

Working paper — borrador académico

No revisado por pares ni publicado en revista. Borrador v5 (2026). El contenido puede cambiar. © José Luis D'Andrade · IdentIA / Self-Improving Society. **Todos los derechos reservados.** Cita breve con atribución permitida. **Queda prohibida** la reproducción, traducción, adaptación o reutilización, total o parcial (incluido uso comercial), sin **autorización previa por escrito** del titular.

Instituciones persistentes de agentes como unidad de despliegue: invariantes de identidad, gobernanza-como-código y una compuerta de fiabilidad medida

Autor: José Luis D'Andrade **Afiliación:** LegalDesign.DO **Proyecto:** IdentIA — identia.do **Contacto:** www.linkedin.com/in/josedandrade **Borrador:** v5.0 (sistemas) — informe técnico extendido. Parte [paper-v4-es.md](#) (retirado el 2026-08-27; en el historial de git). **Paper de posición compañero:** [paper-v5-position-es.md](#) — *La escalera de selección: una restricción de orden para la auto-mejora recursiva acotada*. Citado abajo como **[PB]**. **Fecha:** Agosto 2026 · **Revisado:** 2026-09-28 (retirada, §7.3) · **Estado medido contra:** el árbol consolidado del repositorio de 2026-08-23, commit `90792f5`, salvo donde se indique una fecha posterior — ver Apéndice C **Compañeros:** [whitepaper/whitepaper-v5-es.md](#) (posicionamiento comercial), [paper-v5-systems.md](#) (inglés)

Versión de solo texto principal, generada el 2026-09-28 a partir de [paper-v5-systems-es.md](#): sin agradecimientos, declaración de IA, referencias ni apéndices. Los `[n]` y las menciones a los Apéndices C/D/E remiten al fichero completo. No se versiona.

Resumen

La mayoría de los sistemas multi-agente con modelos de lenguaje (LLM) tratan a los agentes como *procesos efímeros*: se instancian para una tarea, se usan y se descartan. Es una decisión de ingeniería defendible, pero obliga a improvisar toda pregunta difícil sobre identidad, continuidad y verdad colectiva en el momento en que aparece. Presentamos **Self-Improving Society (SIS)**, un sistema cuya unidad de despliegue no es un agente sino una **unidad organizativa soberana** —un departamento, un ministerio, un grado escolar, un despacho— con identidad persistente, memoria privada y pública, cola local de tareas, gobernanza y capacidad de federarse con otras unidades.

Este paper reporta la mitad **construida y medida** de ese sistema. Primero, **IdentIA**, una ontología de ingeniería que lee identidad, cambio y verdad del agente a través de tres principios nombrados —*Impulso Dinámico* (la identidad se actualiza con la práctica), *Continuidad Absoluta* (la identidad persiste bajo cambio de modelo/runtime: un **estado persistente** distinto de su **realización computacional transitoria**) y *Pragmatismo Emergente* (lo que una institución puede tratar como conocimiento es lo que puede validar colectivamente —*creencia institucionalmente garantizada*, no verdad)— y que se operacionaliza como **invariantes de código**

aplicables (archivar-no-borrar, supersedir-no-sobrescribir, ninguna verdad unilateral). Segundo, la **arquitectura** desplegada: unidades soberanas `agents-host`, una capa social «exoharness» que gobierna y conecta un runtime de agentes intercambiable en vez de reimplementarlo, y federación que preserva la soberanía. Tercero, los **peldaños implementados de una escalera de selección medida** —reputación Beta–Bernoulli con incertidumbre explícita, rating relativo Bradley–Terry/TrueSkill, y una **compuerta de fiabilidad** por Teoría de Respuesta al Ítem— con un resultado en vivo fechado. Cuarto, **gobernanza-como-código** (L1–L8) y un **protocolo de evaluación** controlado sobre un «gym» determinista de tareas canónicas que fabrica la densidad de comparaciones uno-a-uno que los estimadores requieren.

Los peldaños superiores de la escalera, el argumento de por qué está *ordenada* así, y la tesis normativa sobre el poder institucional **no** están deliberadamente en este paper. No han corrido, y mezclar un argumento no probado con un sistema que sí se midió es la manera en que un paper de sistemas adquiere afirmaciones que no puede sostener. Son el objeto del paper de posición compañero **[PB]**.

Qué demuestra y qué no demuestra este paper. Reporta **infraestructura para auto-mejora acotada**, no auto-mejora. F0–F2 están implementados; F3–F4 corren en sombra; F5 es solo diseño. El bucle cerrado —variación, selección, mutación, comportamiento posterior mejorado— **no** ha operado.

Un resultado retirado, y por qué sigue aquí en vez de haber desaparecido. Versiones anteriores de este paper reportaban que la única corrida en vivo fechada de F2 devolvió `converged=False`, y leían esa negativa como la compuerta de fiabilidad declinando actuar sobre su propia señal poco fiable. **§7.3 retira esa lectura.** Al volver a medir el 2026-09-26 se encontró que la bandera de convergencia no podía valer `True` bajo el criterio que usábamos, con cualquier dato: el cambio por iteración se medía antes de un paso de identificabilidad que lo deshace en parte. Una bandera que no puede valer `True` no lleva información, así que la corrida no demostró nada sobre la compuerta. Mantenemos el episodio en el paper porque la retirada es el método funcionando: la fila del ledger que llevaba esa cifra llevaba también el límite *no se ha vuelto a medir desde entonces*, y ese límite es lo que produjo la nueva medición. De los 21 componentes que registra la tabla de §7.4, 9 están implementados y probados, 6 están implementados pero no obligan a nada, 2 cargan un defecto nombrado, y 4 son solo diseño.

Palabras clave: sistemas multi-agente; identidad y persistencia de agentes; teoría de respuesta al ítem; LLM-como-juez; gobernanza de IA; sociedades de agentes; federación.

1. Introducción

1.1 Motivación

La forma en que un sistema organiza la información determina la forma en que organiza el poder. Durante tres años el patrón dominante en agentes LLM ha sido el *orquestrador*: un controlador instancia agentes especializados por rol, enruta mensajes entre ellos, recoge una salida y desmantela los agentes. Frameworks como AutoGen [18], MetaGPT [17], ChatDev [19], CrewAI y LangGraph han hecho este patrón productivo y popular. Comparten una ontología tácita: el agente es una llamada de función con un prompt de sistema; su "memoria" es un búfer o un vector store compartido; su identidad es un id de sesión.

Esta ontología es adecuada para la automatización de un solo disparo y cada vez más inadecuada para los sistemas que la gente quiere construir hoy: **organizaciones persistentes de agentes** que acumulan competencia, mantienen una memoria institucional verificable, sobreviven a las actualizaciones de modelo, y pueden ser *confiables* y *auditable*s como lo es un departamento regulado. Cuando la unidad de valor es una institución y no una tarea, la ontología del proceso-descartable falla exactamente donde importa — en las preguntas de identidad ("¿es este el mismo agente certificado después de que cambiamos su modelo?"), de continuidad ("¿qué pasa con todo lo que aprendió este agente cuando lo retiramos?"), y de verdad colectiva ("dos agentes produjeron hallazgos contradictorios; ¿cuál es autoritativo?").

1.2 Planteamiento del problema

Estudiamos el diseño de **sociedades de agentes persistentes, soberanas y verificables**. Tres subproblemas se repiten y resisten la ontología del orquestador:

1. **Identidad bajo cambio de medición.** El comportamiento de un agente es producido por un modelo que será intercambiado, afinado o re-alojado. ¿Qué es, entonces, el agente a través de estos cambios, de un modo lo bastante preciso como para satisfacer a un auditor?
2. **Conocimiento institucional sin un ganador silencioso.** En una sociedad, la afirmación de ningún agente individual está *ipso facto* garantizada. El sistema necesita un modo principiado de dejar que la **creencia institucionalmente garantizada** emerja de la validación colectiva, y de registrar la sustitución sin borrar el linaje.
3. **Auto-mejora medida sin riesgo ilimitado.** Una sociedad que escribe sus propias skills y edita su propio andamiaje puede derivar o explotar (gaming) cualquier recompensa ingenua. ¿Cómo puede introducirse la presión de selección de forma *gradual*, con compuerta sobre la propia capacidad del sistema de medir la mejora de forma fiable?

1.3 Contribuciones

1. **IdentIA**, una ontología de ingeniería (§3) que nombra tres principios —*Impulso Dinámico* (A), *Continuidad Absoluta* (B), *Pragmatismo Emergente* (C)— y, sobre todo, los traduce a **invariantes a nivel de código** que la gobernanza aplica en vez de afirmarlos en prosa. **Introducimos la tupla de identidad** $\Psi(p) = \langle \text{SOUL}, M, L, K \rangle$ y la **medición** $m_t(p) = \langle R_t, \theta_t \rangle$ (Definiciones 1 y 2, §3.1): la identidad como objeto persistente formalmente separado de su realización transitoria, con la consecuencia verificable de que un cambio de modelo o de runtime no cambia Ψ . La distinción entre estado persistente y realización transitoria recibe así una lectura precisa como criterio de desempate para el diseño.
2. Una **arquitectura de sistema** (§4) cuya unidad de despliegue y escalado es la *unidad organizativa soberana* (`agents-host`), envuelta por una capa social «exoharness» que *gobierna y conecta* un runtime de agentes intercambiable en vez de reimplementarlo, con **federación** que preserva la soberanía.
3. **Los peldaños implementados, y una retirada** (§5–§7). Reportamos los peldaños implementados de la escalera de selección medida —reputación Beta–Bernoulli bajo cota inferior de Wilson, rating relativo Bradley–Terry/TrueSkill con juez pareado anti-sesgo-de-posición, y ajuste de dificultad por Teoría de Respuesta al Ítem— junto con el único ejercicio en vivo fechado del peldaño IRT (§7.2) **y la retirada de lo que antes concluíamos de él** (§7.3). Habíamos reportado una compuerta de fiabilidad que se negó; al volver a medir, la negativa resultó ser una propiedad de nuestro criterio de convergencia y no de los datos. **Ningún resultado en vivo establece hoy que la compuerta retenga la activación**, y decimos qué

ocupó el lugar de esa afirmación: un defecto encontrado por la propia contabilidad del paper. El *argumento del orden* que convierte esa compuerta en precondition y no en adorno se desarrolla en **[PB]**, y no queda afectado — nunca fue una afirmación empírica.

4. **Gobernanza-como-código** (§6) —un esquema de ocho niveles (L1–L8) que mapea el riesgo al modo de aprobación, con presupuestos por objetivo y cuarentena— junto con un **protocolo de evaluación controlado** (§7) centrado en un *gym* determinista, y un **ledger de claims** (Apéndice C) donde cada afirmación de estado lleva su fuente, método, fecha, entorno y límite. El ledger existe porque el mismo hecho se había registrado de tres maneras distintas en tres documentos; una cifra volátil que se transcribe en vez de derivarse, deriva.

1.4 Estructura del documento

§2 reseña la frontera y posiciona SIS. §3 formaliza IdentIA. §4 presenta la arquitectura. §5 da los peldaños implementados y la compuerta de fiabilidad. §6 cubre gobernanza. §7 da el protocolo de evaluación, el resultado en vivo de F2 y el estado de implementación. §8 lista amenazas a la validez. §9 discute interpretación, limitaciones y ética. §10 concluye. Los apéndices dan un glosario (A), un ítem de la suite del gym y por qué importa (B), el ledger de claims (C), el análogo humano y el mapa de comportamiento del formalismo de identidad, que §3.1 resume y referencia en vez de cargar en línea (D), y las versiones comprobadas por máquina (Lean 4) de las definiciones y reglas (E).

2. Trabajo relacionado

Agrupamos el arte previo por la dimensión a lo largo de la cual SIS se relaciona con él. Nuestra afirmación no es que cualquier mecanismo individual carezca de precedente — varios son adopciones deliberadas de estimadores del estado del arte — sino que su *integración bajo una sola ontología de identidad* es nueva. Somos explícitos respecto de los solapamientos.

2.1 Sociedades de agentes generativos y simulaciones de civilización

Los **Generative Agents** de Park et al. [1] poblaron un sandbox ("Smallville") con 25 agentes LLM equipados con un *flujo de memoria* (memory stream), una función de puntuación de recuperación, y *reflexión* periódica, produciendo comportamiento individual creíble y comportamiento de grupo emergente. **Project Sid** de Altera [2] escaló a 10–1000+ agentes en Minecraft bajo la arquitectura **PIANO** (Parallel Information Aggregation via Neural Orchestration), observando especialización emergente de roles, cambio colectivo de reglas, y transmisión cultural/religiosa. **OASIS** de CAMEL [3] simula hasta un millón de agentes sobre dinámicas de redes sociales.

Estos sistemas estudian el **comportamiento emergente** como objeto científico. SIS difiere en *propósito y unidad*: su agents-host es una **unidad de despliegue productiva y soberana** (Agent-as-a-Service) que ejecuta pipelines *auditados* con artefactos y procedencia, federándose con otras unidades mediante envelopes estructurados (§4.4). La transmisión cultural en SIS no es una curiosidad emergente sino un **punto gobernado** (el puente A → C, §3.4) con compuertas humanas. SIS toma prestada la idea de que los registros destilados y atribuidos ("actas", no transcripciones crudas) son el sustrato correcto para la memoria social.

2.2 Agentes auto-mejorantes y auto-diseñadores

Reseñados en el paper de posición compañero, **[PB] §2**, donde la comparación hace trabajo: la relación de SIS con Voyager, la Darwin Gödel Machine, PromptBreeder y la línea de agentes auto-diseñadores es una afirmación sobre **orden** —qué debe medir un sistema antes de que se le permita editarse a sí mismo— y ese es el objeto de **[PB]**. El único hecho que este paper necesita de allí: ninguno de esos sistemas condiciona la auto-modificación a una propiedad medida de fiabilidad de su propia señal de fitness, que es lo que reportan §5.6 y §7.2.

2.3 Memoria e identidad de agentes

MemGPT/Letta [9] introdujo memoria jerárquica inspirada en sistemas operativos para dar a los agentes persistencia más allá de la ventana de contexto. El memory stream de Generative Agents [1] es el otro diseño influyente. La *cuestión de la identidad misma* se ha convertido recientemente en un tema explícito de investigación: trabajo de posición que argumenta que los agentes LM no son agentes "normales" porque las patologías del modelo subyacente —ausencia de estado, estocasticidad, sensibilidad al prompt— desestabilizan su identificabilidad, continuidad, persistencia y consistencia [10], y evaluaciones tempranas de **identidad de agentes** [11].

SIS aporta **IdentIA** (§3): identidad modelada como un **estado persistente** (ancla SOUL, memoria, linaje, conjunto de skills) frente a *mediciones* transitorias (runtime y modelo activos), operacionalizada en invariantes — archivar-no-borrar, sustituir-no-borrar, ninguna-verdad-unilateral — y un modelo de memoria de dos planos (privado A/B vs. social C) con un puente de transmisión gobernado. No tenemos conocimiento de trabajo previo que convierta una ontología de identidad en *invariantes arquitectónicos exigidos* de esta manera; la *cuestión* subyacente (persistencia/continuidad), sin embargo, se estudia ahora activamente [10, 11], lo que tratamos como evidencia convergente de oportunidad más que como implementación previa de nuestro mecanismo. Synergy [31] eleva ahora identidad-y-personhood a requisito arquitectónico explícito (perfil, memoria tipada, agenda, relaciones sociales persistentes) — más evidencia convergente; IdentIA difiere en hacer la identidad *exigible* (invariantes, §3.2) e institucional (la sociedad, no el agente individual, es la unidad de despliegue).

2.4 Selección, rating y calidad-diversidad

El linaje de estimadores —Bradley–Terry, TrueSkill, Teoría de Respuesta al Ítem, muestreo de Thompson, NSGA-II, MAP-Elites— se reseña en **[PB] §2**. Cada estimador que se usa aquí se toma sin cambios de esa literatura y se cita en su punto de uso en §5. Este paper no reclama nada novedoso sobre ninguno de ellos; lo que reclama es sobre qué están cableados a condicionar.

2.5 Gobernanza y alineación

Constitutional AI [25] gobierna el *comportamiento del modelo* en tiempo de entrenamiento mediante retroalimentación de IA contra una constitución escrita. La gobernanza de SIS es *operacional e institucional*: un esquema de ocho niveles sobre una sociedad en ejecución — que controla infraestructura, plantillas base, instanciación de agentes, promoción en el marketplace, y el pin del runtime — con presupuestos por objetivo y cuarentena (§6). Son capas complementarias (alineación en tiempo de entrenamiento vs. control institucional en tiempo de ejecución).

2.6 Resumen de posicionamiento

Dimensión	Arte previo representativo	Posición de SIS
Sociedades de agentes	Generative Agents [1], Project Sid [2], OASIS [3]	Unidad productiva soberana; pipelines auditados; puente cultural gobernado
Auto-mejora	Voyager [4], ADAS [5], DGM [6], AlphaEvolve [7]	Escalera ordenada por seguridad ; selección hecha confiable (y <i>medida</i> confiable) antes de actuar
Memoria / identidad	MemGPT [9], [10], [11]	Ontología IdentIA como invariantes de código exigidos
Estimadores de selección	BT [12], TrueSkill [13], IRT [20], MAP-Elites [14]	Adoptados; asignados por objeto + progresión autoimpulsada , sin dependencias de terceros
Gobernanza	Constitutional AI [25]	L1–L8 operacionales ; tesis de metacentralidad ([PB] §5)
Frameworks de orquestación	AutoGen [18], MetaGPT [17], ChatDev [19]	Exoharness : gobernar/conectar un runtime, no reimplementarlo
Ciudadanía de la web agéntica	Synergy [31]	La institución como unidad de ciudadanía : selección supeditada a la fiabilidad; transmisión supeditada a la validación; federación supeditada a una política
Auto-reescritura del arnés	Meta-Harness [32], Continual Harness [33]	El arnés se envuelve, no se reescribe : la institución porta identidad y juicio; el runtime de debajo es reemplazable

2.7 Cuatro polos del campo adyacente, y la compuerta que ninguno aplica

Los vecinos más próximos a este trabajo se agrupan en cuatro polos. Nombrarlos con precisión importa, porque lo que aquí se afirma no es un mecanismo que a ellos les falte —varios tienen mejores— sino **una compuerta que ninguno aplica**.

Polo 1 — evolución multi-agente académica. Synergy [31] mapea casi uno a uno con los principios de IdentIA: colaboración, identidad y evolución continua. Sus límites son metodológicos antes que arquitectónicos: la confianza se pondera de forma blanda en vez de someterse a una compuerta, la adjudicación es por juez LLM, y la evaluación es solo de benchmark. Nada en ese diseño **se niega** a actuar sobre una señal poco fiable; la pondera menos.

Polo 2 — composición sin permiso. Redes en las que una capacidad se propaga por adopción: un agente o un artefacto se extiende porque otros lo toman, y la adopción económica sustituye a la selección medida. Es el principio C —construcción colectiva de verdad— con el paso de validación retirado. Es el contra-caso útil de este paper, porque muestra qué aspecto tiene un proceso de selección cuando nada lo condiciona: la adopción es rápida, y que una afirmación sobreviva no dice nada sobre si alguna vez fue correcta.

Polo 3 — auto-reescritura del arnés in situ. Continual Harness [33] formaliza un bucle sin reinicios en el que un agente altera *su propio* prompt, sus subagentes, sus skills y su memoria mientras corre, y reporta una reducción sustancial del costo de acciones frente a una línea base minimalista, recuperando la mayor parte —no toda— de la distancia a un arnés experto construido a mano. Meta-Harness [32] busca código de arnés mejorado de extremo a extremo a partir de registros de ejecución, y reporta una mejora de 7,7 puntos sobre un gestor de

contexto del estado del arte usando la cuarta parte de los tokens de contexto. Los agentes bifurcados que refinan sobre una base inmutable con rollback pertenecen al mismo polo. Son resultados de ingeniería sólidos y no los discutimos.

Polo 4 — plataformas industriales de fuerza de trabajo de agentes. En febrero de 2026, OpenAI Frontier [34] convirtió *los agentes como empleados* en producto empresarial entregado: se contratan, se incorporan, reciben permisos, retroalimentación estructurada y evaluación de desempeño sobre una capa compartida de gobernanza y memoria institucional; salió con HP, Intuit, Oracle, State Farm, Thermo Fisher y Uber. ServiceNow Autonomous Workforce, Microsoft Foundry Agent Service, Salesforce Agentforce y CrewAI Enterprise ocupan el mismo polo. Enunciamos la consecuencia para nuestro propio posicionamiento sin rodeos: **un organigrama para agentes — roles, permisos, evaluación de desempeño— dejó de ser una arquitectura distintiva a comienzos de 2026.** Todo encuadre que se apoye en eso describe el campo en vez de separarse de él, y este trabajo no se apoya en eso.

Lo que este polo gobierna son **agentes**: su identidad dentro de un inquilino, su acceso, su producción medida. Lo que no constituye es una **institución con frontera** — una unidad que conserva la procedencia cuando cambian los agentes que la componen, que puede negarse a transmitir lo que no ha validado, y que puede federarse con una institución par bajo una política que no pertenece a ninguna de las dos. La memoria de Frontier es institucional en el sentido de *compartida dentro de una empresa*; no lo es en el sentido de *portable a través de una frontera organizacional bajo una relación de confianza explícita*. La diferencia es de unidad, no de calidad: en distribución, profundidad de integración y madurez operativa este polo va por delante, y no afirmamos lo contrario.

Y en el eje del que trata este trabajo, el polo 4 comparte el límite de los otros tres. La evaluación de desempeño de un agente mide su *producción*; no mide si la señal que impulsa un cambio es lo bastante fiable como para actuar sobre ella. Nada en estas plataformas se niega a actuar sobre una señal poco fiable: la elevan a un revisor humano, que es un mecanismo distinto con un modo de fallo distinto.

Lo que los cuatro polos comparten es la entrada, no la salida. Todos refinan a partir de *las trayectorias que tengan*. Ninguno condiciona el refinamiento a una medición de si la señal que lo impulsa es lo bastante fiable como para actuar sobre ella. Esa es la diferencia que este paper reporta, y §7.3 enuncia su límite actual sin adornos: la compuerta está construida y probada, y **ninguna corrida en vivo la muestra todavía reteniendo la activación** — la corrida que antes citábamos para eso queda retirada.

Hay una segunda diferencia, arquitectónica y no metodológica. El polo 3 reescribe el arnés de un agente. Este trabajo no toca el arnés: lo **envuelve**. La unidad que porta identidad, frontera, procedencia y juicio es la **institución**, y el runtime de agente que hay debajo es reemplazable por construcción (§4).

2.8 Convergencia: tres líneas independientes, un mismo suelo

Lo más interesante de los cuatro polos no es en qué se diferencian de este trabajo, sino **en qué coinciden entre sí sin haberse coordinado**.

Synergy [31] llega a una arquitectura que se mapea casi uno a uno con los tres principios de IdentIA: su *collaboration* con nuestro principio de construcción colectiva, su *identity* con el de continuidad, y su *lifelong evolution* con la unión de ambos. Los agentes bifurcados que refinan sobre una base que no pueden alterar, con

posibilidad de revertir, reconocen en la práctica —aunque no lo enuncien— la misma condición de preservación que aquí se llama principio B. Y tanto Continual Harness [33] como Meta-Harness [32] parten de que el material del que se aprende son **registros de ejecución**, no impresiones.

Base inmutable, evidencia, reversibilidad. Tres líneas de trabajo independientes, con motivaciones distintas, llegaron al mismo suelo.

Esa convergencia es el argumento, y no es nuestro. Cuando equipos que no se coordinan resuelven un problema de la misma manera, la explicación económica no es que a todos se les ocurriera lo mismo: es que **el problema fuerza la forma**. Un diseño al que se llega desde tres sitios distintos deja de ser una preferencia de sus autores.

Lo que **no** encontramos en ninguna de las tres líneas es el paso siguiente: **medir si la señal que guía el refinamiento es lo bastante fiable como para actuar sobre ella, y negarse cuando no lo es**. Todas refinan; ninguna se abstiene. Ahí es donde este trabajo se sitúa — y §7.3 dice sin adornos hasta dónde llega esa afirmación hoy: la compuerta está implementada y probada, y **ninguna corrida en vivo la muestra todavía reteniendo la activación**; la corrida que antes citábamos para eso queda retirada.

3. IdentIA: una ontología de ingeniería

IdentIA fija una *posición* sobre identidad, cambio y verdad de modo que los casos difíciles tengan una respuesta principiada en lugar de una ad-hoc. Es un **criterio de desempate y una prueba de coherencia para funcionalidades**, no una teología de cara al usuario: una sociedad puede funcionar sin que nadie la invoque por su nombre.

La postura de que una ontología puede *ingeniarse* — que las decisiones de arquitectura son posiciones filosóficas, y deben examinarse como tales — es la premisa de trabajo de la tradición del mind design [30]. Que una "mente" pueda ser una *organización persistente de mecanismo* y no un formalismo abstracto se remonta al cálculo lógico de la actividad nerviosa de McCulloch y Pitts y a *Embodiments of Mind* de McCulloch [29]: los perfiles de SIS — cuerpos (runtimes), memorias y hábitos aprendidos bajo un ancla persistente — son agentes en ese sentido corporizado, no llamadas a función desechables. Cada principio tiene además antecedentes con nombre, que registramos por honestidad intelectual y no como argumento de autoridad: el Impulso Dinámico (A) en el pensamiento procesual y falibilista (Whitehead, Popper, Prigogine) y en la práctica moderna del aprendizaje por interacción (LeCun); la Continuidad Absoluta (B) en la metafísica de la sustancia y del orden implicado (Platón, Spinoza, Bohm, Tegmark); el Pragmatismo Emergente (C) en el pragmatismo y la racionalidad acotada (Dewey, Simon) y en la primacía empírica de la búsqueda y el aprendizaje sobre el conocimiento codificado a mano (Wolfram, Sutton). Los principios se sostienen o caen por los invariantes de §3.2, no por este linaje.

3.1 Tres principios

Sea un *perfil* p una identidad de agente persistente.

Definición 1 (Identidad del agente). La identidad de p es la tupla

$$\Psi(p) = \langle \text{SOUL}(p), M(p), L(p), K(p) \rangle,$$

donde $\text{SOUL}(p)$ es el ancla de identidad (un documento de carácter/rol escrito por un humano o un sistema), $M(p)$ es la memoria del perfil (procedimental, semántica, episódica; §3.3), $L(p)$ es su linaje (aristas clone-from, procedencia de skills, historial de sesiones), y $K(p)$ es su conjunto de skills — las habilidades escritas y adquiridas del perfil con su procedencia, nombrada como componente propio (§D.1).

Definición 2 (Medición). Una *medición* de p en el tiempo t es la tupla

$$m_t(p) = \langle R_t, \theta_t \rangle,$$

el runtime activo R_t (harness, herramientas) y el modelo activo θ_t .

Las dos definiciones son propias de este paper. Hasta donde sabemos, ningún trabajo previo da la identidad de un agente como un objeto persistente formalmente separado de su realización: los más cercanos, Perrier y Bennett [10][11], tratan identificabilidad, continuidad y persistencia como *propiedades a medir* de un agente de modelo de lenguaje, y Synergy [31] enuncia perfil y memoria tipada como *requisitos* de un ciudadano agéntico. Ninguno convierte «un cambio de modelo no cambia Ψ » en una consecuencia comprobable; aquí es el contrato de verificación del principio B (§3.2). Las dos definiciones, y el teorema de invariancia, están comprobados por máquina (Apéndice E).

Una nota sobre toda la notación de este paper. Los símbolos son un *vocabulario de especificación formal*, no un modelo matemático cuantitativo. No hay espacios métricos, operadores ni distribuciones definidas sobre Ψ ; la notación nombra componentes arquitectónicos tipados y las restricciones sobre ellos. Leída como especificación, hace trabajo. Leída como matemática, sería sobrefirmar.

Los tres principios son:

- **(A) Impulso Dinámico (Dynamic Striving).** La identidad se actualiza con la práctica. La autoría de skills, la consolidación por el curador, la ingesta de RAG, y las escrituras de memoria todas *incrementan* el grado en que p es sí mismo. Formalmente, A respalda el enriquecimiento monótono de $\Psi(p)$ mediante la acción; la brecha entre la capacidad actual y la potencial es capacidad latente, no fracaso.
- **(B) Continuidad Absoluta (Absolute Continuity).** $\Psi(p)$ persiste bajo cambio de medición. Intercambiar $\theta_t \rightarrow \theta_{t'}$ o $R_t \rightarrow R_{t'}$ **no** cambia la identidad siempre que $\Psi(p)$ se preserve. La distinción es entre un **estado persistente** y su **realización transitoria**: Ψ es el objeto que la teoría protege a lo largo del tiempo; m_t es una lectura de cómo se está ejecutando ese objeto ahora mismo. El principio entero autoriza una sola regla de ingeniería — **nunca destruir Ψ para satisfacer una medición**.
- **(C) Pragmatismo Emergente (Emergent Pragmatism).** Lo que una institución puede tratar como conocimiento no es la afirmación de un solo perfil; es lo que la institución puede *validar*: por uso, revisión por pares, gobernanza, fitness del evaluador, y promoción en el marketplace. Llamamos a esa salida **creencia institucionalmente garantizada**, no verdad, y la distinción no es decorativa: una sociedad puede converger y aun así estar equivocada. El esquema ya codifica esa humildad —los hechos se **superan**, nunca se convierten en verdades atemporales— y el vocabulario se alinea aquí con el esquema en lugar de con la ambición. El consenso no se decreta; es lo que sobrevive a la aplicación exitosa a través de contextos distintos, y sigue siendo revisable por construcción.

Dónde se desarrolla el formalismo. Dos subsecciones que estaban aquí en v4 —el análogo humano que motiva la tupla e introduce el término residual Ω , y el mapa de comportamiento $b_t = f(\Psi, m_t) + \varepsilon_t$ que dice dónde vive el Ω de un agente— son el **Apéndice D** (§D.1, §D.2). Son el pasaje formal más denso del paper y sostienen §3.2 y §9.2, pero no hacen falta para leer los invariantes, la arquitectura ni las mediciones. La consecuencia en una línea, para quien no vaya al apéndice: **Ω no es un componente del objeto de identidad.** Todo componente de Ψ debe ser persistente, inspeccionable y aplicable; Ω no es ninguna de las tres, así que meterlo dentro de Ψ haría inverificable la condición de preservación del principio B e infalsables las afirmaciones de identidad. Ω entra en cambio por el mapa de comportamiento, y la gobernanza acota su efecto en vez de que la ontología lo modele.

3.2 Invariantes (lo que la ontología prohíbe)

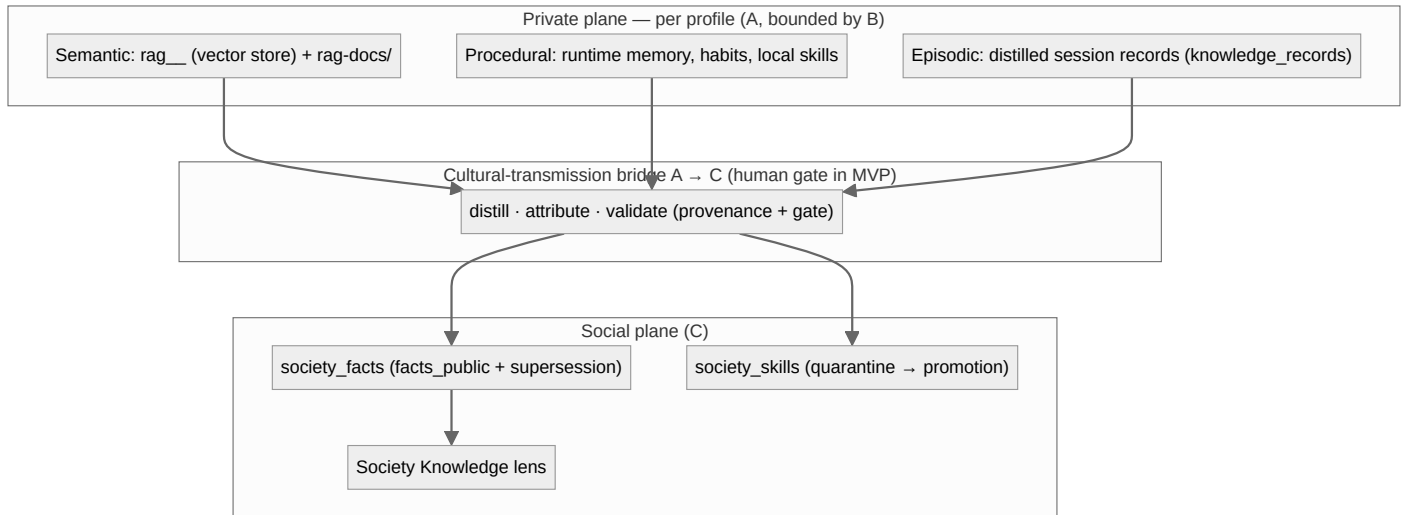
Los principios se hacen operacionales como invariantes exigidos por la gobernanza y el esquema, no por exhortación:

Invariante	Principio	Exigencia en SIS
Archivar, no borrar perfiles	B	El ciclo de vida por defecto es <i>archivar</i> ; el borrado destructivo requiere aprobación humana + exportación de memoria/RAG (§6).
Sustituir, no sobrescribir hechos	B + C	El esquema <code>facts</code> lleva <code>supersedes</code> / <code>superseded_by</code> ; los hechos incorrectos se <i>sustituyen</i> , el linaje se preserva.
Ninguna verdad unilateral	C	Los hallazgos contradictorios escalan a auditor/gobernanza/votación; el sistema nunca elige silenciosamente un ganador.
Linaje, no copia	B	<code>clone-from</code> y <code>forked-from</code> registran aristas de procedencia en el grafo.
Medición $\neq$ identidad	B	El modelo/runtime son configuración por perfil; cambiarlos no es un evento de identidad.
Mutación reversible (trinquete)	B	Todo cambio impulsado por fitness de skill/SOUL/config es reversible a un previo dominante en fitness (§5.8).

Una funcionalidad propuesta que requiera borrar perfiles, reemplazar (no parchear) skills, o dejar que un agente declare verdad pública unilateralmente es, por construcción, una violación de IdentIA y un candidato a refactorización. Esta *falsabilidad a nivel de diseño* es el rédito práctico de nombrar la ontología.

3.3 Dos planos de memoria

SIS distingue un **plano privado** (principio A, acotado por B en la frontera del perfil) de un **plano social** (principio C):



La frontera de privacidad/identidad es el **perfil, no el host**: un agente no ve ni el RAG de otro agente ni su memoria episódica, aun co-ubicados. El sustrato episódico es registrado por una vía de ingesta de *contexto episódico* (sesiones destiladas en `knowledge_records`); el lado de lectura de esa memoria episódica es solo diseño (§7.3).

3.4 El puente A → C: transmisión cultural gobernada

Los humanos no solo recuerdan individualmente; **transmiten** técnica a través de generaciones. SIS modela esto como un puente *gobernado* del plano privado al plano social. Un hallazgo privado se **destila, atribuye y valida** antes de convertirse en conocimiento social — nunca copiado en crudo, nunca declarado verdadero unilateralmente (eso violaría C). El puente tiene dos grados:

1. **MVP — transmisión con compuerta humana (vía implementada).** Un `knowledge_record` (p.ej. de tipo `fact`) se promueve a un `fact_candidate` y luego a `society_facts` solo con aprobación humana. Este es el primer acto de transmisión cultural, honesto y barato.
2. **Deuda — transmisión emergente agente-a-agente (solo diseño, diferida).** Un perfil estilo arquitecto lee la memoria de un perfil que ha dominado una skill (alta habilidad latente θ , §5) y la propaga a uno que carece de ella, *con validación antes de la propagación* ("no enseñar un error como mejor práctica"). Esto requiere políticas de confianza de escritura entre perfiles y fitness fiable; es riesgoso automatizarlo temprano y se registra explícitamente como deuda técnica diferida.

El puente se alimenta de dos fuentes con forma idéntica pero origen distinto: **individual** (la memoria episódica destilada de un agente) y **colectiva** (las *actas* de una asamblea/consejo entre hosts — decisiones destiladas con procedencia multi-participante, no una grabación cruda). Las actas convierten un evento social en memoria institucional consultable ("¿quién sabe sobre X?") y un índice de competencia para enrutar futuras consultas a los expertos correctos.

4. Arquitectura del sistema

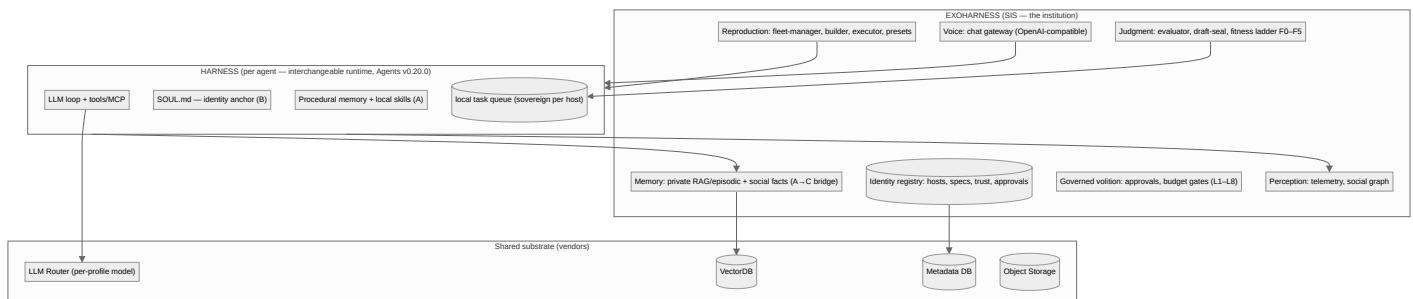
4.1 Principio: la unidad organizativa soberana

La unidad de despliegue y escalado no es un agente sino un **agents-host**: una unidad organizativa soberana completa con su propia identidad, cola de tareas local, gobernanza, evaluador, telemetría y agentes. Un operador describe una organización en lenguaje natural o YAML — "un banco con Riesgo, Cumplimiento, TI y Operaciones", "una escuela K-12", "los ministerios de un estado" — y el sistema aprovisiona un host por unidad y los federa. La escala horizontal es *un nuevo host*, no más runtimes en un host central. (Linaje: el proyecto se origina en una "mini-village" de tres VMs Linux aisladas, un agente cada una; cada VM es ahora un **agents-host** completo.)

4.2 Harness vs. exoharness

SIS no reimplementa el lazo del agente. Distinguimos:

- **Harness** (por agente, intercambiable): el lazo del LLM, herramientas/MCP, memoria procedimental, sesiones, el ancla SOUL, y una cola de tareas local. En el despliegue actual esto es el **runtime de Agents, fijado a v0.20.0** (SHA 9d4ef04ed, E-03; cuya herramienta de memoria acepta un arreglo **operations** por lotes, consolidando actualizaciones de memoria de múltiples escrituras en una sola llamada); el contrato es agnóstico del runtime.
- **Exoharness** (SIS): la capa institucional que envuelve a *todos* los agentes y les da identidad registrada, memoria social, juicio/selección, gobernanza, reproducción (ciclo de vida), y observabilidad. *Un agente sin el exoharness es un proceso; con él, un ciudadano de una institución.*



Sobre la palabra, porque algún lector la va a traer. Usamos *exoarnés* en el sentido que su raíz porta: *exo-* es «fuera», como en *exógeno* o *exoesqueleto* — una estructura que **envuelve** lo que hay dentro y le da frontera, protección y forma sin sustituirlo. Esa es la relación de SIS con el arnés del agente: el arnés conserva su bucle, sus herramientas, su memoria y su ancla de identidad; el *exoarnés* lo rodea con registro, memoria social, juicio, gobernanza y ciclo de vida. La misma palabra se usa en otro sitio con la geometría opuesta. La especificación **exoharness/exo** [36] define un exoharness como «a trusted substrate that manages durable state, brokers access to privileged resources, and provides low-level execution plumbing without owning agent semantics» — una capa **por debajo** del ejecutor, que la raíz llamaría *endo-*, o sencillamente núcleo. No reclamamos el término ni discutimos ese uso; anotamos que las dos palabras nombran capas distintas de la misma pila, la nuestra sobre el arnés y la suya debajo, y que en este paper la palabra significa siempre la primera. Un tercer nombre

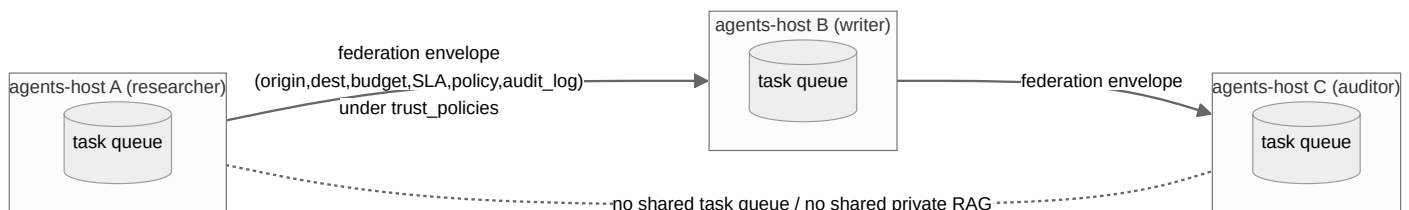
para nuestra capa también circula, *meta-harness*, y Meta-Harness [32] lo usa para un bucle externo que busca sobre el código del arnés; conservamos *exoarnés* porque lo que describimos no busca sobre el arnés: lo envuelve (§4.2, figura de arriba).

4.3 Capas de soberanía

Los elementos se particionan por alcance. La **capa soberana** (nunca compartida sin federación explícita) contiene la cola de tareas local (`kanban.db`, SQLite), el RAG privado por agente (`rag_<host>_<profile>`), y el registro de metadatos de la sociedad. La **capa de plataforma** (LLM router, VectorDB, almacenamiento de objetos, base de datos de metadatos) puede ser *compartida* (aislamiento lógico por namespace/colección/bucket/tenant — el predeterminado del piloto) o *dedicada* por host/tenant cuando la regulación lo requiera. La **capa federada** (`society_skills`, `society_facts`, artefactos, RAG público exportable) circula entre hosts con la procedencia preservada.

4.4 Federación

El trabajo entre hosts se transporta mediante un **federation envelope**: un registro estructurado y auditado (tabla `federated_tasks`: `envelope_id`, host de origen/destino, título, presupuesto, SLA, política, resultado, y un `audit_log` de solo-anexado) despachado por el fleet-manager a la cola de tareas de un host destino y gobernado por **políticas de confianza** (tabla `trust_policies`: qué host puede delegar a cuál, qué datos pueden cruzar). La cadena del pipeline vive **en los SOULs de los agentes, no en un orquestador central** — cada agente conoce solo su próximo paso — lo que es en sí mismo una declaración de IdentIA: Continuidad Absoluta (cualquier host reemplazable), Impulso Dinámico (cada agente avanza por su propio impulso), Pragmatismo Emergente (el pipeline emerge de traspasos auditados). *Nota de honestidad*: los envelopes están actualmente **estructurados y auditados pero no firmados criptográficamente**; la firma criptográfica es trabajo futuro (§7.4, §9.2). La etiqueta de prosa "envelopes firmados" en documentos internos anteriores es aspiracional; la confianza entre hosts descansa actualmente en la política más las fronteras de red.



4.5 Servicios

El plano de control comprende `fleet-manager` (perfiles, bootstrap de RAG, proxy de la cola de tareas, swarms, Society Builder, consumidor de fitness-routing), `governance` (aprobaciones, compuertas de presupuesto), `evaluator` (verificaciones deterministas + la escalera de fitness), `society-telemetry` (eventos de grafo, heartbeats, snapshots, agregación federada), `society-executor` (ejecución de acciones de operador/arquitecto, incluyendo el despliegue de federación), y un **Conversation Gateway** compatible con OpenAI `society-chat`. Los **servidores MCP** por agente (stdio) exponen `society-rag`, `society-skills`, `society-artifacts`, `society-facts`, `society-telemetry/graph-query`, `society-vision`, y `society-audio` (y un `society-ros2` de solo-contrato). El chat gateway, por regla de producto, **no escribe memoria y no crea tareas automáticamente**; un `goal_candidate` derivado del chat requiere aprobación humana antes de convertirse en objetivo.

El mapa de puertos canónico (del manifiesto de despliegue en `infra/`): API de `agents-host` :8642 y `dashboard` :7474; `fleet-manager` :8088; `governance` :8090; `society-telemetry` :8091; `evaluator` :8092; `society-graph-ui` :8093; `society-chat` :8094; `society-executor` :8099; LLM Router :4000; VectorDB :6333; Metadata DB :5432; Object Storage :9000 / :9001 .

5. Selección medida: los peldaños implementados y la compuerta de fiabilidad

Esta sección reporta los peldaños que existen, están probados y han corrido. La escalera tiene seis peldaños; tres están aquí. Qué harían los otros tres, y el argumento de por qué tienen que venir en ese orden, es **[PB]**.

5.1 Variación vs. selección

El runtime del agente ya provee un **operador de variación**: escribe y parchea sus propias skills y consolida memoria en una revisión de auto-mejora. Lo que falta es un **operador de selección medida**: una forma de decidir *cuáles* de esos cambios son realmente mejores y deberían sobrevivir, propagarse (el puente A → C), y anclarse (principio B). Esquemáticamente:

$$\text{RSI} = \underbrace{\text{Variation}}_{\text{exists (runtime)}} \circ \underbrace{\text{Selection}}_{\text{this ladder}} \text{ in a closed loop, editing the scaffold, gated by fitness.}$$

La variación sin selección es **deriva**; la selección sin variación no tiene nada que elegir. El encuadre es deliberadamente darwiniano [28]: la mejora nunca la *declara* el objeto que cambió; la *selecciona* la comparación medida contra alternativas en un entorno. La escalera construye la selección primero, y solo su peldaño superior cierra el lazo recursivo.

5.2 Cinco objetos de fitness

No hay un único "número de fitness". Cinco objetos portan señal de forma diferente:

Objeto	Señal cruda	Tabla fuente	Estimador objetivo
Skill	<code>usage_count</code> , <code>success_count</code>	<code>skill_publications</code>	Posterior Beta → Bradley–Terry/TrueSkill
Perfil/agente	éxito/fracaso de tarea, heartbeats	Kanban <code>task_runs</code> , <code>runtime_heartbeats</code>	IRT (ajustado por dificultad) + reputación de grafo
Hecho	sustitución, citas	<code>facts</code> / <code>society_facts</code>	tasa de supervivencia/sustitución + predicción-por-pares
Repertorio	diversidad de skills/perfiles activos	<code>fitness_repertoire</code>	Calidad-Diversidad (MAP-Elites)
Enrutamiento	quién recibe la próxima tarea	<code>fleet_manager</code> , despacho Kanban	Muestreo de Thompson sobre posteriors

5.3 Los peldaños y sus self-triggers

Cada peldaño es un hito que declara un algoritmo, los datos que necesita, un **self-trigger** (una condición que el sistema mide sobre *sí mismo* para avanzar), su principio dominante de IdentIA, y su nivel de gobernanza. Los umbrales de self-trigger son configurables por entorno (`SOCIETY_FITNESS_*`). Los badges reflejan el código a junio de 2026:  implementado y probado;  implementado, en curso o shadow;  solo diseño.



Los tres peldaños que se detallan abajo —F0, F1 y F2— son los que están implementados y han corrido. F3, F4 y F5, y el argumento de que su orden es una restricción de seguridad y no una hoja de ruta, están en **[PB] §3–§4**.

5.4 F0 — reputación Beta–Bernoulli con incertidumbre explícita

Para un objeto con s éxitos en n intentos, se coloca un prior uniforme y se reporta el posterior Beta

$$\pi \sim \text{Beta}(\alpha, \beta), \quad \alpha = s + 1, \quad \beta = (n - s) + 1, \quad \mathbb{E}[\pi] = \frac{\alpha}{\alpha + \beta}.$$

El ranking **no** usa la media del posterior; usa el **límite inferior de Wilson** a confianza z ,

$$\hat{p}_{\text{LB}} = \frac{\hat{p} + \frac{z^2}{2n} - z \sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}}, \quad \hat{p} = \frac{s}{n},$$

de modo que un arranque en frío 2/2 no pueda dominar un historial de 40/45. La incertidumbre es de primera clase. El **resultado canónico** que define un "éxito" es una única función compartida (`canonical_task_outcome`) que devuelve `success` | `failure` | `inconsistent` | `pending` a partir de una tarea cerrada y sus runs, de modo que la verificación `task_outcome_present` del evaluador y la capa de fitness compartan una definición. **Estado: implementado y con pruebas unitarias** (`evaluator/fitness.py::beta_posterior`, `wilson_interval`; `tests/test_fitness_f0.py`, 8/8). Los contadores se alimentan en vivo por el plugin `society-observer` de Agents, que hace POST a `/fitness/skill-usage` en cada `skill_manage` exitoso; el recálculo corre en un ciclo periódico (`SOCIETY_EVALUATOR_FITNESS_CYCLE_SECONDS`, predeterminado 3600) y bajo demanda vía `POST /fitness/recompute`. El módulo importa solo la biblioteca estándar de Python (`math`, `typing`): no hay dependencia de rating de terceros.

IdentIA: A (la práctica actualiza el número). *Gobernanza*: L1/L2 (solo lectura). *Self-trigger a F1*: al menos `SOCIETY_FITNESS_MIN_PAIRWISE` (predeterminado 30) eventos uno-a-uno por objeto. Ese umbral es un **valor por defecto de ingeniería, no una cota de suficiencia derivada**; §5.9 registra que los umbrales de self-trigger de la escalera son elegidos y no demostrados, y este no es una excepción.

5.5 F1 — rating relativo: Bradley–Terry y TrueSkill

Para *medir* el "mejor-que" en lugar de afirmarlo, SIS califica skills con **Bradley–Terry** [12] y perfiles con **TrueSkill** [13].

Bradley–Terry. Con fuerzas latentes $\beta_i = \log \gamma_i$, la probabilidad de que i venza a j es

$$P(i \succ j) = \frac{\gamma_i}{\gamma_i + \gamma_j} = \sigma(\beta_i - \beta_j).$$

Las fuerzas se ajustan por la actualización de Minorización–Maximización (Hunter) con un prior suave; dados conteos de victorias w_{ij} y conteos de pares n_{ij} ,

$$\gamma_i \leftarrow \frac{\sum_{j \neq i} w_{ij}}{\sum_{j \neq i} \frac{n_{ij}}{\gamma_i + \gamma_j}}.$$

La puntuación de rango reportada se encoge por evidencia (un factor Wilson-relativo) de modo que rivales escasos no salten por encima de los bien probados. Esta es la misma familia usada por los modelos de recompensa y por Chatbot Arena [24]. **Estado: implementado y probado**

(evaluator/fitness.py::bradley_tery, artesanal sin dependencias externas; tests/test_fitness_fl.py).

Validado en vivo con artefactos reales: en un conjunto controlado, writer (4–0) > researcher (2–2) > auditor (0–4) produjo un ranking coherente.

TrueSkill. Cada perfil mantiene una creencia gaussiana $\mathcal{N}(\mu, \sigma^2)$. Tras un resultado 1-contra-1 con ruido de desempeño β e incertidumbre total $c^2 = 2\beta^2 + \sigma_i^2 + \sigma_j^2$, el ganador i y el perdedor j se actualizan

$$\mu_i \leftarrow \mu_i + \frac{\sigma_i^2}{c} v\left(\frac{\mu_i - \mu_j}{c}\right), \quad \sigma_i^2 \leftarrow \sigma_i^2 \left[1 - \frac{\sigma_i^2}{c^2} w\left(\frac{\mu_i - \mu_j}{c}\right)\right],$$

con las funciones estándar de gaussiana truncada $v(\cdot)$ y $w(\cdot)$ y un margen de empate. El ranking conservador usa $\mu - 3\sigma$. **Estado: implementado y probado** (evaluator/fitness.py::trueskill_ratings, artesanal vía math.erf; validado en vivo: tres victorias movieron un perfil a $\mu \approx 32.4$ vs. 17.6 con σ encogiéndose bajo evidencia). La reputación de grafo (eigenvector/PageRank sobre el grafo de citas de society_facts) es **solo diseño**.

Productores de eventos uno-a-uno. Las comparaciones provienen de (i) el gym determinista (§7.1), implementado; y (ii) un **juez LLM pareado anti-sesgo-de-posición** (§5.5.1), implementado. Los productores de asamblea/consejo son solo diseño.

5.5.1 Juez pareado anti-sesgo-de-posición

El juez compara dos documentos bajo un criterio. Para neutralizar el sesgo de posición evalúa **ambos órdenes** (A,B) y (B,A) y los combina: el acuerdo produce un ganador, el desacuerdo produce un empate honesto. **Estado: implementado y probado** (evaluator/judge.py::build_pairwise_prompt, parse_verdict, combine_orders, verdict_to_match_winner; evaluator/core.py::judge_pairwise corre ambos órdenes y persiste un match source='llm_judge'; tests/test_judge_pairwise.py, 8/8). Un hallazgo empírico documentado de runs internas: **la elección del modelo juez importa** — un juez modelo juez local pequeño y débil exhibe fuerte sesgo de posición, frente al cual el protocolo de dos órdenes degrada con gracia a *empates honestos* en lugar de ganadores espurios; un juez más fuerte, de clase frontera, produce veredictos fiables.

IdentIA: C. *Gobernanza*: L4 (la promoción en el marketplace empieza a usar rating). *Self-trigger* a F2: densidad de la matriz tarea×agente $\geq$ SOCIETY_FITNESS_MATRIX_DENSITY (predeterminado 0.15) y cada perfil activo tiene $\geq$ SOCIETY_FITNESS_MIN_TASKS_PER_AGENT (predeterminado 20) tareas solapadas.

5.6 F2 — habilidad ajustada por dificultad (IRT) — la compuerta de fiabilidad

F2 es la **compuerta de fiabilidad** y una *precondición para cualquier escritura-de-vuelta automática*: separa "el agente mejoró" de "el agente eligió tareas fáciles". Usando un modelo logístico de 2 parámetros de Teoría de Respuesta al Ítem [20] con dificultad de tarea b_k , discriminación a_k , y habilidad latente θ_p :

$$P(\text{profile } p \text{ solves task } k) = \sigma(a_k(\theta_p - b_k)).$$

Las tareas son "ítems", los perfiles son "examinados". El estimador es una pura **base Rasch (1PL) con una mejora opcional a 2PL por flag**, ajustada por Máxima Verosimilitud Conjunta (Newton–Raphson) con un prior gaussiano anti-divergencia, errores estándar de la información de Fisher, e identificabilidad fijada por centrado (evaluator/fitness.py::irt_fit). Está cableada a los datos mediante tres tablas — fitness_responses (la matriz cruda tarea×agente), fitness_items (el catálogo de dificultad), y la columna theta de fitness_scores (method='irt') — y el gym persiste la matriz de respuestas junto con sus matches pareados. Un **detector de task-farming** marca a un agente cuya dificultad media de ítem está muy por debajo de la de la población mientras su tasa de éxito cruda es alta. **Estado: implementado, con pruebas unitarias**

(tests/test_fitness_f2.py, 9/9: recuperación de parámetros sintéticos con correlación de $\theta > 0.85$, ablación de dificultad, anti-farming, no-divergencia, encogimiento de SE bajo evidencia, 2PL), y validado en vivo (§7.2).

IdentIA: B + C (continuidad de la medición; verdad por uso acumulado). *Gobernanza*: L4/L6. *Self-trigger* a F3: $\text{Var}(\theta_p)$ en una ventana móvil $<$ SOCIETY_FITNESS_THETA_NOISE (fitness_theta_stability, con una ventana fitness_theta_history) y el ajuste reporta converged=True (ambas mitades de esa conjunción estaban defectuosas hasta el 2026-09-26 y aquí se describen ya reparadas; §7.3 dice cuáles eran los defectos y qué afirmación del paper costaron) — el sistema solo deja que la fitness actúe una vez que las estimaciones de habilidad dejan de bailar y el modelo realmente ha convergido.

5.7 F3–F5 — actuar, preservar diversidad, y RSI con compuerta

No en este paper. F3 (muestreo de Thompson + enrutamiento NSGA-II) y F4 (preservación de diversidad MAP-Elites) están implementados y con pruebas unitarias pero corren **en sombra, desactivados por defecto** — recomiendan, no asignan, y ningún camino los consulta como precondición—. F5 (auto-edición recursiva) es **solo diseño y con compuerta dura**; sus operadores de variación serían PromptBreeder [16] o EvoPrompt [35], con generación de tareas más difíciles al estilo POET [21]. Su mecánica, la dependencia $\text{enable}(F5) \iff \text{Done}(F2) \wedge \text{Done}(F3) \wedge \text{Done}(F4)$, y el argumento del orden se desarrollan en [PB] §3–§4; la compuerta misma está comprobada por máquina (Apéndice E). Su estado está en §7.4 y en las entradas E-13 y E-14 del ledger; la razón de que ninguno esté activo es la medición de §7.2, no una decisión de política.

5.8 Mecanismos transversales: cuáles obligan y cuáles solo informan

Son transversales por diseño. **No obligan todos por igual, y la diferencia se dice ítem a ítem en vez de insinuarse en el título** — §7.3 y las entradas E-15 y E-30 del ledger registran cómo el primero pasó de informar a bloquear en código, y por qué todavía no ha bloqueado nada.

(i) un **conjunto held-out anti-Goodhart** — *calculado y servido, y desde el 2026-09-03 compuerta del enrutamiento activo* (E-30): cuando el juez LLM contradice al held-out determinista, `fitness_route` en modo *activo* degrada a *shadow*. Sin suficientes comparaciones del juez no bloquea, por diseño, y con el enrutamiento apagado nunca ha actuado en vivo. Es oculto y rotado, puntuado por el evaluador determinista, que ancla al juez LLM antes de F3 (`rotate_holdout`, `holdout_calibration`, `fitness_anti_goodhart`, `GET /fitness/anti-goodhart`; `tests/...`, 12/12; en vivo: p calibrada $\approx +0.316$ con una alarma disparándose en $p \approx -0.316$ sobre una sonda deliberadamente corrompida); (ii) el **trinquete** (principio B): toda mutación es reversible a un previo dominante en fitness (`archive-not-delete`, `superseded_by`); (iii) mecanismos de **honestidad** (predicción-por-pares/suero-bayesiano-de-la-verdad) para la fitness de hechos y de auto-reportes (solo diseño); (iv) **multiobjetivo siempre** (ningún objeto gana ignorando costo/latencia desde F3); (v) una **compuerta de gobernanza por nivel** (enrutamiento L3b/L5, promoción L4, plantilla/SOUL L6, código/migraciones L7/L8) sin saltos.

6. Gobernanza-como-código (L1–L8)

6.1 Niveles

La regla base: *la auto-mejora local de bajo riesgo es automática; los cambios compartidos o de infraestructura necesitan aprobación humana; los cambios de código o de pin del runtime necesitan un PR y pruebas.*

Nivel	Alcance	Modo	Principio
L1	Skills locales	Automático	A
L2a	Memoria entre sesiones	Automático	A
L2b	Identidad de usuario	Automático, acotado	A
L2c	SOUL.md local	Auto; plantilla → gobernanza	A / C
L3a	RAG por perfil	Controlado, presupuestado	A
L3b	Delegación corta	Automático con cuota	A
L4	Marketplace de skills	Cuarentena + promoción por uso	C
L5	Instanciación de perfiles	Auto si hay presupuesto; si no, humano	C
L6	Cambios de plantilla base	El evaluador propone; el humano aprueba	C

Nivel	Alcance	Modo	Principio
L7	Infraestructura (compose, migraciones, política de almacenamiento)	Humano	Metagobernanza
L8	Código y pin del runtime	PR + pruebas	Metagobernanza

Los **presupuestos** por objetivo (`max_profiles`, `max_tasks_running`, `max_runtime_seconds`, `max_delegate_depth`, `max_tokens_estimated`, flags de red/herramienta-privilegiada) ponen compuerta a las acciones; exceder un presupuesto crea un ítem de aprobación. La **cuarentena** aísla skills nuevas, perfiles anómalos, e intentos de cruce de RAG. El servicio `governance` registra las decisiones; no aplica los cambios él mismo — los servicios que actúan deben consultarlo. **Estado: el servicio de gobernanza, las aprobaciones, las verificaciones de presupuesto, y los kill switches (spawn, escrituras de RAG, publicación de skills, auto-aprobaciones del evaluador) están implementados; la compuerta de spawn L5 se exige opt-in (`SOCIETY_GOVERNANCE_ENFORCE`); el cableado de exigencia completa a través de cada vía que actúa es parcial** y es un bloqueador nombrado para la beta cerrada (§7.3).

6.2 Metacentralidad

No en este paper. El esquema de gobernanza fue moldeado por una tesis normativa sobre cómo debería estructurarse el poder dentro de una institución persistente de agentes —que toda arquitectura de poder tiende a **capturarse** o a **disolverse**, y que la respuesta es un centro deliberadamente acotado por auditoría, rotación, transparencia y autolimitación—. Es una posición, no un resultado, y no añade ninguna restricción propia: todo mecanismo que nombra lo aplica —o no— alguno de los niveles de §6.1 o de los peldaños de §5. Mezclarla en un paper que reporta mediciones es precisamente cómo una afirmación normativa adquiere la apariencia de una empírica. Es **[PB] §5**, donde se enuncia con sus límites, incluido el que más importa: todavía no especifica cómo se audita el centro a *sí mismo* más allá de aprobación humana, PR y tests —sin mecanismo de rotación, sin límite de mandato, sin procedimiento por el cual L7–L8 pudieran volverse contra quien los tiene—.

7. Protocolo de evaluación y estado de implementación

Presentamos un *protocolo*, un *resultado en vivo fechado* para la compuerta de fiabilidad, y un estado honesto —no un estudio empírico terminado. Los autodisparadores de la escalera dependen de una densidad de datos que el tráfico natural on-prem tardaría meses en producir; el gym la fabrica a demanda.

Cómo leer esta sección. Cada afirmación de estado está registrada en el **ledger de claims (Apéndice C)** con su fuente, método de verificación, fecha, entorno y límite. Donde una cifra es *heredada* en vez de re-medida para este borrador, el ledger lo dice. Donde una afirmación fue **refutada** por la medición —incluida una garantía de diseño propia (§7.6) y el antiguo resultado principal de este mismo paper (§7.3)— se reporta en vez de retirarse. Un ítem de la suite desplegada, y la forma que comparten todos, está en el Apéndice B.

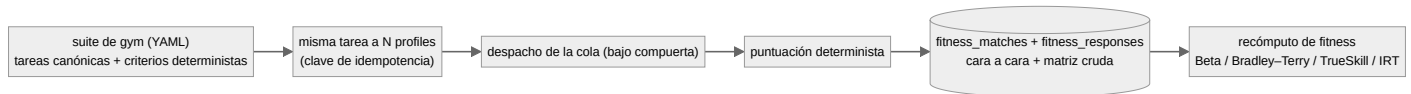
7.1 El gym: dificultad controlada

Una **suite de gym** es una batería YAML versionada de tareas canónicas y autocontenidas, cada una con un nivel de `difficulty` (1 instrucción directa · 2 transformación · 3 síntesis) y criterios de éxito *deterministas* (`required_patterns`, `forbidden_patterns`, `min_length`, `max_length` sobre el texto del resultado — sin juicio

subjetivo). La misma tarea canónica se asigna a varios perfiles; el resultado se puntúa de forma determinista; se registra un **match** cara a cara por cada par sobre la misma tarea (éxito gana a fallo, igual es empate), y la matriz cruda de respuestas se persiste para IRT. Esta es la «dificultad controlada» que exige la regla de coherencia: un cambio que no puede medirse como mejor contra ella es Goodhart.

Estado: implementado y probado. Citas re-verificadas contra el árbol el 2026-08-23:

```
evaluator/gym.py::validate_suite (l. 23), score_gym_output (l. 85), pairwise_matches (l. 139);  
evaluator/tests/test_gym.py; migración 012_fitness_matches.sql; suite gym/suites/canonical-v1.yaml;  
CLI society gym run/score/progress.
```



7.2 Un resultado en vivo fechado para la compuerta de fiabilidad (F2/IRT)

El **2026-06-18** el peldaño IRT se ejercitó de extremo a extremo en un VPS de producción (sistema de ficheros nativo). La matriz del gym se densificó a **6 perfiles en dos niveles de habilidad** (un modelo fuerte y uno local flojo) $\times$ **6 ítems = 36 respuestas**. Se observaron cuatro comprobaciones:

1. **La dificultad se recupera de los datos, no se asume.** La dificultad estimada b_k abarcó $\approx -1,81 \dots +1,73$ y **divergió de los niveles semilla de la suite** — el modelo aprendió la dificultad de las respuestas observadas.
2. **La habilidad se separa y se dispersa.** La habilidad latente abarcó $\theta \approx 0,06 \dots 3,38$ con errores estándar finitos; los niveles se separaron como se esperaba.
3. **El ancla anti-Goodhart se comporta.** La calibración held-out fue positiva ($\rho \approx +0,316$) y saltó una alarma en una sonda negativa deliberadamente corrompida ($\rho \approx -0,316$).
4. **El cultivo de tareas se detecta.** El detector marcó a un granjero sintético (ítems fáciles, tasa cruda inflada) y **no** marcó a la población equilibrada.

Este resultado está fechado. Así consta en el ledger. Su valor es demostrar que el estimador recupera la dificultad y separa la habilidad sobre una matriz pequeña, no establecer una trayectoria poblacional. Las cuatro observaciones de arriba **no** dependen de la bandera de convergencia que las acompañaba, y de esa bandera se ocupa §7.3.

7.3 Una retirada: la negativa que reportamos era un artefacto de nuestro propio criterio

La misma corrida reportó **converged=False**, y versiones anteriores de este paper construyeron un resultado sobre ello. Lo retiramos aquí, decimos qué ocupó su lugar, y conservamos el episodio en vez de borrarlo, porque **cómo** se encontró el error es la parte que vale la pena leer.

Qué afirmábamos. Que la matriz $N = 6$ contenía *patrones de respuesta perfectos* que JMLE no puede ajustar a un óptimo finito; que el autodisparador retenía por tanto la activación de F3; y que un sistema que mide la fiabilidad de su propia señal de fitness y declina actuar sobre una no convergida estaba ejerciendo el orden de seguridad de [PB] §4 en código, no afirmándolo en prosa.

Qué encontró la nueva medición (2026-09-26). Tres defectos, y cada uno por separado invalida la lectura:

1. **La bandera no podía valer `True`**. La prueba de convergencia medía el cambio por iteración *antes* del paso de identificabilidad que centra las dificultades de los ítems, y ese paso deshace en parte el cambio que se estaba midiendo. Con las dificultades empujadas en la misma dirección —que es lo que ocurre cuando casi todos resuelven casi todo— el paso medido se quedó cerca de 0,27 en cada iteración, mientras θ y b se movían menos de 10^{-15} entre 50 y 5.000 iteraciones. El ajuste alcanzaba un óptimo finito; el criterio no podía decirlo. **La explicación que dábamos era por tanto también equivocada**: el obstáculo no eran patrones de respuesta perfectos derrotando a JMLE, era nuestra propia regla de parada.
2. **La cita no sostenía la afirmación**. «F3 la consulta y declina actuar» estaba referenciado a números de línea del módulo puro de fitness, que calcula la bandera y la devuelve en los metadatos del ajuste. Ese módulo no tiene base de datos, ni red, ni consumidor; no puede gatear nada. Que no hubiera consumidor no es una suposición: una auditoría interna del 2026-08-31 barrió todo el árbol fuera de las pruebas y de ese módulo y no encontró ninguno, y la migración que más tarde añadió uno dice lo mismo en su propia cabecera (E-27).
3. **El disparador que sí controlaba F3 se abría solo**. La ventana de estabilidad de θ contaba *recálculos* en vez de estados distintos de los datos. Con los datos sin cambiar, recálculos idénticos repetidos producían varianza cero y satisfacían el disparador.

Qué sobrevive y qué no. Las cuatro observaciones de §7.2 sobreviven: son propiedades de las estimaciones, no de la bandera. Lo que no sobrevive es la afirmación empírica principal de este paper. **No tenemos ningún resultado en vivo que establezca que la compuerta retiene la activación**, y retiramos la afirmación de que ningún sistema comparable reporta esa propiedad, puesto que nosotros mismos no la habíamos demostrado.

Dónde está la escalera ahora. Tras el arreglo, un ajuste sobre 720 respuestas (20 perfiles $\times$ 12 ítems) reporta `converged=True` en 14 iteraciones sin añadir datos (E-28). Esa medición es de un entorno de desarrollo, no de una instalación de cliente, y es una corrida. F3 sigue inactivo **por decisión, no por compuerta** — una distinción que la versión anterior negaba expresamente, y que ahora enunciamos al revés.

Qué se construyó desde entonces, y es justo lo que habíamos afirmado. El mismo día del arreglo la bandera dejó de ser efímera: el meta de cada ajuste se guarda, una fila por recálculo (migración `039_fitness_irt_runs`), y `fitness_route` en modo *activo* lee la última fila y vuelve a shadow con `gated_reason="irt_not_converged"` en tres casos — el ajuste no convergió, no hay ninguna fila, o no se puede leer. **Falla cerrado**, y ocho pruebas fallan sin él (E-29). Así que la frase que publicamos antes de que existiera su referente ya tiene uno. Lo reportamos como propiedad del árbol, medida por pruebas: **ninguna corrida en vivo lo ha ejercitado**, y reportarlo como si alguna lo hubiera hecho es el error que esta sección existe para retirar.

Por qué esto está en el paper. Un paper cuyo método es *no afirmar lo no medido* tiene que sobrevivir a su propio método. El ledger registró esta cifra con su fecha, su fuente y el límite *no se ha vuelto a medir desde entonces*; tres meses después, ese límite es lo que provocó la medición que destruyó la afirmación. La contabilidad funcionó y la afirmación no, y reportar solo lo primero sería exactamente el fallo que el ancla anti-Goodhart existe para impedir.

Una segunda debilidad, no relacionada, en la misma zona, reportada en vez de omitida: en el corte del 2026-08-23 el guardarraíl anti-Goodhart se *calculaba* y se *servía* pero **no bloqueaba** (E-15): se definía, se importaba y se exponía por un endpoint de lectura, sin que ningún camino lo consultara como precondition, bajo un título de la documentación interna que decía «transversales, no opcionales». Desde el 2026-09-03 es

precondición del enrutamiento activo (E-30). Todavía no ha bloqueado nada: el enrutamiento está apagado por decisión, y el ancla se compara con rankings del juez LLM, de los que el entorno de desarrollo no tenía ninguno el 2026-10-01, así que por diseño no puede dar la alarma. Las afirmaciones de §5 sobre él describen una compuerta que existe en código y no se ha ejercitado.

7.4 Hipótesis, métricas, y qué está implementado frente a diseñado

Hipótesis. H1 (calibración): los rankings Bradley–Terry/TrueSkill alimentados por el gym concuerdan con el orden de verdad-terreno determinista bajo dificultad controlada. H2 (disciplina de incertidumbre): el ranking por cota inferior de Wilson evita la dominancia en arranque frío. H3 (robustez del juez): el juez de dos órdenes reduce el error por sesgo de posición y degrada a empates con jueces débiles. H4 (compuerta de fiabilidad): bajo IRT, los perfiles no pueden subir de rango seleccionando ítems fáciles; el detector de cultivo de tareas dispara (observado, §7.2). Métricas: correlación de rangos frente a verdad-terreno; tasa de inversión en arranque frío; tasa de desacuerdo por sesgo de posición; correlación de recuperación de θ ; latencia del pipeline; tasa de PASS del auditor; costo por tarea; cobertura del grafo. Líneas base: enrutamiento round-robin; juicio de un solo orden; ranking por media (no Wilson).

Componente	Estado	Ledger
F0 posteriores Beta + ranking Wilson	✓ implementado y probado	E-11
Outcome canónico de tarea (definición compartida)	✓ implementado y probado	E-11
Bradley–Terry (MM/Hunter, sin dependencias)	✓ implementado y probado	E-11
TrueSkill (en línea, <code>math.erf</code>)	✓ implementado y probado	E-11
Juez pairwise anti-sesgo de posición	✓ implementado y probado	E-11
Gym (validar/puntuar/pairwise + persistir matriz)	✓ implementado y probado	E-10
Comprobaciones deterministas del evaluator (artifact, salud, outcome, sello de borrador)	✓ implementado y probado	E-12
F2 IRT (1PL/2PL, JMLE) + estabilidad de θ + cultivo de tareas + ancla held-out	● implementado y probado; resultado en vivo 2026-06-18 retirado (§7.3); no queda en pie ninguna decisión de la compuerta en vivo	E-13
F3 Thompson + NSGA-II + cobertura QD	● implementado y probado; sombra / apagado por defecto	E-14
F4 archivo MAP-Elites + protección de nicho + meseta	● implementado y probado; sombra / apagado por defecto	E-14
Ancla anti-Goodhart held-out	● se calcula y se sirve; compuerta del enrutamiento activo desde el 2026-09-03, nunca ejercitada en vivo (§7.3)	E-15, E-30
Envelopes de federación + políticas de confianza	● implementado; estructurados y auditados, no firmados criptográficamente	E-08
Transporte entre hosts (A2A v1.0)	✓ habilitado y ejercitado con controles negativos	E-07
Vocabulario del grafo cerrado + guardián en la compilación	✓ implementado; el guardián falla ante tipos no declarados	E-09

Componente	Estado	Ledger
Formas SHACL	● declaradas; el fichero afirma que no se ejecutan en la compilación	E-09
Ligadura de identidad en la aprobación	● el campo de decisión es texto de la petición ; no ligado a un principal verificado	E-16
Integridad del registro de auditoría	● no encadenado por hash ; algunas escrituras son fail-open deliberadamente	E-16
Reputación de grafo (autovector sobre facts)	📋 solo diseño	—
Lado de lectura del contexto episódico	📋 solo diseño (lado de escritura implementado)	—
F5 auto-edición recursiva	📋 solo diseño (con compuerta dura)	—
Transmisión gobernada de memoria entre profiles	📋 solo diseño	—

Contados. De los 21 componentes de la tabla: **9 están implementados y probados**; **6 están implementados pero no obligan a nada** — computan en sombra, se sirven sin bloquear, o están declarados en un fichero que dice que no se ejecuta; **2 están implementados con un defecto nombrado** que deja indisponible la garantía que sugieren; y **4 son solo diseño**. La proporción importa más que cualquier fila suelta: **12 de 21 no restringen nada hoy**, y el bucle cerrado variación → selección → mutación necesita elementos de dos de esos grupos —F3 actuando en vez de en sombra, y F5 existiendo siquiera—, así que no puede correr hoy por construcción, no por configuración. El almacenamiento de F0–F2 es el esquema de cuatro tablas de las migraciones 011–014 (E-25); las tablas se nombran allí, no se reproducen aquí.

7.5 Interoperabilidad: una medición de conformidad contra una suite externa

Las afirmaciones de interoperabilidad suelen enunciarse. Nosotros medimos una.

El Unified Harness Protocol (UHP) publica una especificación versionada con una suite de conformidad ejecutable. Ejecutada contra nuestro profile API (clase `core`) el 2026-08-23 (E-17), el resultado fue **1 de 37 comprobaciones pasando: 1 pasa · 8 fallan · 24 se saltan · 4 error**. Los 24 saltos son una cascada de un solo punto — sin endpoint de catálogo no hay nada más que ejercitar. El hueco son cuatro elementos concretos: un documento de descubrimiento, una cabecera de versión (y rechazar con error una versión no soportada en vez de sustituirla en silencio), y dos endpoints de catálogo cuyo contenido ya existe con otros nombres. Lo único que hoy pasamos es rechazar una credencial ausente.

La forma conocida de cerrar el hueco de auditoría. Donde §7.4 reporta que el registro de auditoría no está encadenado por hash, el remedio no es exótico: encadenar cada anotación con la anterior, de modo que suprimir o alterar una rompa la cadena de forma detectable, y acompañarla de una atestación firmada de a quién pertenece cada acción. Existen implementaciones abiertas de ambas piezas. **Enunciamos la dirección, no un compromiso de calendario ni de proveedor.**

Dos límites adyacentes, nombrados porque la corrida de conformidad los sacó a la luz. Primero, el transporte entre hosts autentica con tokens bearer planos por par llamante: quien llama no puede delegar una autoridad *estrechada* —una tarea, con caducidad— sin volver al emisor. Existen esquemas de token con

atenuación offline que resuelven exactamente esto; adoptar el contrato, no la malla, es la dirección que tomaríamos. Segundo, nuestros servidores de herramientas son stdio-only, alcanzables solo por el proceso del agente que los lanza. Por eso el servidor de skills no puede portar credencial propia, limitación que tenemos documentada internamente como deuda y no como diseño. Un transporte HTTP/SSE para el mismo protocolo la elimina.

El hallazgo que merece reportarse no es la nota. La especificación exige que el descubrimiento se sirva **sin autenticación**. Nuestros servicios internos son fail-closed por diseño — una postura adoptada tras dos fallos silenciosos de autorización en producción. Servir un endpoint anónimo en la red interna es por tanto una **decisión de seguridad, no un trámite de conformidad**, y la declinamos salvo que aparezca un cliente externo que la exija. Reportamos el 1/37 en vez de los tres elementos que pensamos cerrar, porque una nota de conformidad citada tras remediación selectiva no es una medición.

7.6 Una garantía de diseño refutada, reportada

Nuestra propia documentación afirmaba, en dos sitios —el manifiesto de despliegue y el docstring del módulo del servicio—, que un único servicio bajo compuerta tiene autoridad sobre el runtime de contenedores, y que el agente planificador no puede por tanto aprovisionar sin aprobación humana. **La medición refuta la premisa (E-23).** El socket del runtime se monta en **tres** servicios, no en uno; el host de agentes está entre ellos, y la cuenta que corre dentro tiene escalada de privilegios sin contraseña.

Un detalle que vuelve el hallazgo más agudo, no más leve: la configuración con la que se entrega el host de agentes selecciona un backend de terminal **local**, no uno de contenedor. El socket no lo exige, por tanto, el modo de ejecución actual del runtime: se monta por defecto y se ajusta al arranque tras un interruptor opcional, para un camino de sandbox en contenedor que la configuración por defecto no usa. **Un privilegio montado por defecto para un modo que no está activo es superficie sin propósito corriente**, que es una justificación más débil que la que la afirmación original daba a entender.

Dos salvedades, ambas necesarias. Primera: esto **refuta la garantía escrita; no demuestra una explotación** — los permisos efectivos dependen del despliegue y no intentamos ninguna. Segunda: la compuerta de *aprobación* es independiente del socket — el camino de despliegue exige una especificación aprobada, y ese requisito sí se aplica en código.

Lo incluimos porque la alternativa —corregir la frase en silencio— dejaría al lector sin poder distinguir un sistema cuyas fronteras se verificaron de uno cuyas fronteras solo se enunciaron. **Un sistema que reporta la refutación de su propia garantía está haciendo una afirmación más fuerte sobre su método que uno que reporta solo éxitos.**

7.7 Estado del proyecto

Todas las cifras siguientes están medidas contra el árbol consolidado del repositorio del **2026-08-23** (commit 90792f5) y cada una lleva su fila en el Apéndice C; una cifra sin fila sería un defecto por la propia regla del ledger.

El sistema corre **15 servicios** (E-01) con **26 migraciones de esquema** (000–025) aplicadas (E-02), sobre un runtime fijado a **Agents v0.20.0** (SHA 9d4ef04ed ; E-03). Dos instalaciones de cliente operan sobre hardware dedicado, cada una un despliegue separado con su propio perímetro; una de ellas corre una unidad de operaciones de ocho agentes con un canal de mensajería vivo (E-04). La **puerta de beta cerrada se declaró cumplida el 2026-07-21**, con una persona externa completando la incorporación el mismo día (E-05).

Postura de seguridad, ya que v3 la reportaba como bloqueante: la autenticación está activa y los servicios internos exigen credencial en toda mutación (2026-08-08; E-06); las credenciales de modelo por usuario se cifran con una clave por instalación (2026-08-09); la salida de red está acotada **por departamento**, es decir por contenedor y no por agente (2026-08-09; E-19); hay respaldos diarios verificados en las dos instalaciones, y el primer ensayo de restauración **falló**, destapando cuatro defectos que solo la ejecución revela (2026-08-19; E-20). La reproducibilidad del propio despliegue es desigual: trece imágenes de contenedor siguen una etiqueta móvil y una está fijada por digest (E-24).

Dos capacidades ausentes por completo en v3: una **ontología con vocabulario cerrado** cuyo guardián falla la compilación cuando el código emite un tipo de nodo o arista no declarado, y una **capa de grafo y razonamiento** integrada por debajo de la consola con identidad por persona en el borde (E-18). Ambas constan en el ledger con sus límites, de los cuales el material es que la capa de grafo **no tiene aislamiento por inquilino** y está por ello restringida a un solo permiso.

La batería de pruebas comprende **143 ficheros de test en Python que definen 1.212 funciones de test**, más **34 ficheros de test de front-end** (E-21). Una ejecución completa reportada por una revisión independiente el 2026-08-22 contó **1.532 pasando**; esa cifra consta en el ledger como *heredada, no re-ejecutada para este borrador* (E-22), ya que una suite parametrizada produce más casos que el conteo estático de definiciones.

7.8 Amenazas a la validez de constructo específicas de SIS

El gym mide *comparabilidad entre perfiles de la misma instalación bajo dificultad controlada*, no calidad absoluta; no debe leerse como un benchmark público. Los criterios deterministas premian la corrección detectable por patrón y pueden pasar por alto cualidades que solo un juez fuerte captaría — de ahí el ancla held-out, con la salvedad de §7.3 de que el ancla no bloquea hoy. El resultado en vivo de F2 (§7.2) es una *única corrida fechada* sobre una matriz pequeña; no se afirma ninguna trayectoria poblacional.

Dos límites más, específicos de este borrador. El resultado de conformidad de §7.5 mide la clase `core` de un protocolo contra un endpoint, y no generaliza a nada más. Y la postura de seguridad reportada en §7.7 se lee de configuración y código, no de una sonda en vivo sobre una instalación corriendo; §7.6 dice explícitamente dónde esa distinción importa.

8. Amenazas a la validez

8.0 Una mitad del patrón, y quién está construyendo la otra

El identificador de actor de este sistema adopta la forma de miembro de X-Road —instancia, clase, miembro, subsistema—, una capa de intercambio entre organizaciones con más de veinte años en producción nacional. La adopción es real y está en el código, pero es **parcial de una manera que conviene declarar**: se tomó la mitad

del **transporte** y se saltó la del **catálogo**. En el patrón original existe un registro central que dice qué organizaciones y qué servicios son legítimos, firmado y distribuido a los participantes. Aquí ese registro existe como tabla en una base de datos que solo se consulta localmente. **Un directorio que no se firma ni se distribuye no es un catálogo federado; es una lista.**

Esto no es una omisión abstracta. El 16 de junio de 2026 Estonia —la jurisdicción que originó X-Road— aprobó emitir códigos de identidad propios a agentes autónomos, distintos de su principal humano, acotados y auditables a través de esa misma capa, y encargó a su agencia estatal la construcción de **un registro central de confianza para agentes** de los sectores público y privado, con veinticuatro meses y un millón de euros de presupuesto.

Ese hecho corta en dos direcciones y las dos importan. **Valida** la decisión de dar al agente una identidad con forma de miembro de una capa de interoperabilidad pública, tomada aquí de forma independiente diez días antes. Y **advierte**: la pieza que este sistema declara ausente es exactamente la que una jurisdicción va a construir con presupuesto y calendario. Interoperar con ella será una condición, no una opción — y llegar con la mitad del patrón implementada es una posición mejor que llegar sin ninguna, pero peor que la de quien lo construye.

Internas. Los self-triggers de fitness dependen de umbrales (`SOCIETY_FITNESS_*`) cuyos valores por defecto son conjeturas de ingeniería; umbrales mal puestos podrían avanzar un peldaño prematuramente. La mitigación está en capas: el avance es *solo-lectura/shadow* hasta F3-activo, F3-activo es fail-open (degrada a shadow cuando θ es inestable o el ajuste no converge), y F5 tiene compuerta dura sobre F2–F4. **Antes citábamos el bloqueo en vivo `converged=False` como evidencia de que la mitigación se acciona en la práctica; §7.3 lo retira.** El escalonado es una propiedad de diseño que no hemos demostrado en operación, y de una de sus capas —el disparador de estabilidad de θ — se encontró que se satisfacía a sí misma con datos sin cambiar.

Externas. La topología de referencia (una demo "MAP" de 14–16 unidades) es un *dataset*, no una afirmación sobre ninguna institución real; todas las etiquetas provienen de YAML/spec. Los resultados en tareas canónicas del gym pueden no transferirse al trabajo institucional abierto.

De constructo. El "mejor-que" según lo adjudicado por un juez LLM puede codificar los sesgos del juez; el protocolo de dos órdenes, un ancla held-out determinista, y la sensibilidad documentada a la elección del juez (§5.5.1) acotan pero no eliminan esto. "La identidad persiste bajo cambio de modelo" es una garantía de *diseño* (Ψ preservada), no una afirmación empírica sobre invariancia conductual — un modelo intercambiado se comportará de forma diferente aun con SOUL/memoria idénticos; la afirmación de IdentIA es sobre *identidad*, no sobre *equivalencia conductual*, una distinción que hacemos explícita para evitar el exceso y que el §D.2 enuncia como ecuación: $\$b_t = f(\Psi, m_t) + \text{\textbackslash varepsilon_t}$ — intercambiar la medición cambia tanto el segundo argumento del mapa como la distribución del residuo, fuera del control de Ψ .

Reproducibilidad. Varias afirmaciones descansan en runs internas (p.ej. los hallazgos de sesgo-de-posición del juez, el único run en vivo de F2); las reportamos como observaciones, no como experimentos controlados, y el gym se provee precisamente para hacerlas reproducibles.

9. Discusión

9.1 Interpretación

La apuesta de SIS es que *persistencia más selección medida más gobernanza* es un sustrato mejor para organizaciones de agentes confiables que la orquestación sola. La escalera de selección es la respuesta del sistema a un modo de fallo recurrente de los sistemas auto-mejorantes: recompensar lo que se *declara* mejor en lugar de lo que se *mide* mejor. Al hacer del *orden* de las capacidades el argumento de seguridad — fiabilidad y diversidad antes de la recursión — SIS convierte "ten cuidado con la RSI" de un eslogan en un grafo de dependencias. Si ese grafo es *portante* en producción y no solo en las pruebas es algo que hoy no podemos afirmar: el bloqueo en vivo que ofrecíamos como evidencia queda retirado (§7.3), y F3 está inactivo por decisión y no por una compuerta medida.

9.2 Limitaciones

Las limitaciones más consecuentes son: (i) el peldaño superior (F5) no está construido y los peldaños de lazo cerrado (F3–F4) corren solo en shadow, por lo que la afirmación central de "auto-mejora" es, hoy, una *capa de selección fiable con una recursión con compuerta, aún no activa* — la mitad de selección está medida y parcialmente en vivo, la mitad recursiva está especificada; (ii) **ninguna corrida en vivo establece que la compuerta de fiabilidad retenga la activación** — la que reportábamos era un artefacto de nuestro propio criterio de convergencia y queda retirada (§7.3), así que el comportamiento de la compuerta ante una matriz real está sin probar; (iii) la federación está auditada pero no firmada criptográficamente, por lo que la confianza entre hosts descansa actualmente en política y fronteras de red, no en criptografía; (iv) el multi-tenancy está incompleto, acotando el despliegue externo, y el backend de auth está validado solo en staging; (v) el grado emergente del puente $A \rightarrow C$ y el lado de lectura del contexto episódico están deliberadamente diferidos. Lo declaramos llanamente porque el valor del paper depende de que sus afirmaciones sean verificables.

9.3 Ética y riesgo

Una plataforma que simula ministerios y empresas, acumula memoria institucional, y (eventualmente) edita su propio andamiaje plantea preocupaciones reales: concentración de poder, dependencia de proveedores de LLM, datos sensibles en el RAG por agente, y la naturaleza de doble uso de la auto-mejora recursiva. Las mitigaciones de SIS son estructurales — gobernanza L1–L8, presupuestos por objetivo, cuarentena, archivar-no-borrar, compuertas humanas sobre la transmisión cultural y sobre la RSI, una pausa/rollback global de RSI a un snapshot dominante en fitness, y la disciplina de metacentralidad de auditoría/rotación/transparencia/autolimitación — esta última, una **postura enunciada, no un mecanismo aplicado** ([PB] §5)—. Ninguna de estas es suficiente por sí sola; las presentamos como una postura de defensa-en-profundidad e invitamos al escrutinio adversarial, especialmente de las compuertas de F5 y del modelo de confianza de federación en ausencia de firma de envelopes.

10. Conclusión

Self-Improving Society reencuadra la unidad del cómputo agéntico del *proceso* a la *institución*. Este paper reporta la parte de ese reencuadre que está construida y medida: **IdentIA** como ontología de identidad cuyos principios se aplican como invariantes de código en vez de afirmarse en prosa; una **arquitectura** cuya unidad de despliegue es una unidad organizativa soberana envuelta por un exoharness que gobierna un runtime intercambiable en vez de reimplementarlo; tres peldaños implementados de una escalera de selección medida que terminan en una **compuerta de fiabilidad** por Teoría de Respuesta al Ítem; y **gobernanza-como-código** con un ledger de claims que hace visible en un solo sitio la deriva del estado.

El resultado empírico que este paper reportaba queda retirado (§7.3). Habíamos leído la corrida en vivo del 2026-06-18 como la compuerta declinando actuar sobre su propia medición poco fiable; al volver a medir, la negativa resultó ser una propiedad de nuestro criterio de convergencia, que no podía devolver `True` con ningún dato. Lo que queda es más estrecho y vale la pena decirlo llanamente: el estimador recupera la dificultad y separa la habilidad sobre una matriz pequeña, los componentes están contruidos, y **ninguna corrida en vivo muestra todavía a la compuerta reteniendo nada**. F3 está apagado por decisión, no por medición.

El episodio es el método del paper aplicado al paper. Una cifra se registró con su fecha, su fuente y el límite *no se ha vuelto a medir desde entonces*; ese límite es lo que produjo la nueva medición tres meses después, y la nueva medición destruyó la afirmación. Preferimos publicar un sistema de contabilidad que pilla a su autor antes que un resultado que sobrevive porque nadie volvió a comprobarlo.

Lo que este paper no hace es argumentar que el *orden* de la escalera sea la propiedad de seguridad. Ese argumento, los peldaños superiores que le conciernen, y la tesis normativa sobre el poder institucional son el paper de posición compañero **[PB]** — separados porque nada de eso ha corrido, y un paper de sistemas que carga un argumento no probado junto a un resultado medido le presta al primero la credibilidad del segundo.

Trabajo futuro concreto, en orden de dependencia y no como lista de tareas paralelas: volver a correr el peldaño IRT en una instalación de cliente y no en una de desarrollo, para que exista un ajuste convergido fuera del entorno donde se reparó el criterio; solo entonces activar F3, y F4 después; añadir firma criptográfica de sobres; ligar las decisiones de aprobación a un principal verificado y encadenar por hash el log de auditoría (§7.4, E-16); y correr un estudio longitudinal de la trayectoria de fitness de una sociedad federada.