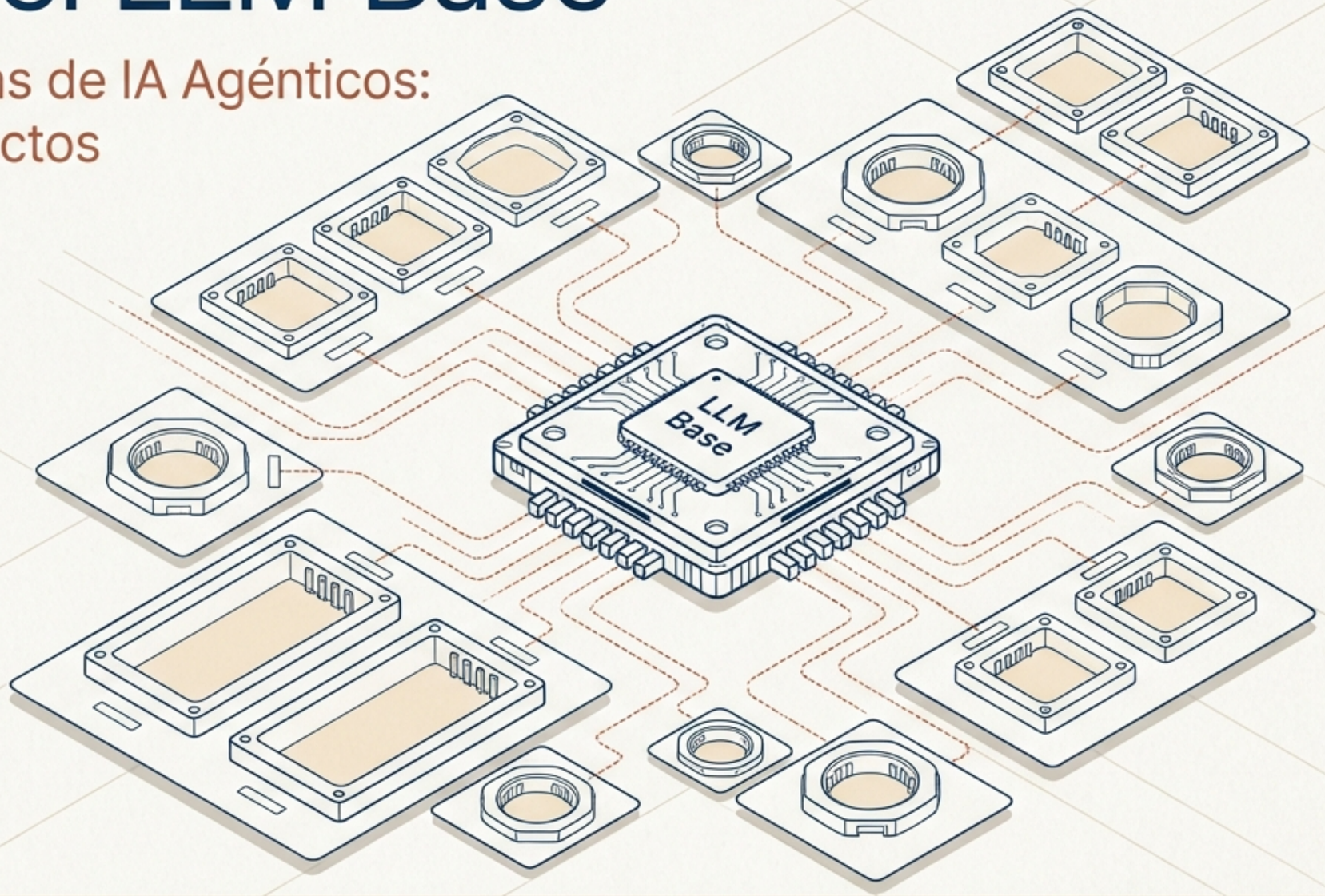
Más Allá del LLM Base

Construyendo Sistemas de IA Agénticos: El Manual para Arquitectos



NotebookLM

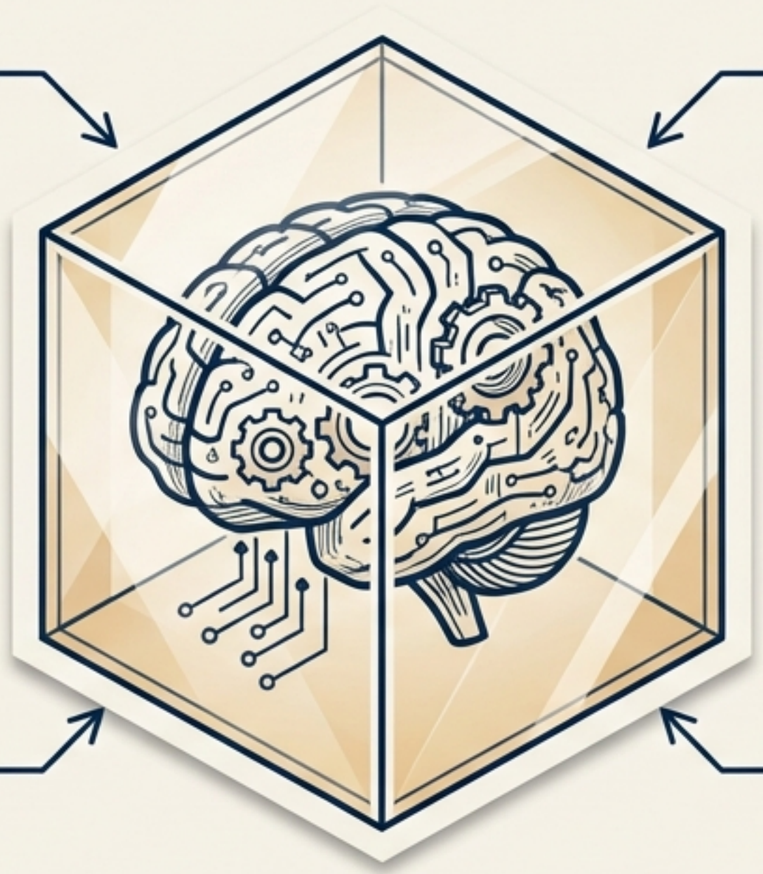
El Límite del Modelo Aislado

Conocimiento Desactualizado

Fechas de corte estáticas. El modelo desconoce tendencias recientes o vocabulario actual.

Ventana de Contexto Limitada

Imposibilidad de procesar archivos masivos. Amnesia en textos largos.



Alucinaciones y Falta de Fuentes

Incapacidad de citar documentos reales. Riesgo crítico en sectores legales o médicos.

Comportamiento Difícil de Controlar

Inconsistencia en tono y formato. Riesgo de respuestas tóxicas o indeseadas.

Un modelo fundacional es un motor potente, pero sin un chasis, no es un vehículo funcional.

NotebookLM

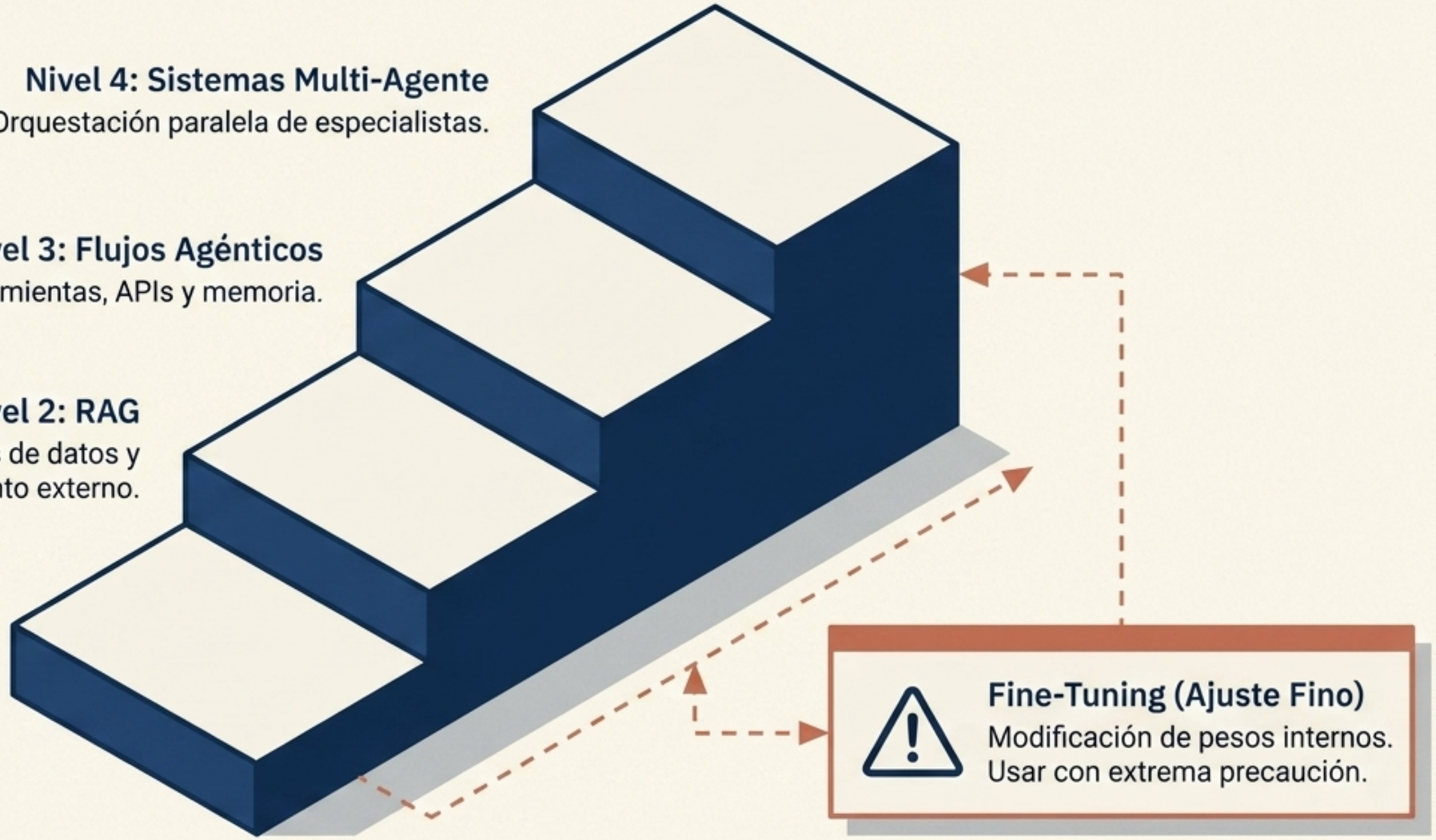
El Espectro de Optimización

Nivel 4: Sistemas Multi-Agente
Orquestación paralela de especialistas.

Nivel 3: Flujos Agénticos
Uso de herramientas, APIs y memoria.

Nivel 2: RAG
Conexión a bases de datos y conocimiento externo.

Nivel 1: Prompt Engineering
Guía estructural y de formato.



Fine-Tuning (Ajuste Fino)
Modificación de pesos internos.
Usar con extrema precaución.

NotebookLM

Dinámica Humano-IA: ¿Centauro o Cyborg?

El Centauro



Estrategia: Dividir y delegar.

Tareas masivas enviadas de una sola vez al modelo (ej. Crea esta presentación completa). Útil, pero propenso a errores en los límites de capacidad.

El Cyborg



Estrategia: Iteración continua.

Interacción rápida, correcciones en tiempo real y entrelazamiento de habilidades. (El enfoque recomendado para tareas complejas dentro de la frontera de la IA).

NotebookLM

Prompt Engineering: La Primera Capa de Control

Zero-Shot Genérico

Resume este documento.

Baja predictibilidad.
Sin contexto ni formato.

Plantilla Optimizada / Few-Shot

Actúa como un experto en energías renovables... → [Rol]

...presentando ante responsables de políticas. → [Audiencia]

Extrae exactamente 5 viñetas... → [Formato]

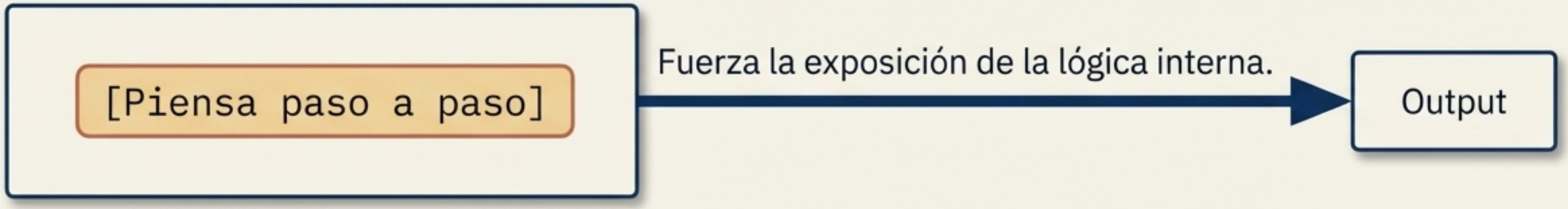
Usa estos metadatos del usuario: [Idioma: Español] → [Contexto Inyectado]

Takeaway: La precisión de la salida es directamente proporcional a la estructura de la entrada.

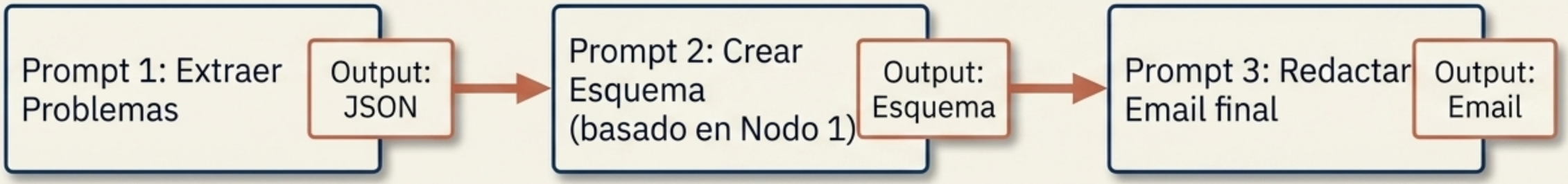
NotebookLM

Técnicas Avanzadas: Pensamiento y Encadenamiento

Chain of Thought (CoT)



Chaining (Encadenamiento)



El encadenamiento aísla errores. Facilita la depuración.
Permite evaluar cada paso lógico por separado.

NotebookLM

La Trampa del Ajuste Fino (Fine-Tuning)

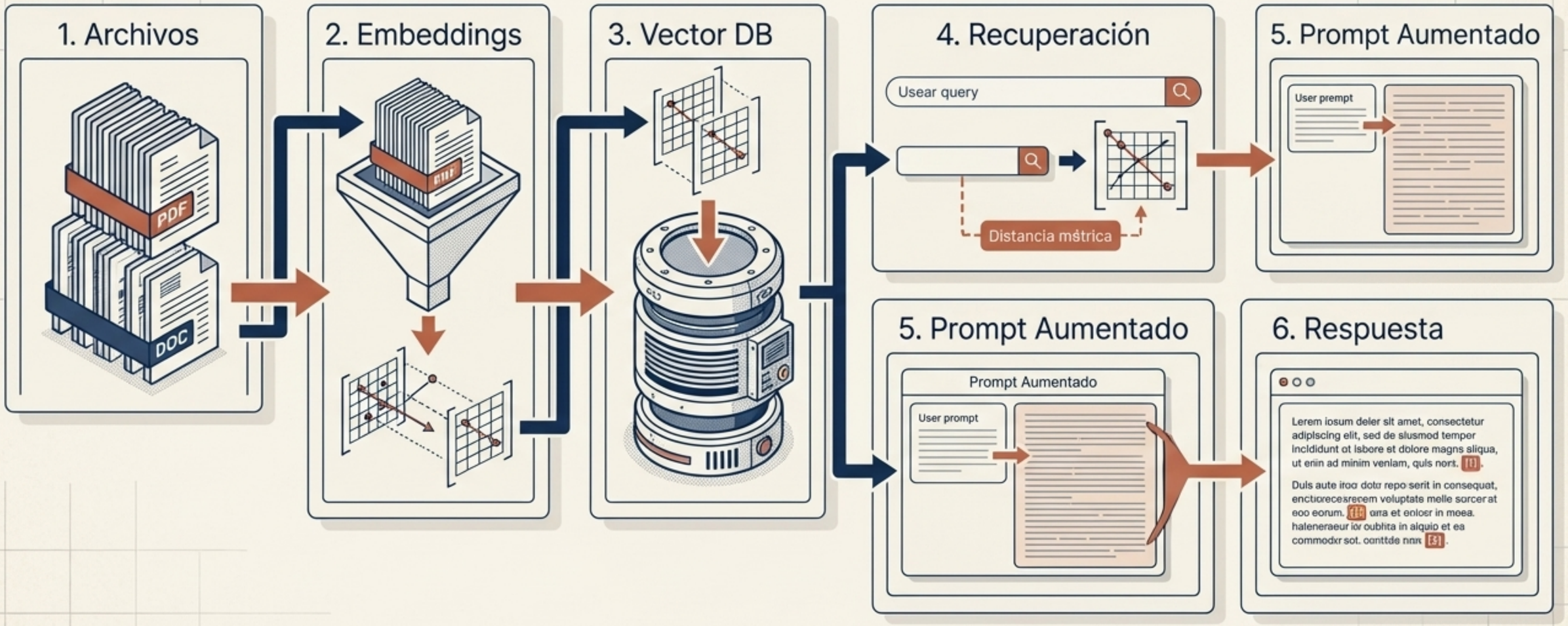
Estrategia	Costo/Tiempo	Riesgo de Obsolescencia	Ideales Para...
Ingeniería de Prompts	Bajo	Bajo	Flexibilidad general e iteración rápida
RAG	Medio	Bajo	Bases de conocimiento extensas y factuales
Fine-Tuning	**Alto**	**Altísimo**	Dominios hiper-específicos o formatos estrictos (ej. código propietario, lenguaje legal)

⚠ Anécdota: El Caso del Bot de Slack

Un intento de afinar un modelo con historiales de chat corporativos resultó en un bot que imitaba los retrasos humanos (Lo haré mañana) en lugar de ejecutar la tarea, demostrando el grave riesgo de sobreajuste conductual.

NotebookLM

RAG: Conectando el Conocimiento Externo

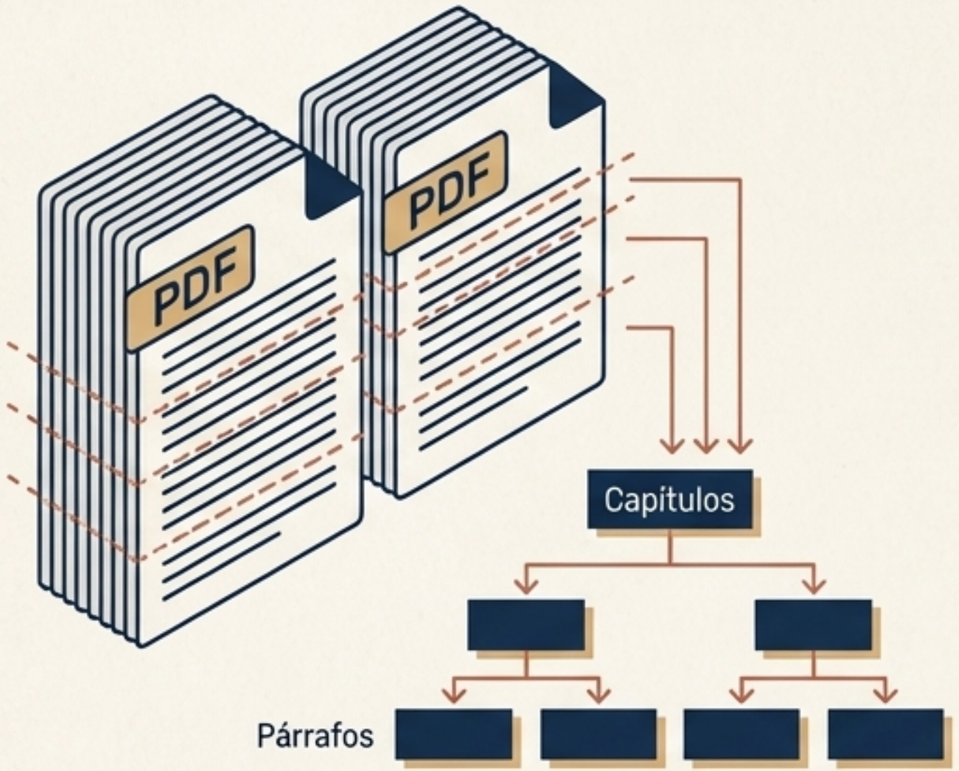


RAG resuelve las alucinaciones al forzar al modelo a leer un documento específico antes de hablar.

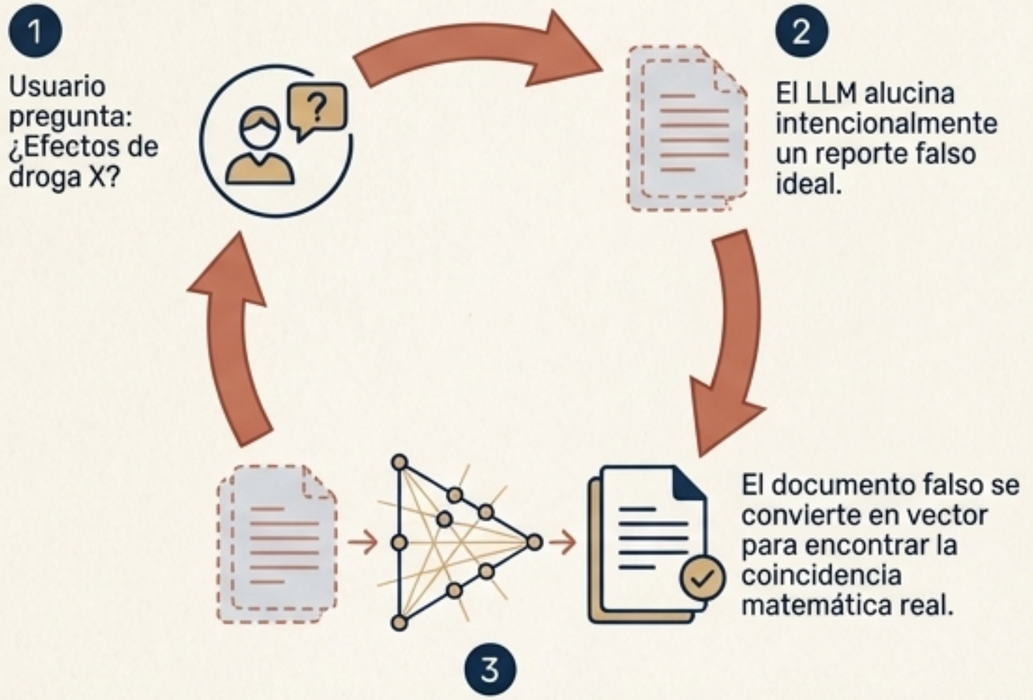
NotebookLM

Optimizando la Recuperación: Chunking y HyDE

Técnica 1: Chunking



Técnica 2: HyDE (Hypothetical Document Embeddings)

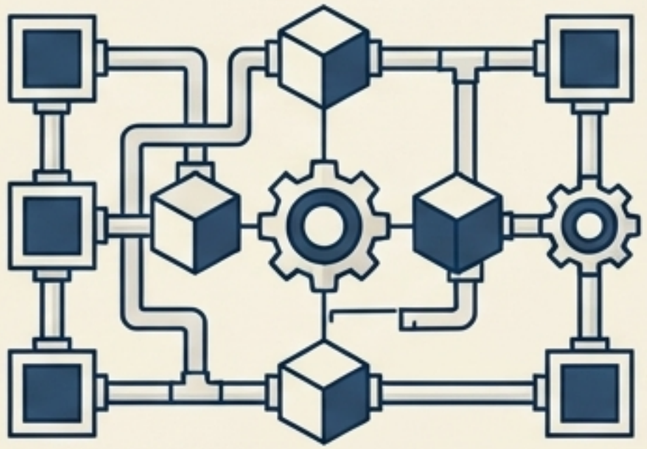


Takeaway: Cuando la pregunta no se parece semánticamente a la respuesta, generamos una respuesta hipotética para cruzar el abismo matemático.

NotebookLM

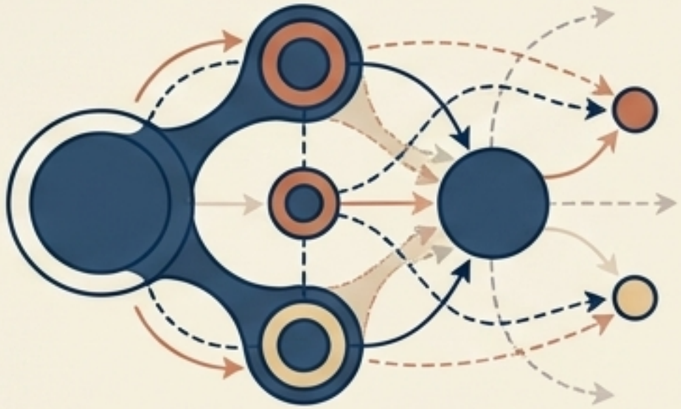
El Cambio de Paradigma: Determinista vs. Difuso

Software Tradicional (Determinista)



- Inputs: JSONs estructurados.
- Lógica: Condicional rígida (If/Then).
- Resultados: 100% predecibles.
- Un error de sintaxis detiene todo el sistema.

Software Agéntico (Difuso)



- Inputs: Texto o imágenes de forma libre.
- Lógica: Interpretada dinámicamente.
- Resultados: Probabilísticos y variables.
- El sistema intenta auto-corregirse ante errores.

La ingeniería difusa requiere Guardrails (barreras de contención) y sistemas Human-in-the-loop para mitigar el riesgo de resultados impredecibles.

NotebookLM

Anatomía de un Flujo de Trabajo Agéntico



Memoria de Trabajo (RAM)
Contexto inmediato y rápido
(Ej. Nombre en la sesión actual).

Memoria a Largo Plazo
Almacenamiento profundo
(Ej. Preferencias históricas).

Herramientas (Tools)
Las manos del agente (Búsqueda web, ejecución de código, APIs).

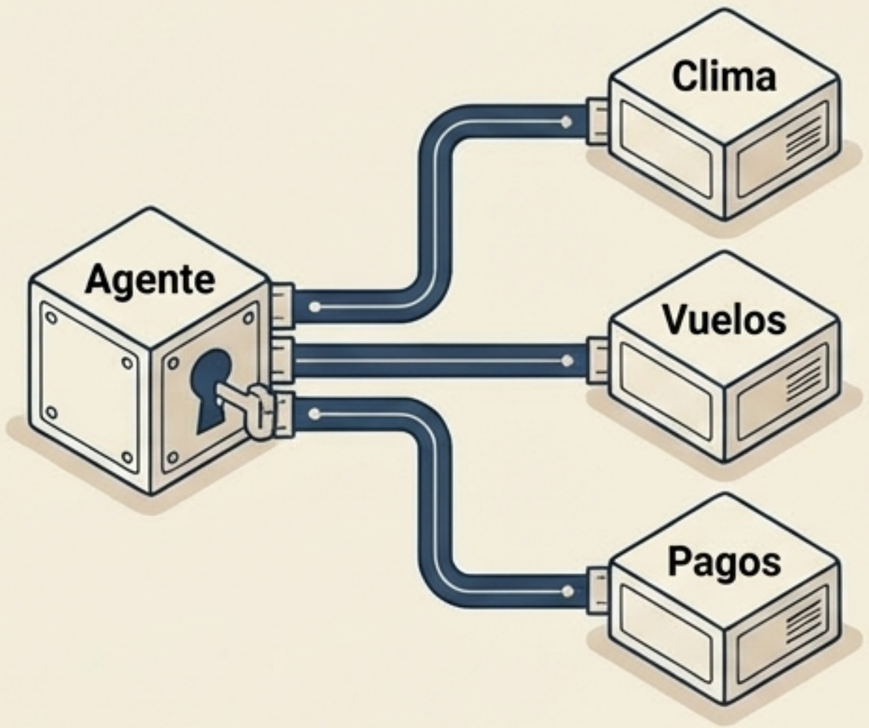
Planificador
Orquestador lógico que divide tareas complejas en pasos.

Un agente no es solo un prompt; es una arquitectura donde el LLM actúa como el motor de razonamiento central coordinando memoria y acción.

NotebookLM

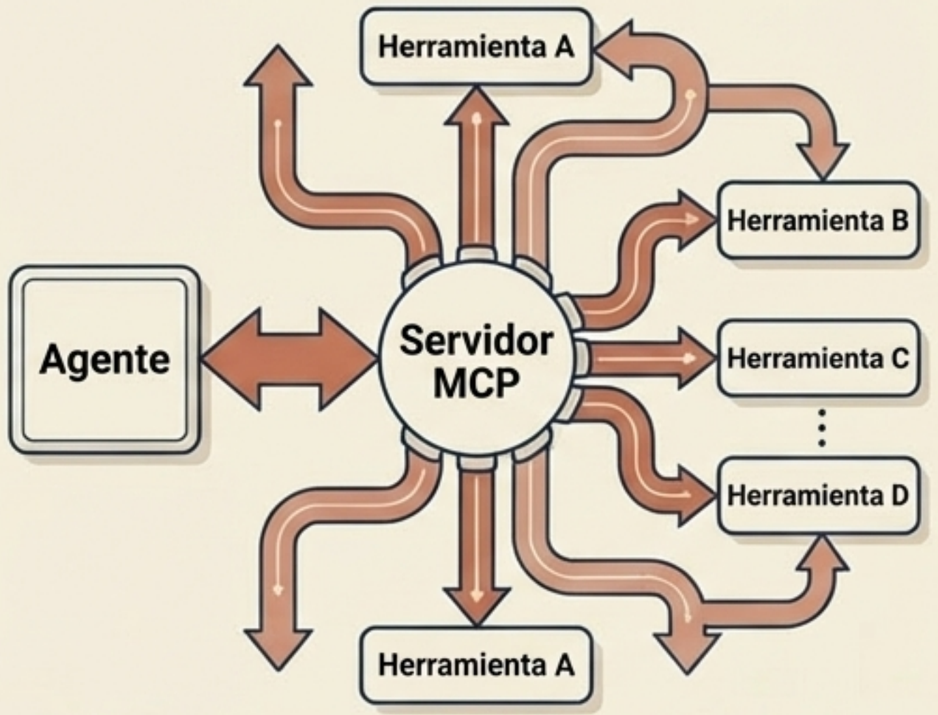
Conexión de Herramientas: Del API al MCP

API Tradicional



Conexiones rígidas.
El LLM debe ser programado con el formato JSON exacto para cada herramienta.
Difícil de escalar.

MCP (Model Context Protocol)



Negociación dinámica.
El Agente y la Herramienta comunican los requisitos de datos en tiempo real.
Ecosistema estandarizado y escalable.

NotebookLM

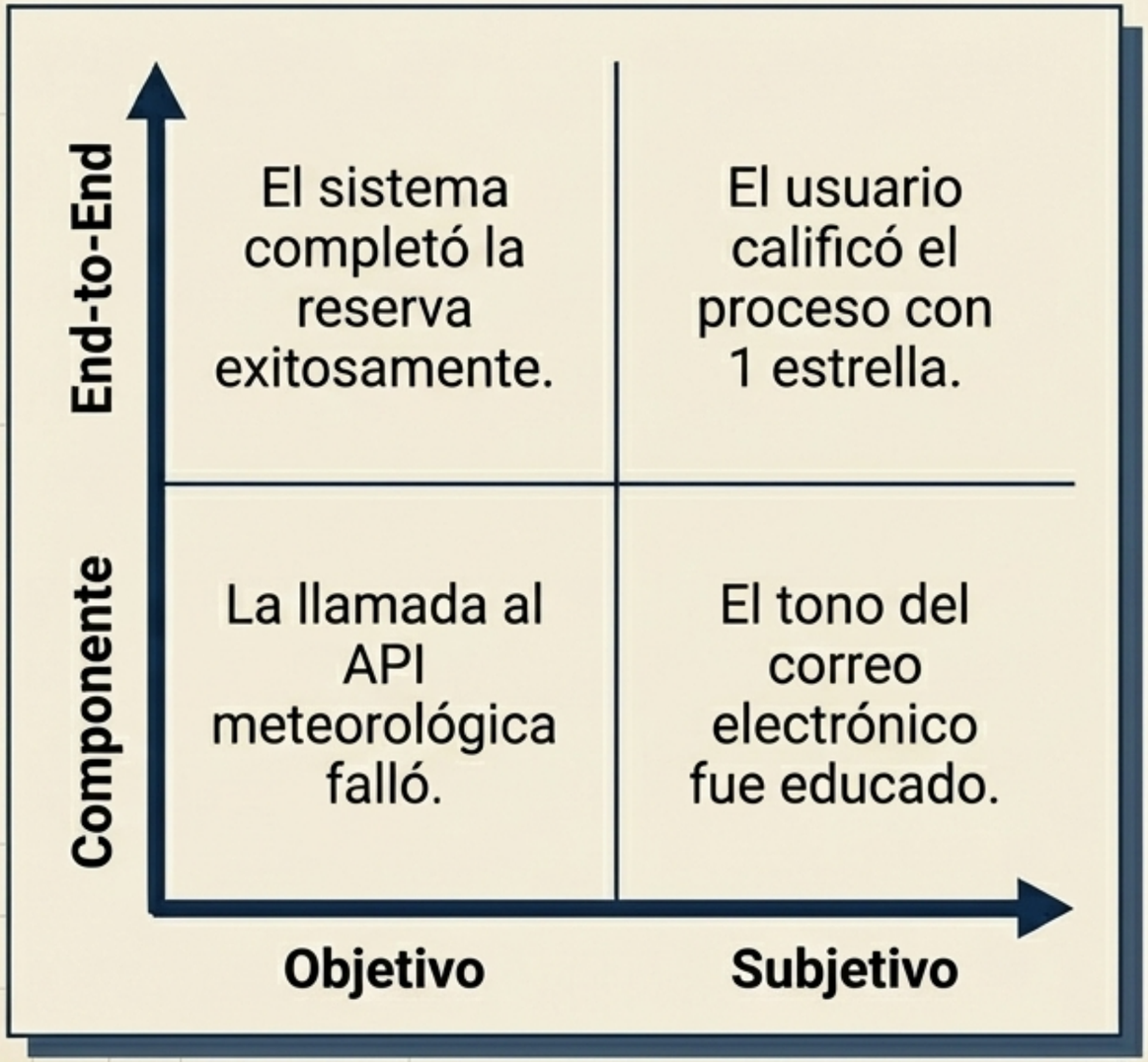
Sistemas Multi-Agente: Orquestación en Paralelo



¿Por qué no un solo mega-agente? La paralelización permite mayor velocidad, evita la confusión del modelo central y facilita la reutilización de especialistas.

NotebookLM

Evals: ¿Cómo sabemos si funciona?





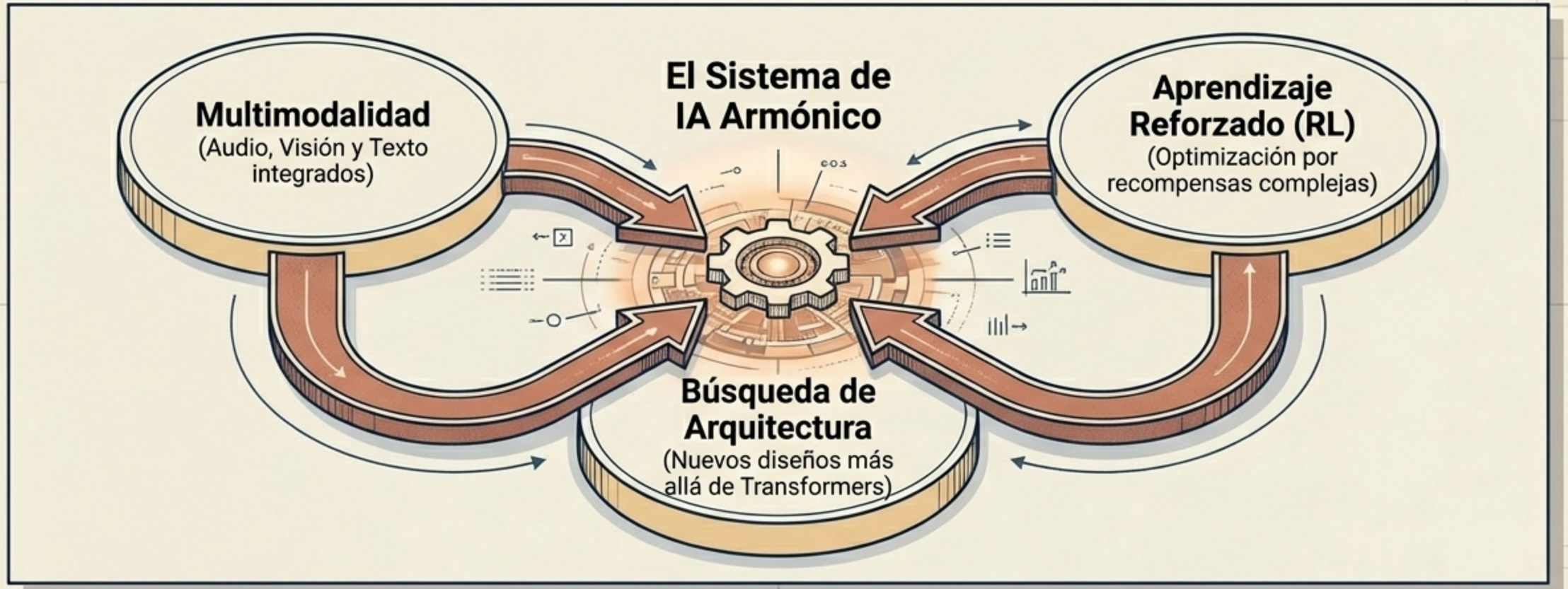
LLMs como Jueces

Para evaluar rúbricas subjetivas a escala (como “educación” o “claridad”), **no** uses humanos.

Implementa un modelo secundario (ej. Llama) programado **estrictamente como un juez automatizado en tus pruebas unitarias.**

NotebookLM

Síntesis: El Futuro de la Arquitectura



Los métodos cambian cada mes. Construye para la arquitectura de hoy, pero diseña tus sistemas con la modularidad necesaria para **adaptarse a la velocidad del mañana.**

NotebookLM