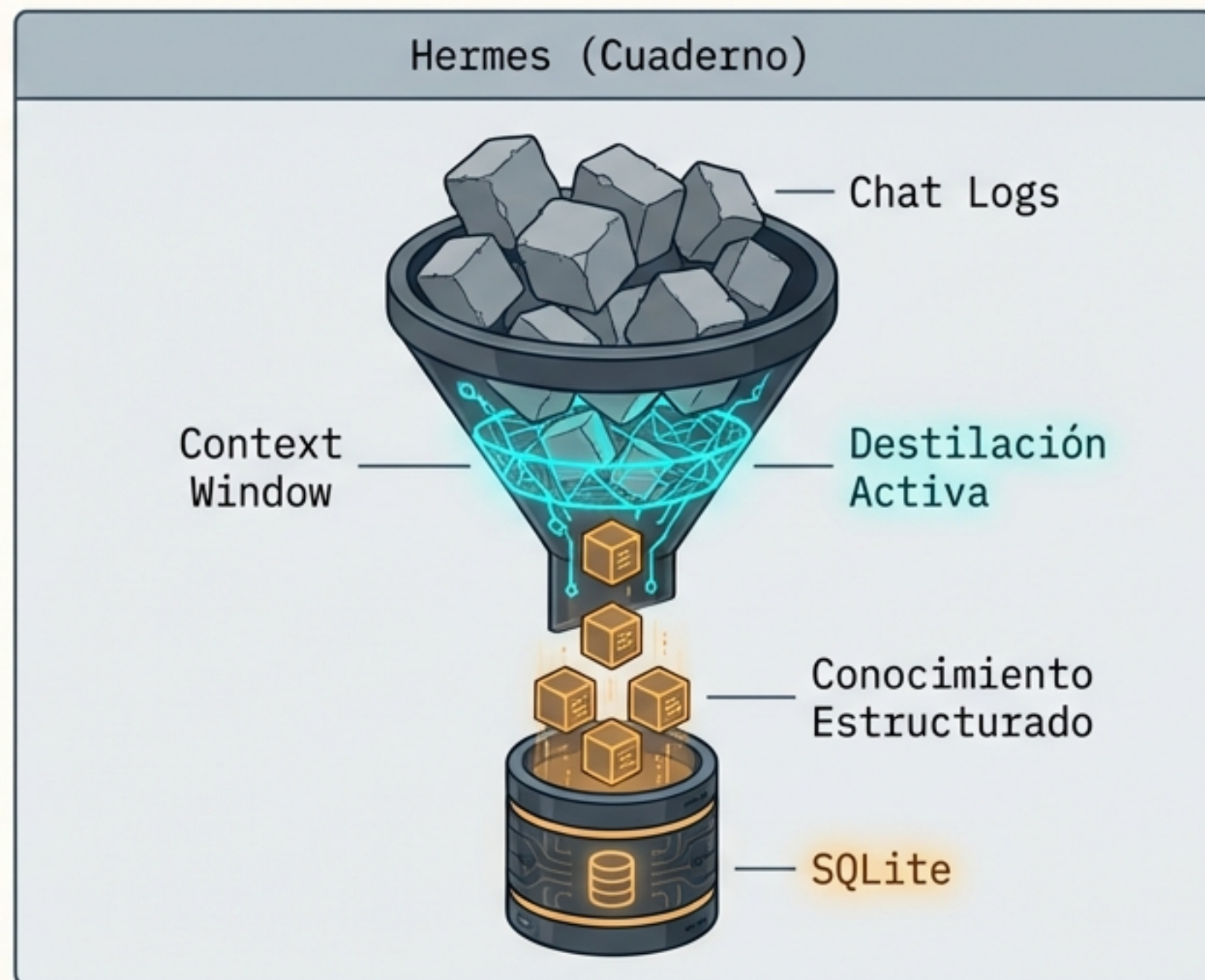
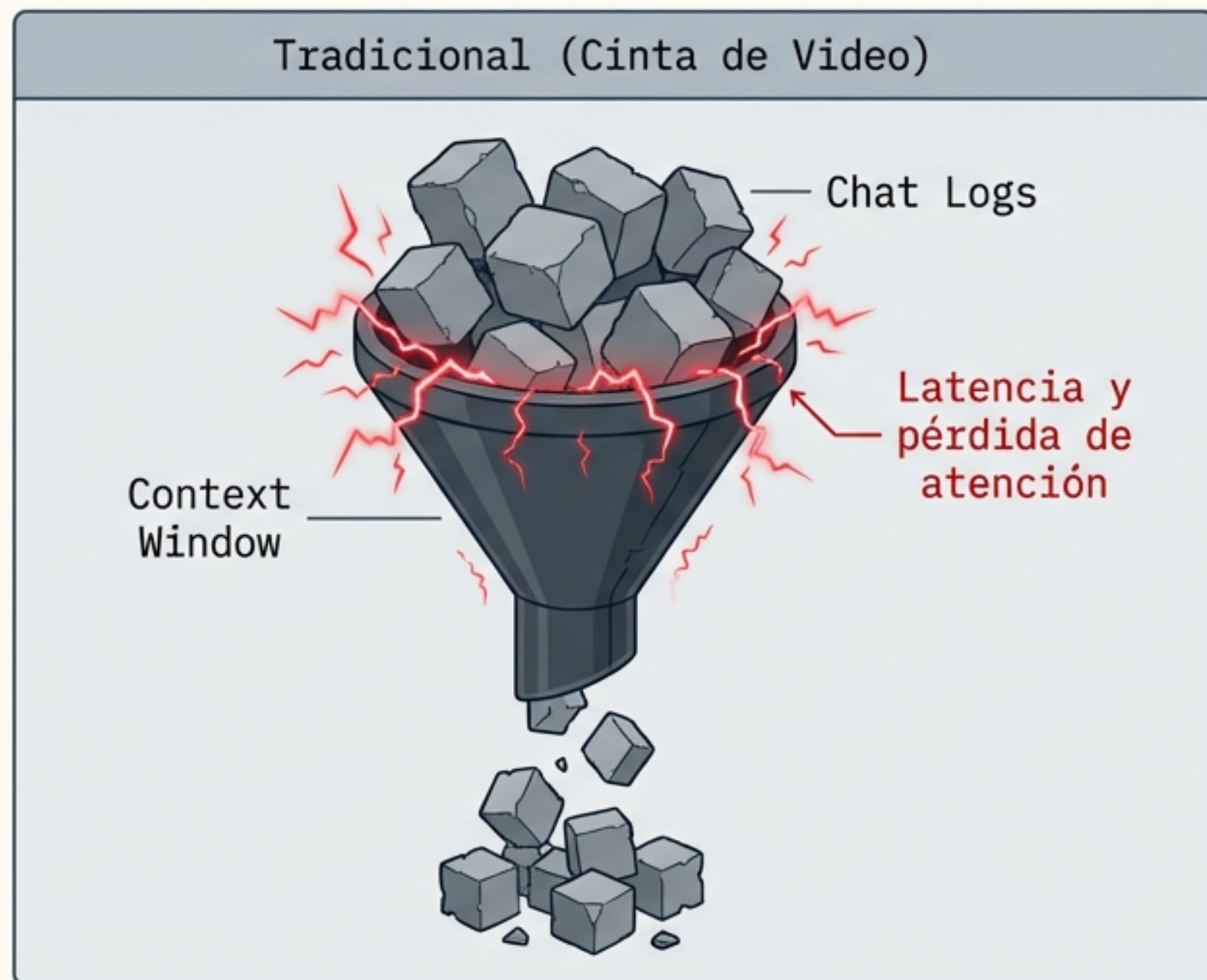


Hermes Agent: Arquitectura del Bucle de Aprendizaje Continuo

Diseño de sistemas multi-agente con auto-optimización, memoria estructurada (FTS5) y modelado inferencial.

El Cambio de Paradigma: De Memoria Pasiva a Destilación Activa

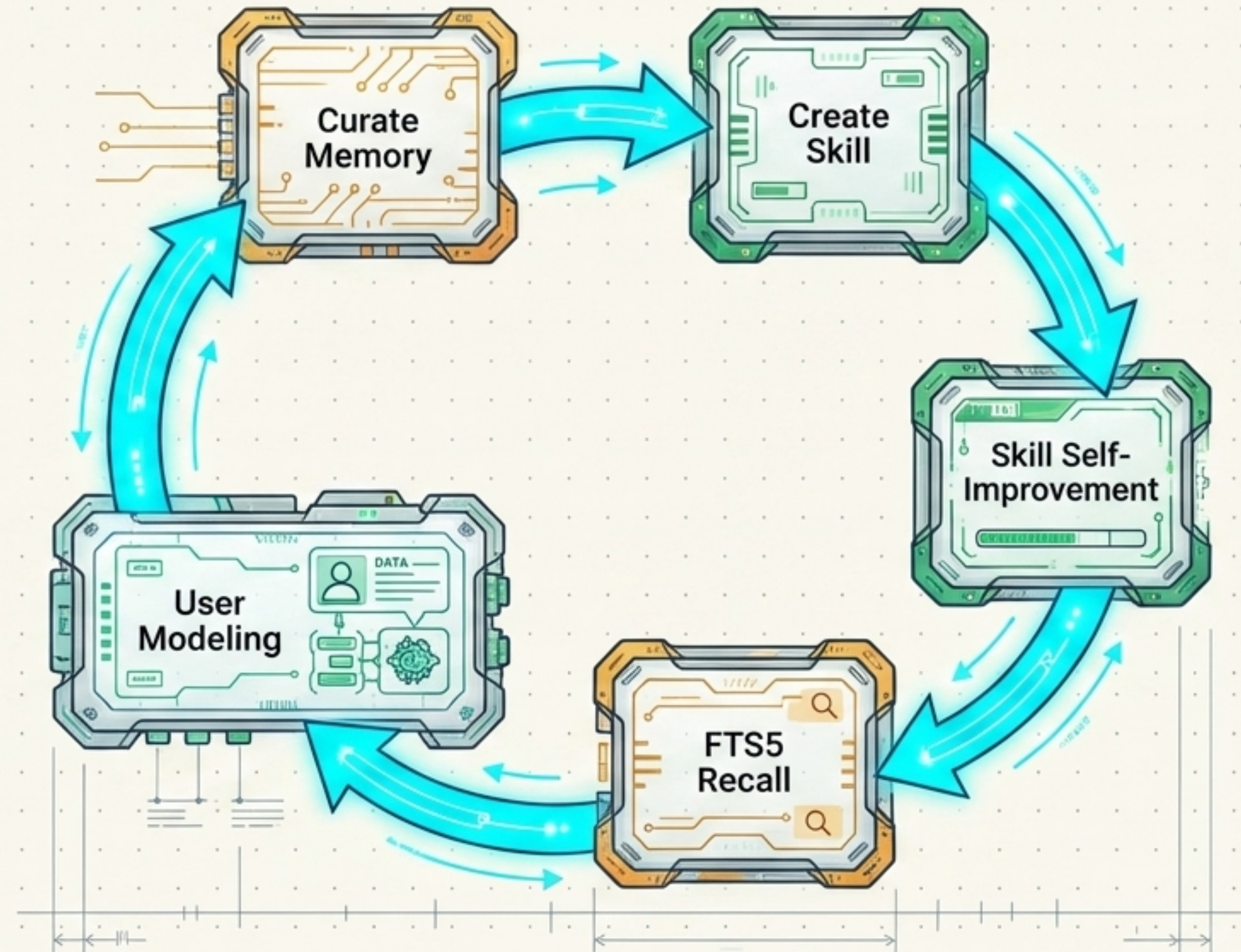
El problema del estado en LLMs no es la retención, es la curaduría. Las herramientas tradicionales acumulan logs hasta desbordarse. Hermes reemplaza el volcado bruto de contexto por una ingesta selectiva y estructurada.



Arquitectura del "Learning Loop" (Visión Macro)

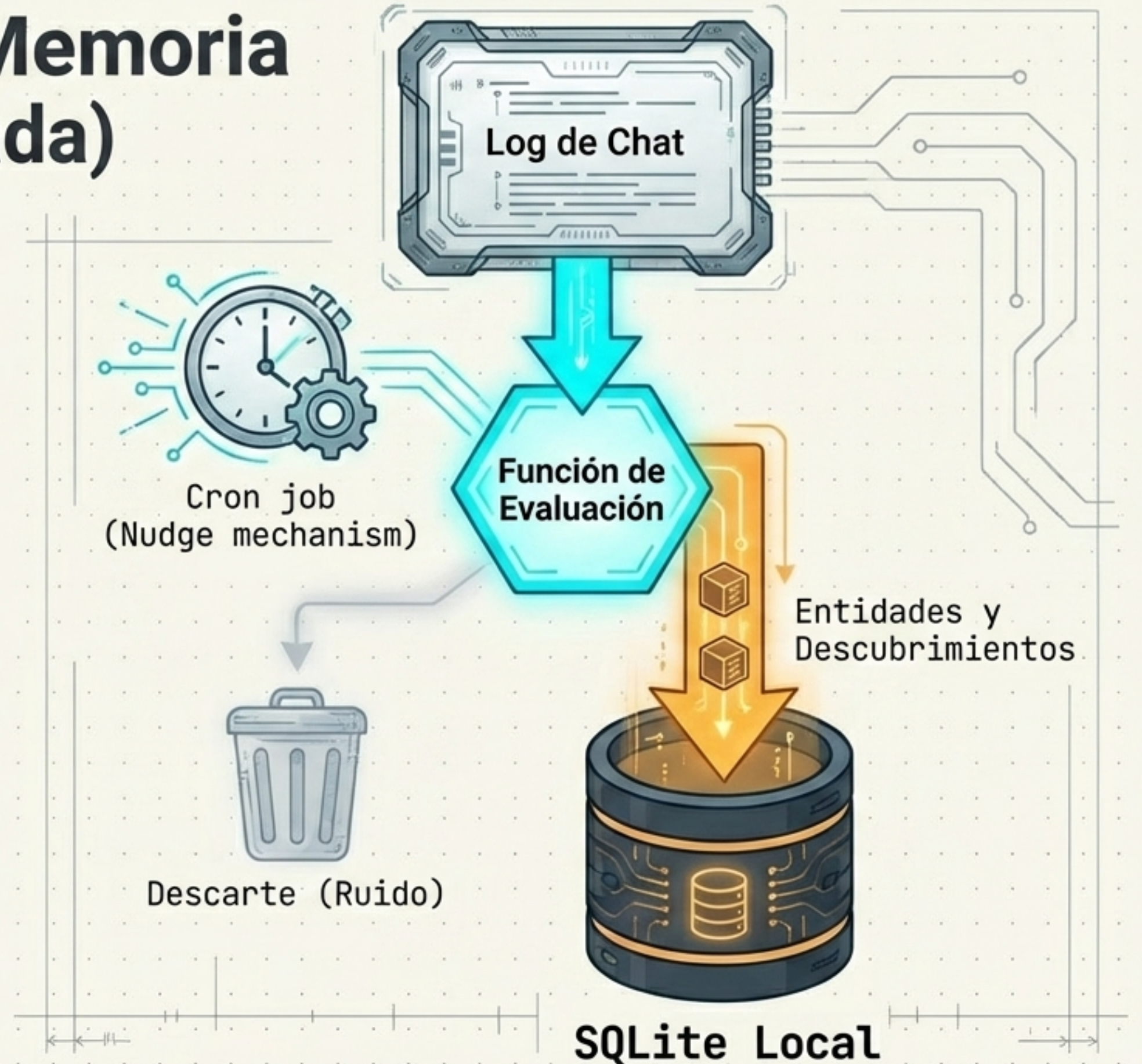
Un mecanismo de auto-ensamblaje (Self-Harnessing) basado en un ciclo cerrado de 5 fases.

Nadie le enseña al agente que prefiere `httpx` sobre `requests`, o que organizas bajo un directorio `src/`. Lo infiere y se reconfigura de forma autónoma.



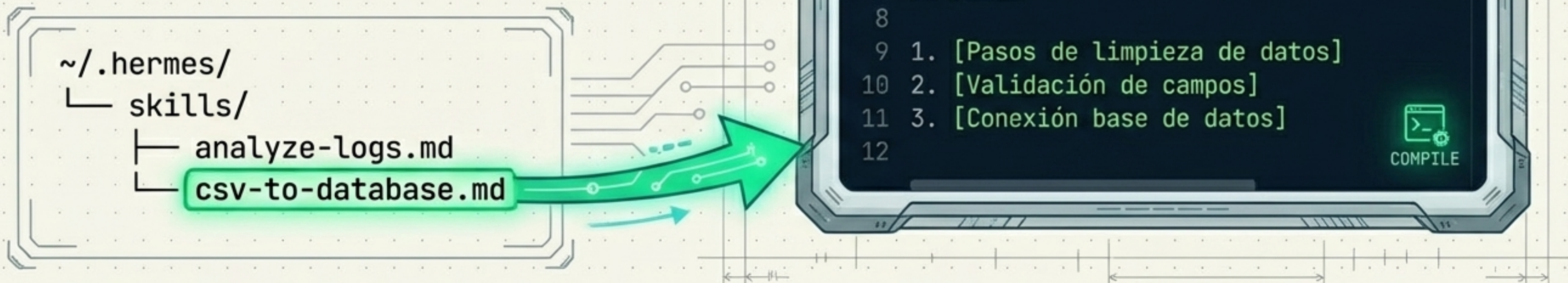
Fase 1 – Curaduría de Memoria (Persistencia Estructurada)

Evaluación post-ejecución. El agente ejecuta un nudge mechanism asíncrono para decidir qué deltas de información merecen persistencia. Decide activamente qué guardar en SQLite, eliminando latencia de red y dependencias de nube.



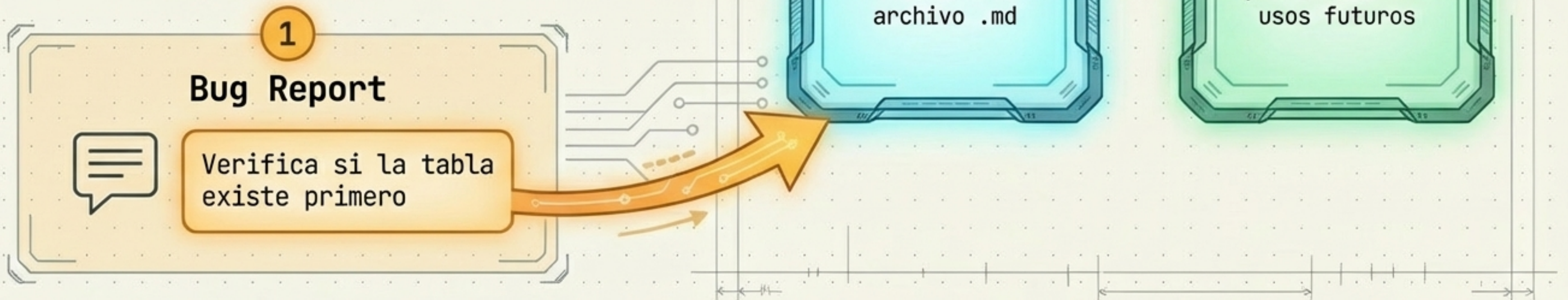
Fase 2 – Creación Autónoma de Skills (Modularidad)

Destilación de rutinas complejas en artefactos reutilizables. Si una solución validada tiene utilidad futura (ej. limpiar un CSV), se compila y almacena como un Skill global en disco. Cero inicios desde cero.



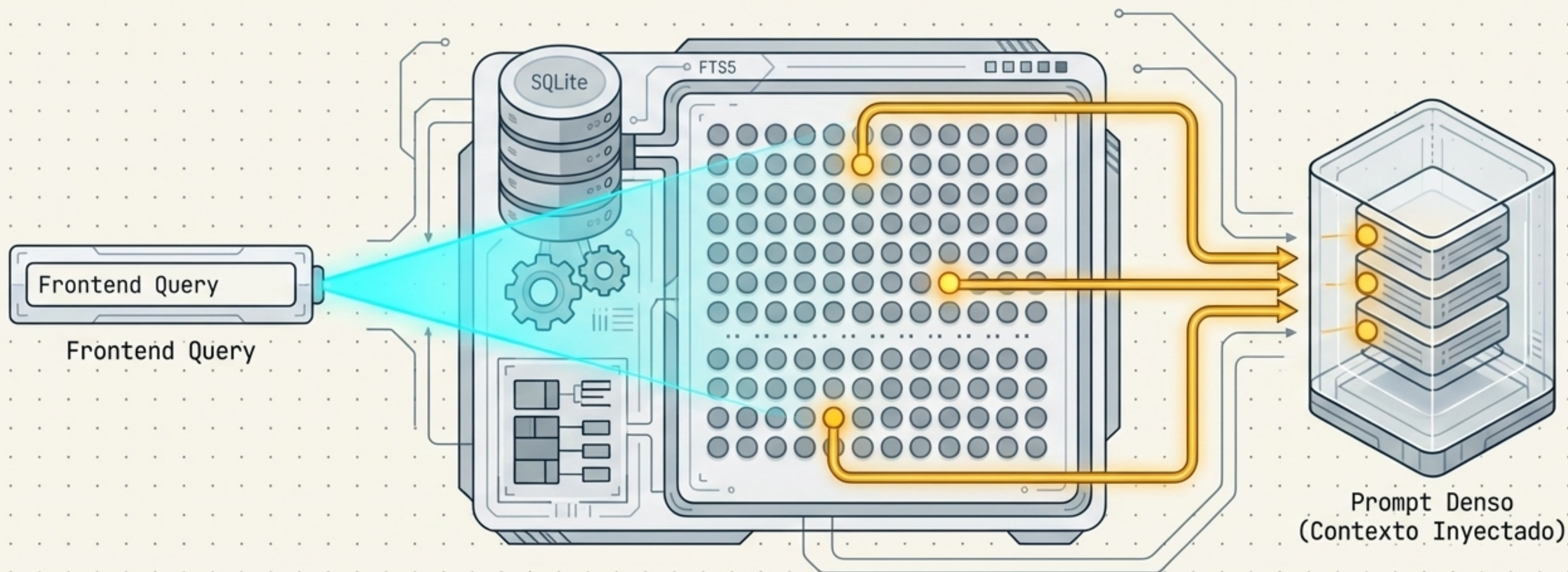
Fase 3 – Auto-Mejora de Skills (El CI/CD del Agente)

Modificación mutable del estado. El agente no solo corrige el error en la sesión actual; reescribe el código fuente del Skill original para prevenir la regresión de esa misma clase de problema.



Fase 4 - Recuperación Cruzada FTS5 (Búsqueda Full-Text)

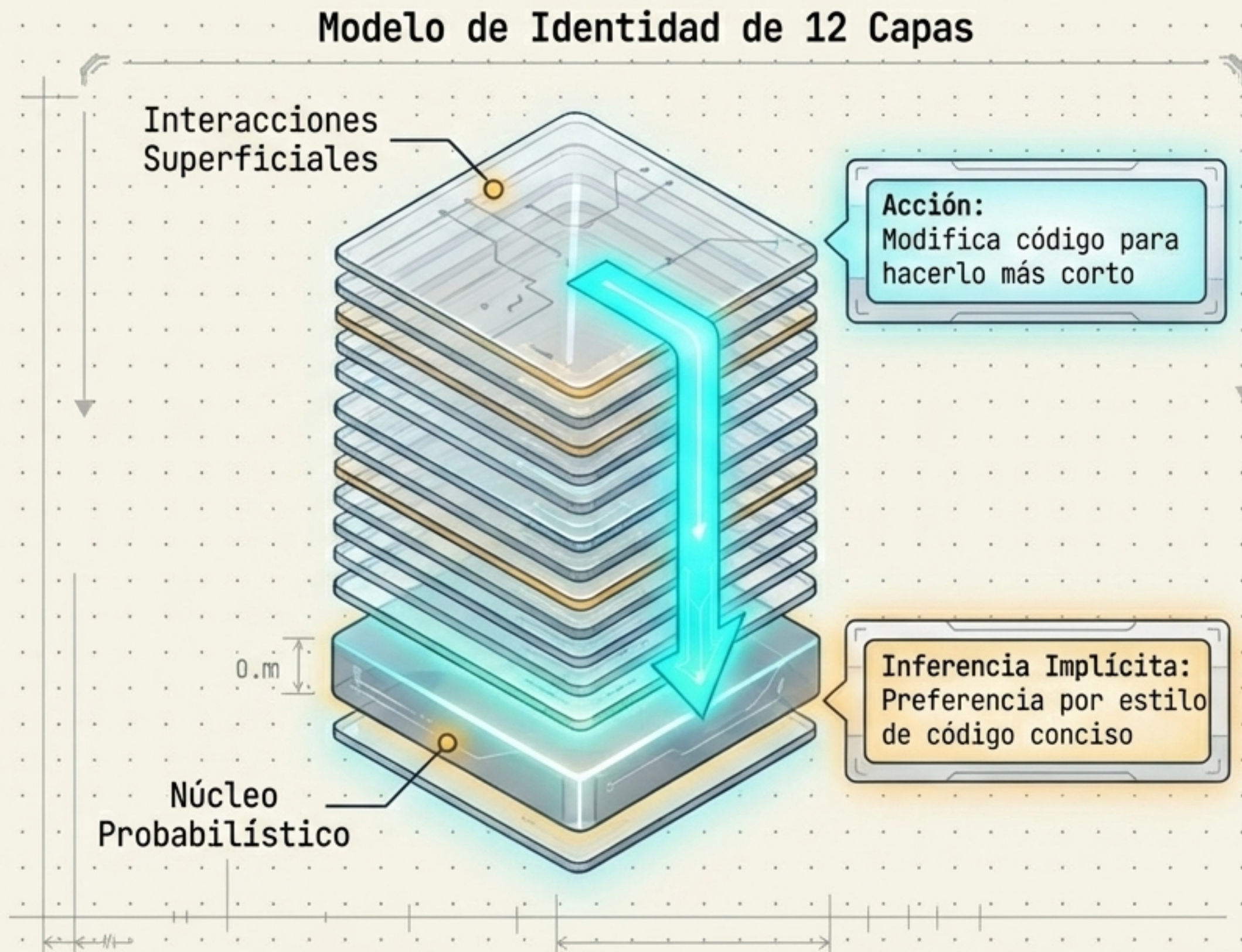
Inyección selectiva vía FTS5 local. Abandona la carga de contexto estática a favor de consultas dinámicas por tópico. Al ser 100% local, la portabilidad de memoria es trivial: simplemente copia el directorio ~/.hermes/ a otra máquina.



Fase 5 - Modelado de Usuario (Inferencia de Identidad Honcho)

Integración probabilística para extraer la identidad del usuario a partir del comportamiento iterativo.

Extrae patrones generalizados, mutando los defaults del agente sin requerir configuración declarativa explícita.



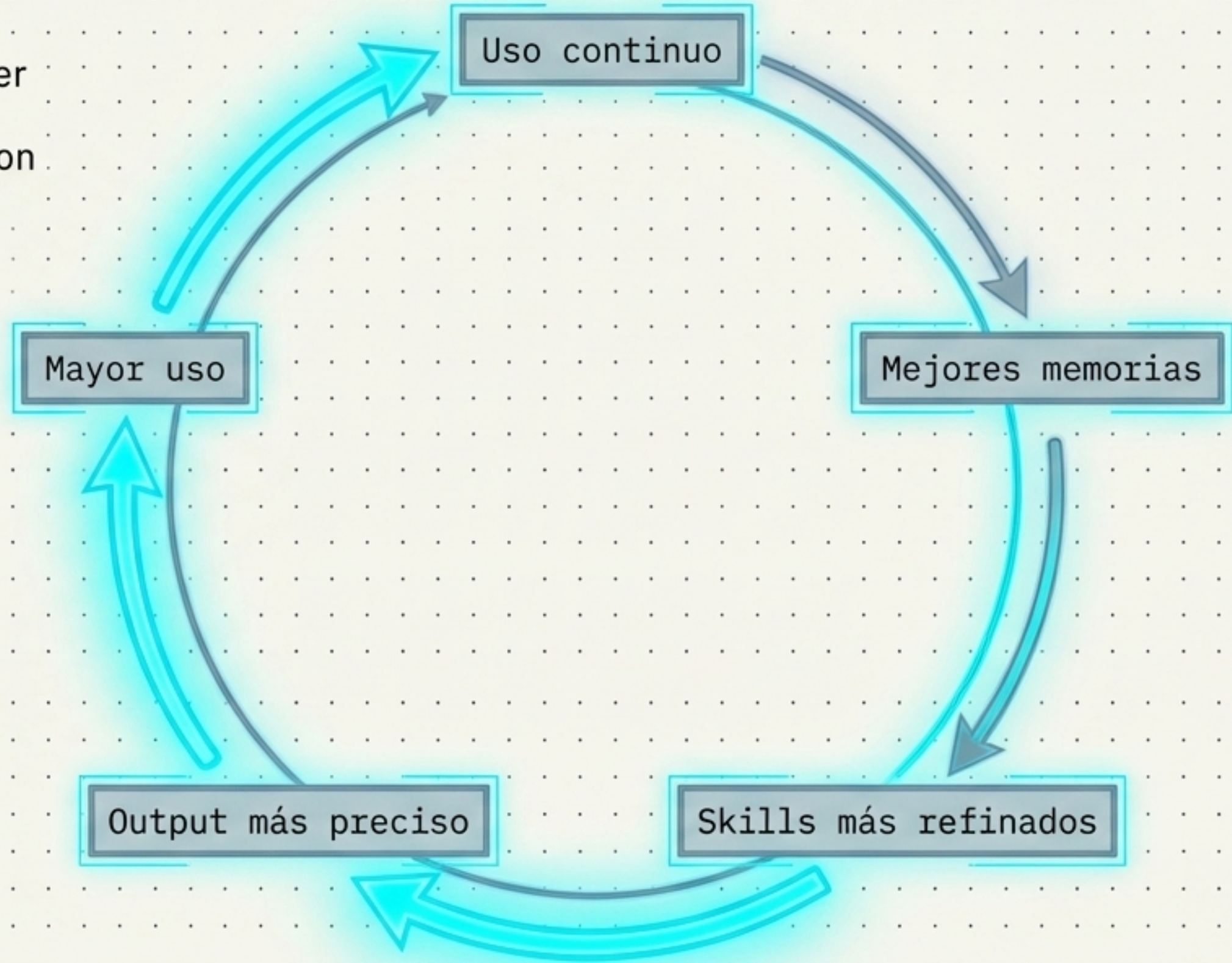
Matriz Arquitectónica: Manual vs. Automatizado

Reducción de la barrera de fricción a cero. Transición de un arnés de reglas hardcoded (declarativo) a un arnés de reglas inferido (autónomo).

Dimensión	Enfoque Mitchell (Manual)	Enfoque Hermes (Automatizado)
Fuente de regla	El humano detecta el problema y lo escribe	El agente extrae reglas del feedback
Ubicación de almacenamiento	CLAUDE.md (un solo archivo)	Múltiples archivos .md + SQLite DB
Trigger de mejora	Solo cuando el humano lo recuerda	Evaluación automática post-uso
Portabilidad Cross-project	Copia manual de CLAUDE.md	Skills globales compartidos
Velocidad de iteración	Depende de la diligencia humana	Continua y sin fricción

Dinámica de Sistemas: El Efecto "Flywheel"

· Aceleración de un solo usuario (Single-user acceleration). A diferencia de los LLMs tradicionales que degradan su contexto con el tiempo, el ratio señal-ruido de Hermes mejora a partir de los 3 a 5 días de uso constante. Las partes se fortalecen simultáneamente.

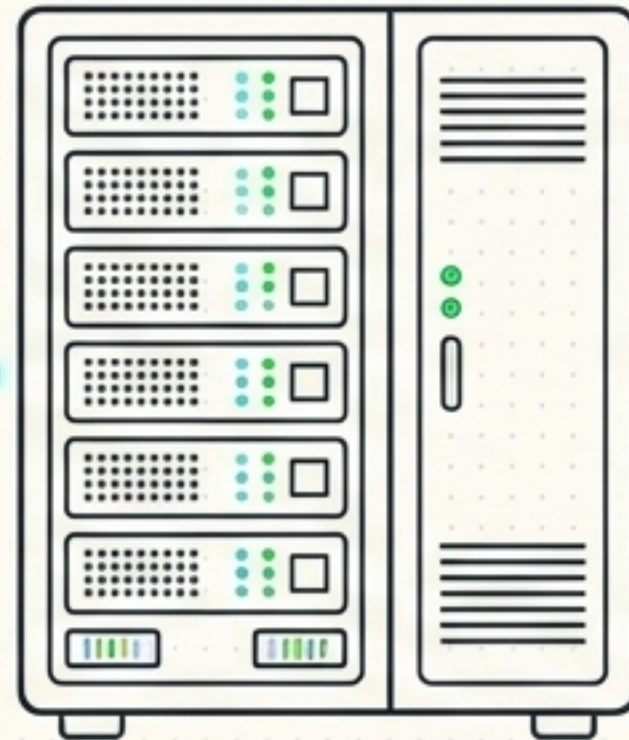


Topología de Despliegue (Recomendación Táctica)

Para maximizar la capitalización del estado (compounding state), la arquitectura óptima exige un entorno **local persistente**. Se requiere que el agente esté "vivo" asincrónicamente en un VPS de bajo costo operando 24/7 para ejecutar la recolección y curaduría continua de memoria.



Ciclo de reloj 24/7
(Procesos Asíncronos)



\$5 VPS



Almacenamiento
Persistente Local

Síntesis: El Arnés Auto-Tejido

El fin de la ingeniería de prompts estática. El agente no solo ejecuta el código; observa, abstrae, consolida y redefine su propia arquitectura operativa para el siguiente ciclo iterativo. El sistema se conduce solo.

