkillAxe. arXiv:2606.10546

7. Wang et al. (2024). Agent Workflow Memory. arXiv:2409.07429

8. Belikova et al. (2026). Managing Procedural Memory. arXiv:2606.23127

9. Cemri et al. (2025). Why Do Multi-Agent LLM Systems Fail? arXiv:2503.13657

10. Kim et al. (2025/26). Towards a Science of Scaling Agent Systems. arXiv:2512.08296

11. Yao et al. (2024). tau-bench. arXiv:2406.12045

12. Jimenez et al. (2024). SWE-bench. ICLR 2024 / arXiv:2310.06770

13. Jimenez et al. (2026), AI Scientists as Engines of Discovery: A Case for Development within Reformed Institutions. arXiv:2606.22859, v1.

ARQUITECTURA Y MARCO CONCEPTUAL

14. Anthropic (2024). Building effective agents

15. OpenAI Developers. Build skills · Customization · Subagents

16. Jimenez, R., Bolliet, B., Villaescusa-Navarro, F., et al. (2026). AI Scientists as Engines of Discovery: A Case for Development within Reformed Institutions. arXiv:2606.22859

17. Candia, C. (2026). When Answers Are Everywhere: The New Scarcity in the Age of AI. Libro no publicado.

BACKUP

Un caso de evaluación: síntesis científica

DISEÑO PROPUESTO

¿El flujo de evidencia mejora solo la respuesta final o también qué evidencia se recupera, contrasta y hace visible?

1 · RESPUESTA DIRECTA

modelo → respuesta

2 · RETRIEVAL / RAG CON FUENTES

recuperación → respuesta

3 · FLUJO DE EVIDENCIA

mapa afirmación–fuente → contraste → revisión → gate

MÉTRICAS

CALIDAD DE LA RESPUESTA
exactitud · cobertura · estabilidad entre corridas
CALIDAD DE LA EVIDENCIA
soporte afirmación–fuente · evidencia primaria · líneas independientes · resultados contrarios · correcciones y retractaciones
COSTO OPERACIONAL
tiempo · tokens · costo · intervención humana

La prueba es si mejora la evidencia que llega a revisión, no solo la redacción de la respuesta.

Cómo evaluar esta arquitectura como sistema de trabajo

¿Reduce falsos cierres, retrabajo y pérdida de estado, aun si no aumenta el acierto bruto?

DISEÑO PROPUESTO

1 · AGENTE DIRECTO

objetivo → respuesta
sin expediente de ejecución

→

2 · SKILL SOLA

procedimiento reutilizable
sin corrida trazada

→

3 · WORKFLOW TRAZADO

harness · handoffs · gates
estado y evidencia persistente

→

4 · MULTIAGENTE

roles separados
revisión y transferencia explícita

MÉTRICAS

CONFIABILIDAD

$pass^k$ · falsos cierres · estabilidad entre corridas

CONTINUIDAD Y AUDITORÍA

tiempo para retomar · evidencia completa · revisión reproducible

COSTO OPERACIONAL

tiempo · tokens · intervención humana · retrabajo

La prueba no es solo si acierta más: es si deja menos cierres prematuros y más evidencia reutilizable.

Diseño de evaluación: mismas tareas, mismo modelo y presupuesto; varias corridas por condición; criterios externos definidos antes de observar resultados.