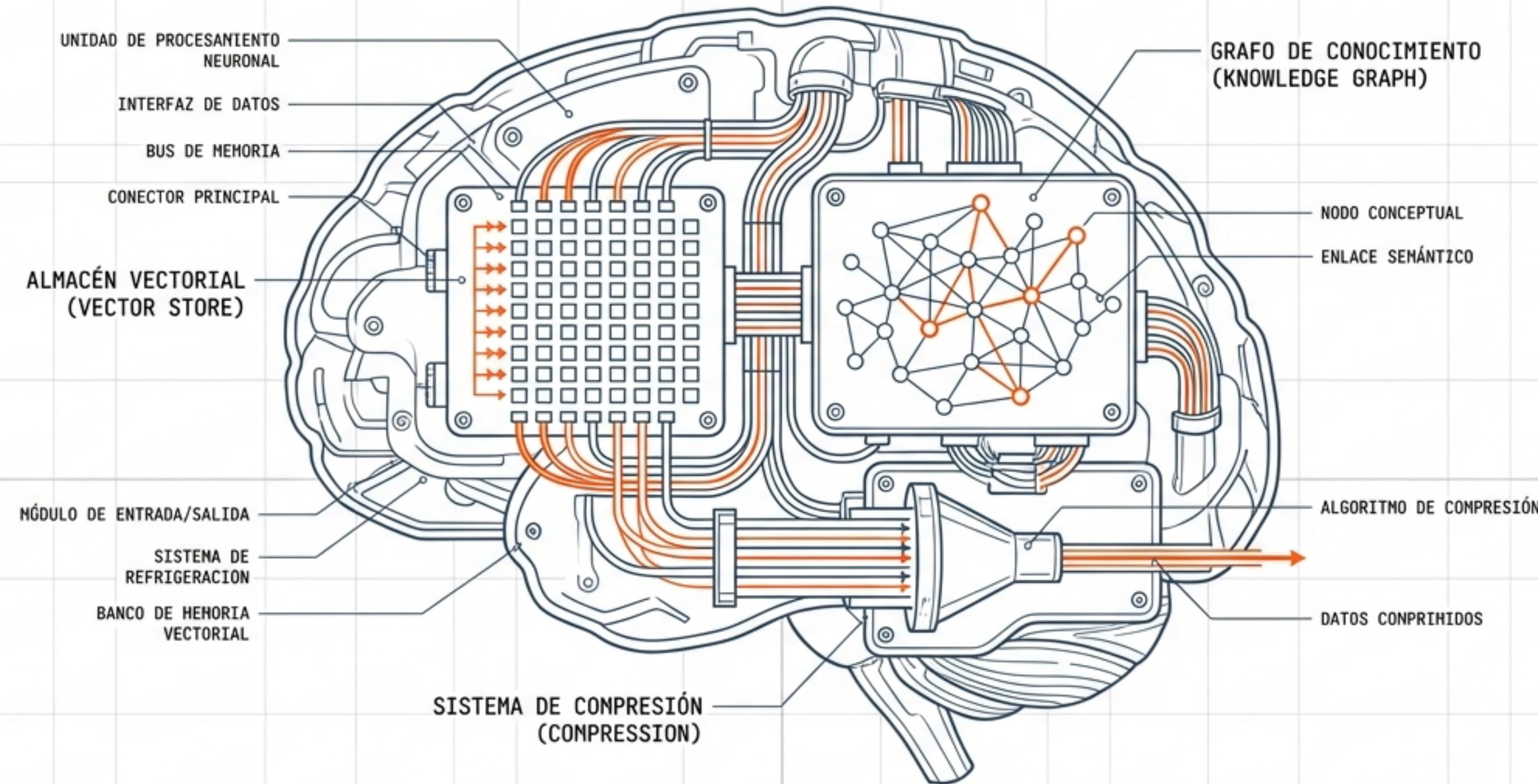


MÓDULO 5.2: DISEÑO DE MEMORIA AVANZADA



Arquitecturas de Memoria para Agentes de IA

Implementación de Vector Stores, Grafos de Conocimiento y Estrategias de Compresión.

Documento de Lectura Técnica

TECHNICAL SPECIFICATIONS & CONTEXT

El Desafío: La Ventana de Contexto y el Olvido

[CRITICAL LIMITATION DETECTED]

Vacíete: 95K [EB-55689]
Tekhoese: cef095AS_503
Ratuso: 1088L-HOV8.TOS

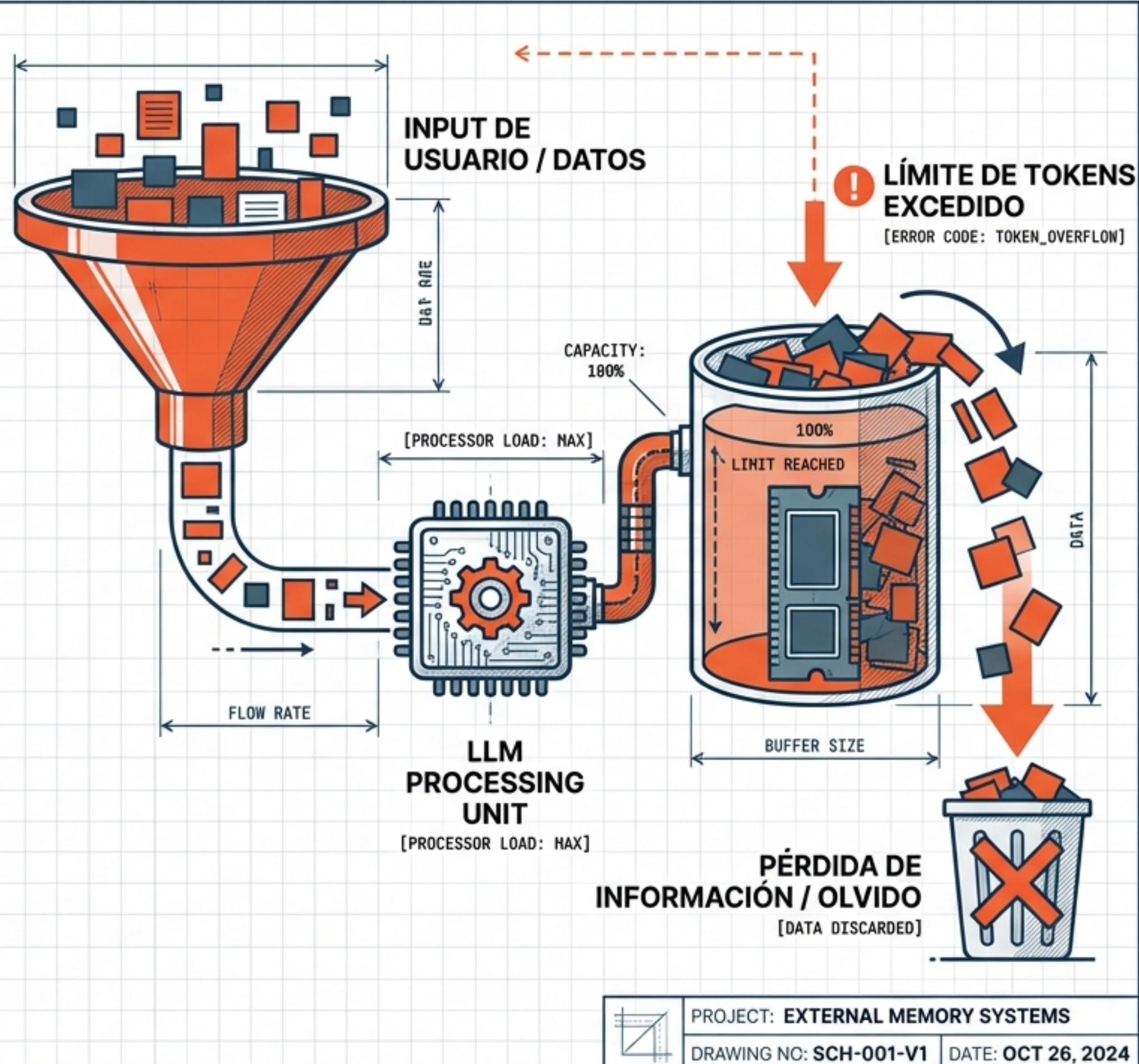
- Los LLMs poseen una 'Ventana de Contexto' finita (Context Window).
- La memoria perfecta es inútil si no se puede encontrar la información o si cuesta una fortuna en tokens procesarla.
- La Meta: Construir una memoria **externa** que sea **infinita**, infinita, recuperable y económicamente viable.

[MEMORY ARCHITECTURE GOAL]

ERR:C:\E95.18.289.93
ORARRO: E65 AT S.85.865
[JetBrains Hone]

[nemer puttic data prectess
esternal sets see sea zoot-terota
Geceet coca sosetane:
[JetBrains Hone]

SPLIT LAYOUT



Vector Stores: El Estándar para Búsqueda Semántica

Definición:

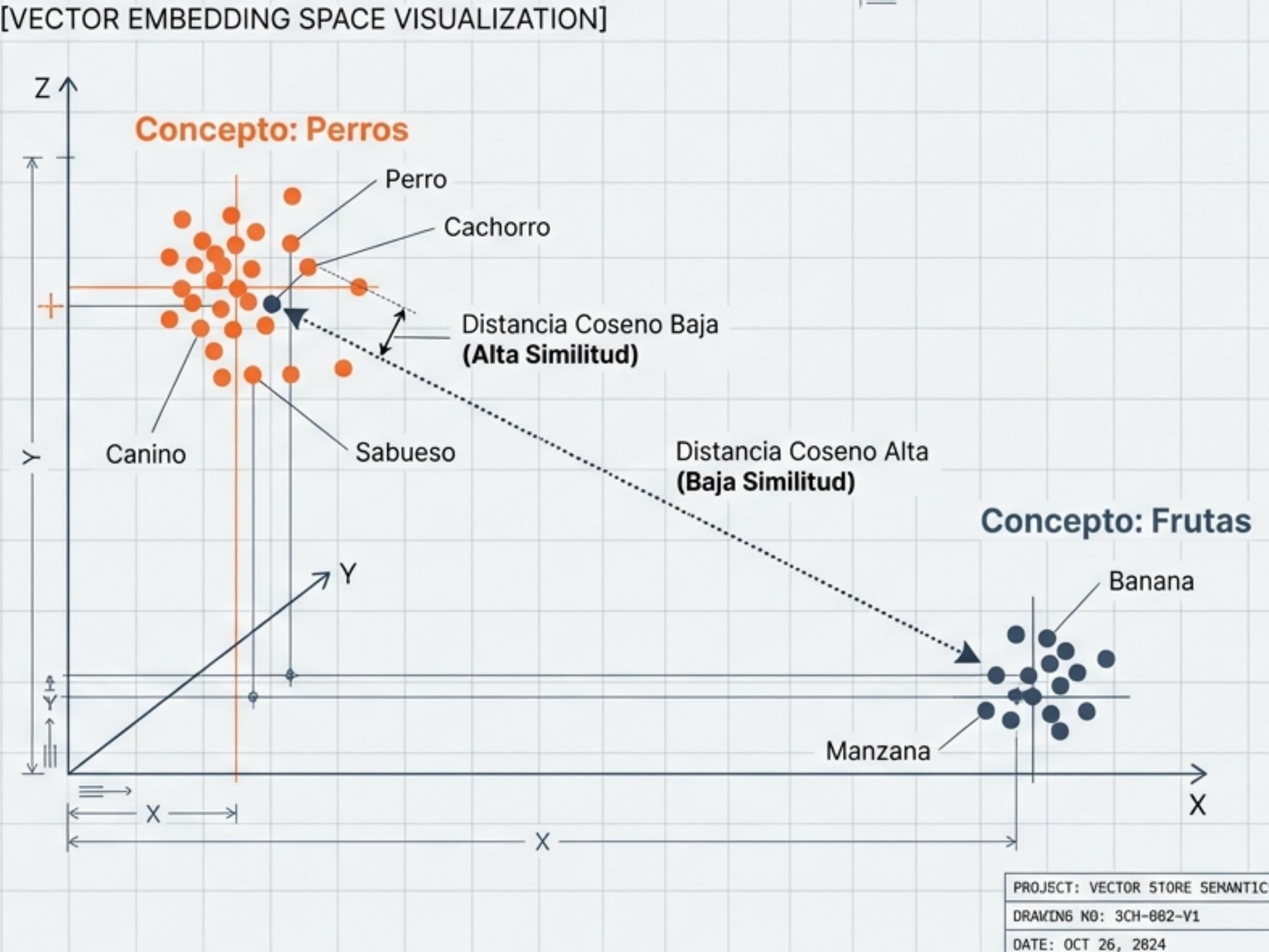
Los Vector Stores funcionan como la “base de datos nativa” para la IA.

La Diferencia:

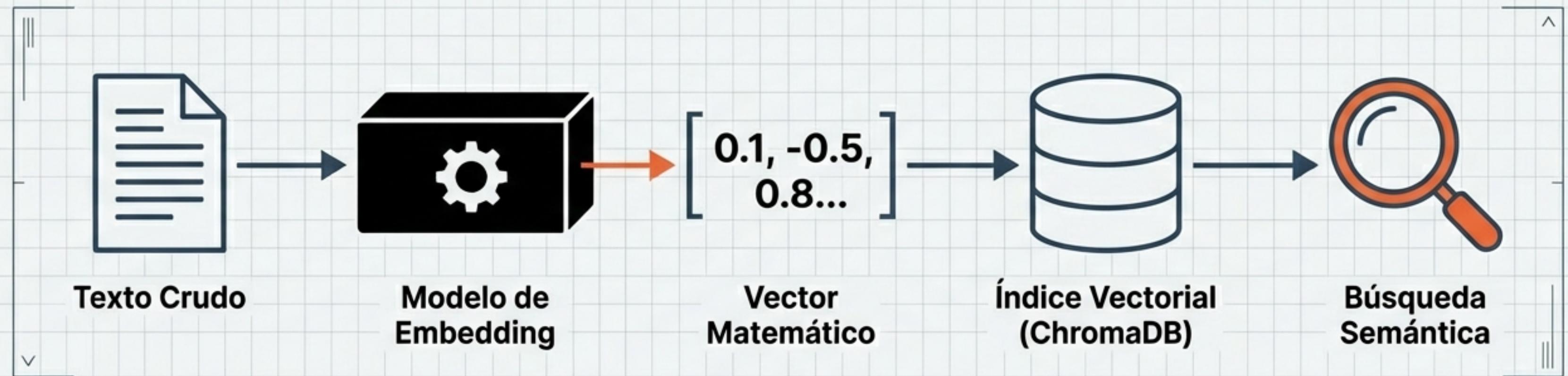
Optimizados para encontrar información por significado (similitud semántica), no solo por coincidencia exacta de palabras clave.

Capacidades:

- Escalabilidad a millones de documentos.
- Integración nativa con pipelines de RAG.



Arquitectura y Flujo de Datos Vectorial



1. Store

Conversión de texto a embedding numérico y almacenamiento.

2. Search

Utiliza similitud de cosenos para encontrar vectores cercanos en el espacio latente.

3. Filter

Uso crítico de Metadata para refinar la búsqueda antes del cálculo vectorial.

4. Manage

Operaciones CRUD (Create, Read, Update, Delete) de documentos.



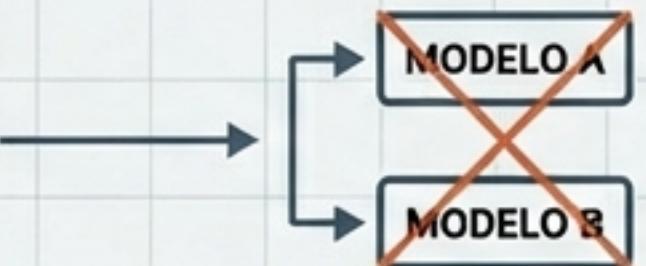
Implementación y Mejores Prácticas (Vector Stores)

Evitando errores comunes en producción (ChromaDB / Pinecone).



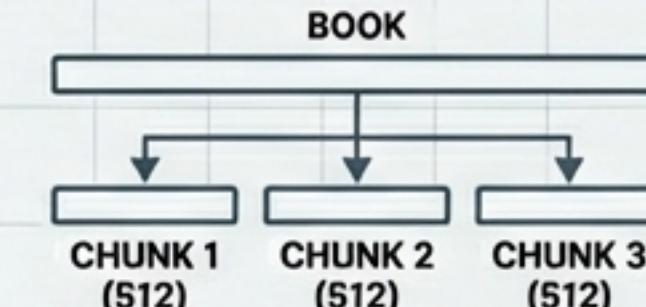
Consistencia de Modelos

Error Crítico: Cambiar el modelo de embedding a mitad de camino rompe todo el índice. Siempre usar el mismo modelo para insertar y consultar.



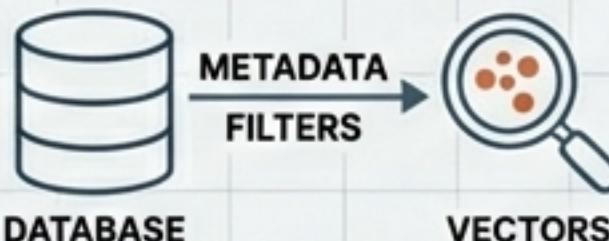
Estrategia de Chunking

No incrustar libros enteros como un solo vector. Dividir el texto en fragmentos (chunks) lógicos (ej. 512 tokens) para mantener el contexto preciso.



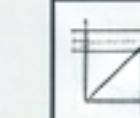
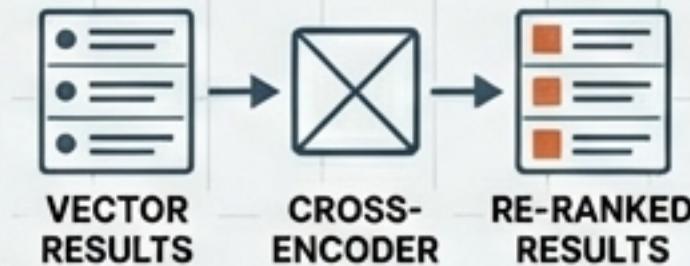
Filtrado Híbrido

Aplicar filtros de metadata en la base de datos **antes** de la búsqueda vectorial para reducir la latencia y aumentar la precisión.



Re-ranking

Reordenar los resultados vectoriales utilizando un modelo Cross-Encoder para mejorar la relevancia final de los documentos recuperados.



La Limitación de los Vectores: El "Eslabón Perdido"



El Problema:

- Los vectores son excelentes para encontrar cosas "similares" (por "vibra"), pero deficientes para relaciones exactas.
- Carecen de razonamiento transitivo (A lleva a B, B lleva a C).
- **Solución:** Se requiere una estructura que capture relaciones explícitas (Grafos).

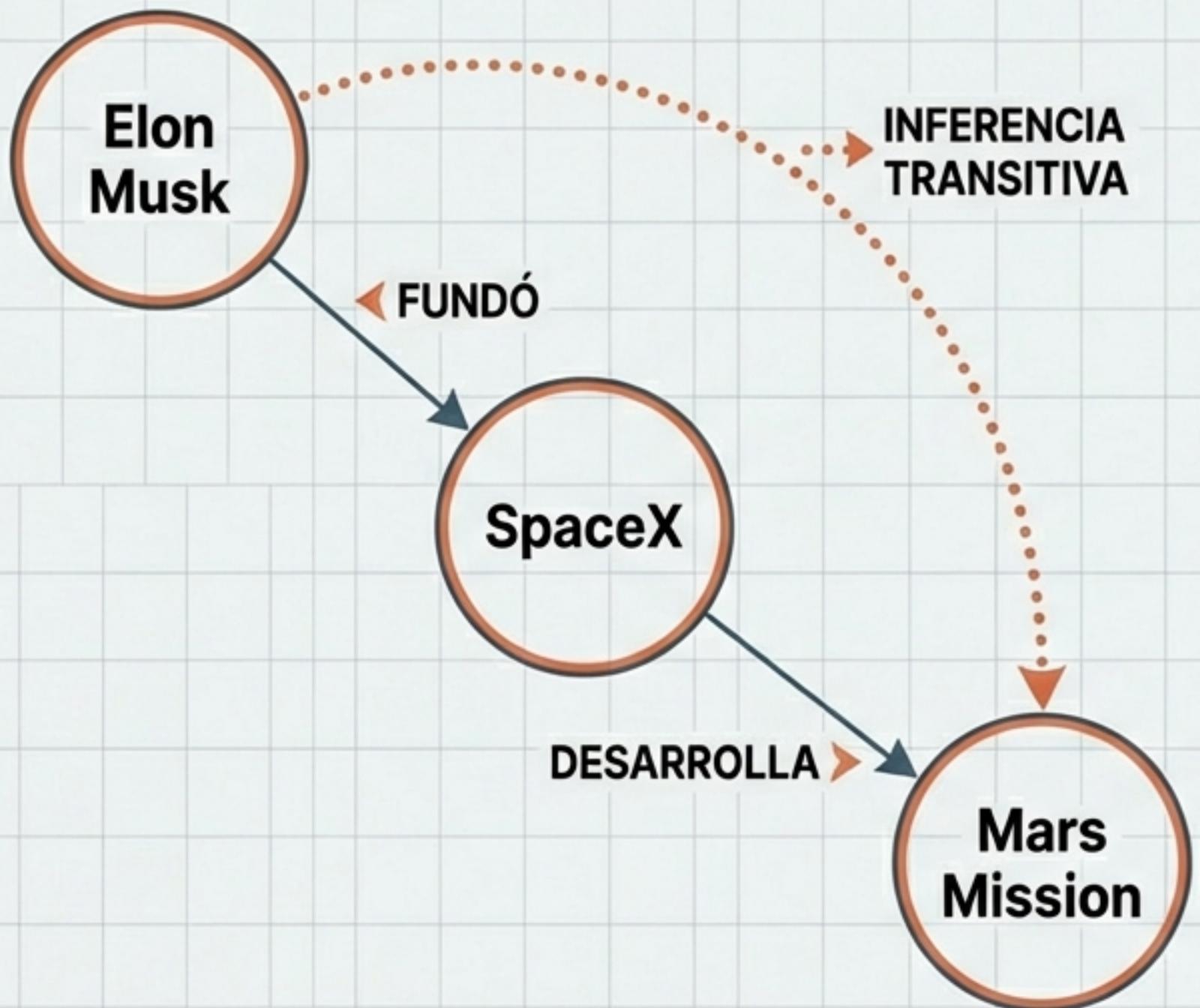


PROJECT: VECTOR LIMITATIONS & GRAPHS

DRAWING NC: VEC-LIM-001-V1

DATE: OCT 28, 2024

Grafos de Conocimiento: Razonamiento Estructurado



“Los grafos capturan el ‘cómo’ y el ‘por qué’ de las conexiones, no solo el ‘qué’.”

Core Concepts

- **Nodos (Entidades)**: Los sustantivos (Persona, Lugar, Concepto).
- **Aristas (Relaciones)**: Los verbos que conectan nodos (TRABAJA_EN, ES_PARTE_DE).
- **La Ventaja Única**: Razonamiento Transitivo. Si A afecta a B, y B afecta a C, el grafo permite deducir que A afecta a C.
- **Explicabilidad**: Muestra la ruta lógica exacta.



Comparativa de Arquitecturas: Vectores vs. Grafos

VECTOR STORES

Mejor para: Búsqueda difusa, texto no estructurado, consultas generales.

Fortaleza: Similitud semántica y escala masiva.

Debilidad: Alucinación de relaciones, falta de lógica dura.

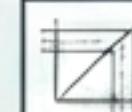
GRAFOS DE CONOCIMIENTO

Mejor para: Hechos explícitos, razonamiento multi-salto (multi-hop), trazabilidad.

Fortaleza: Precisión estructurada y explicabilidad.

Debilidad: Más difícil de construir y escalar.

Insight: Los agentes más avanzados utilizan una arquitectura híbrida (GraphRAG).



PROJECT: VECTOR vs. GRAPH ARCHITECTURE COMPARISON

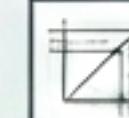
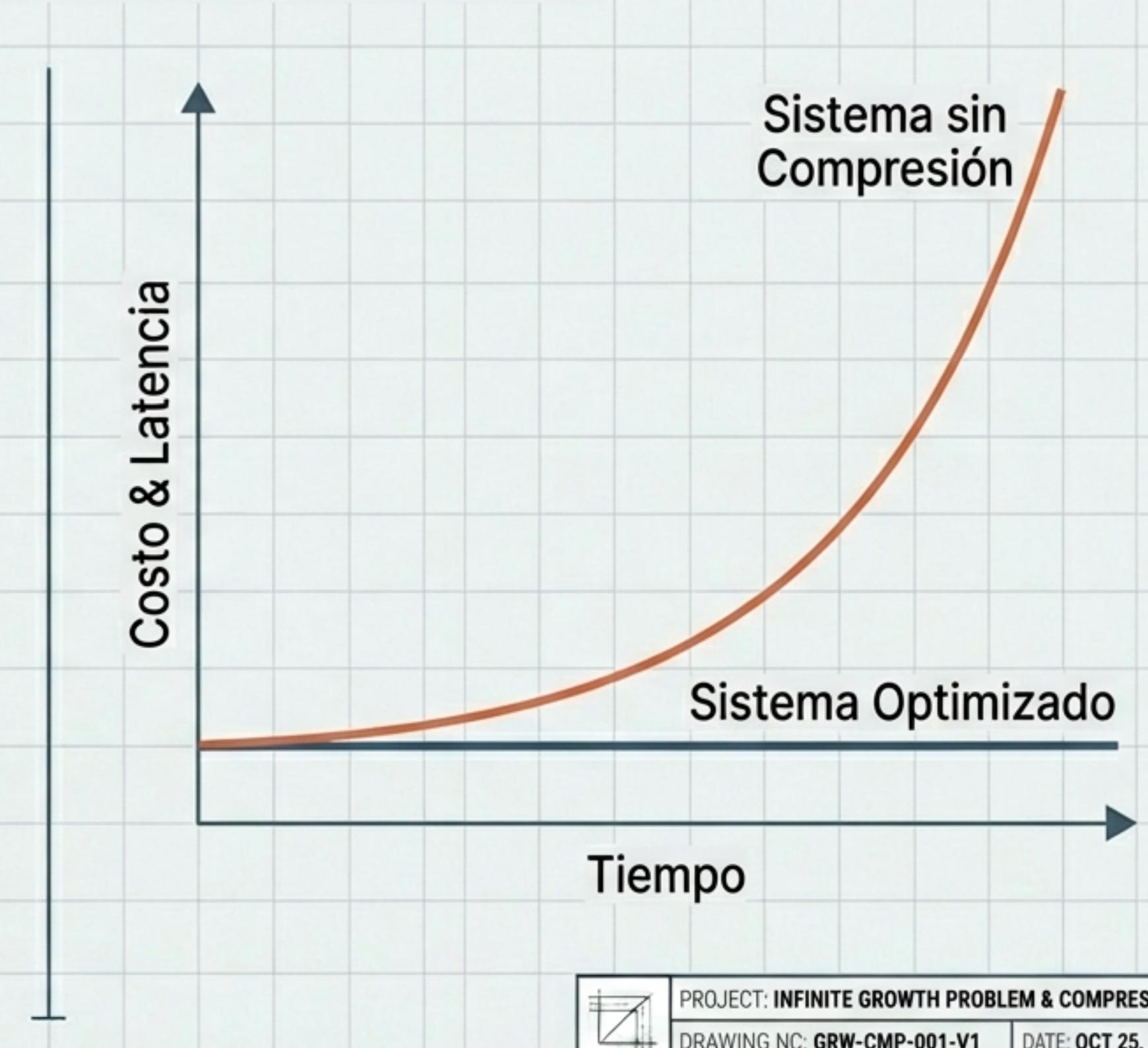
DRAWING NC: VEC-GRA-COM-001-V1

DATE: OCT 25, 2024

El Problema del Crecimiento 'Infinito'

Entropía del Sistema:

- A medida que los agentes operan, los logs y memorias crecen indefinidamente.
- Mantener todo en la ventana de contexto es costoso y confunde al modelo (exceso de ruido).
- **Necesidad:** Un sistema de "recolección de basura" inteligente que preserve la sabiduría pero descarte el ruido.

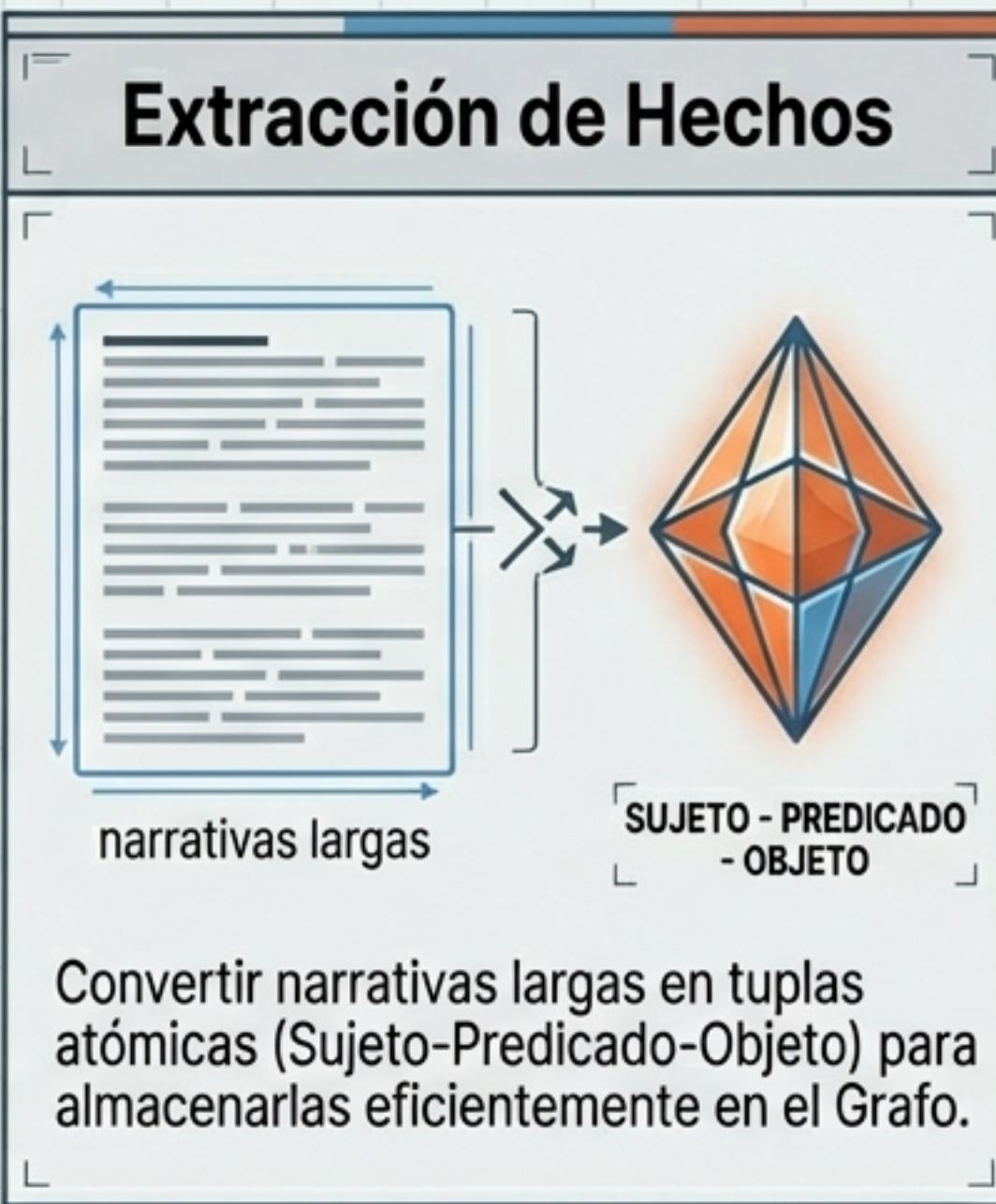
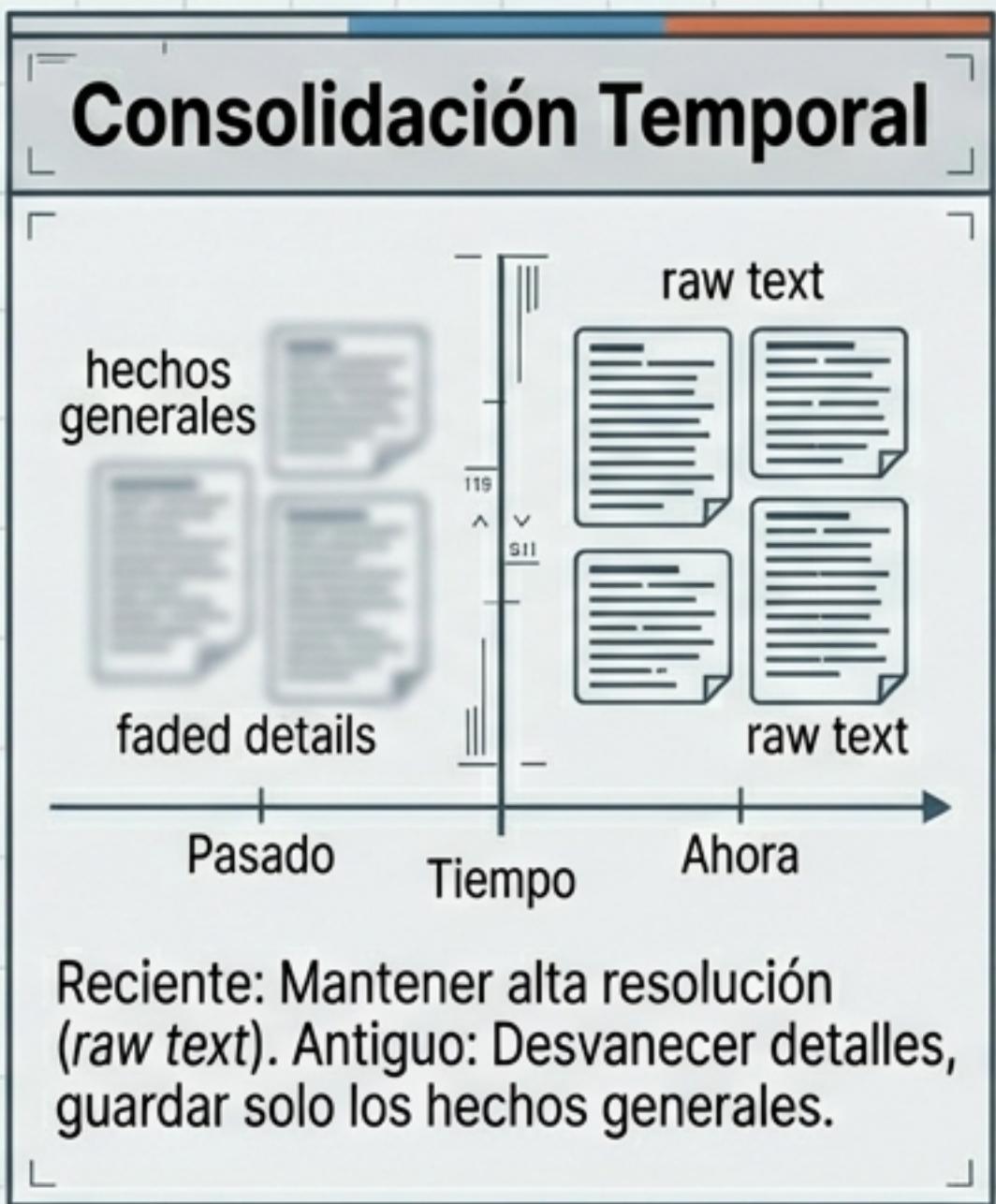
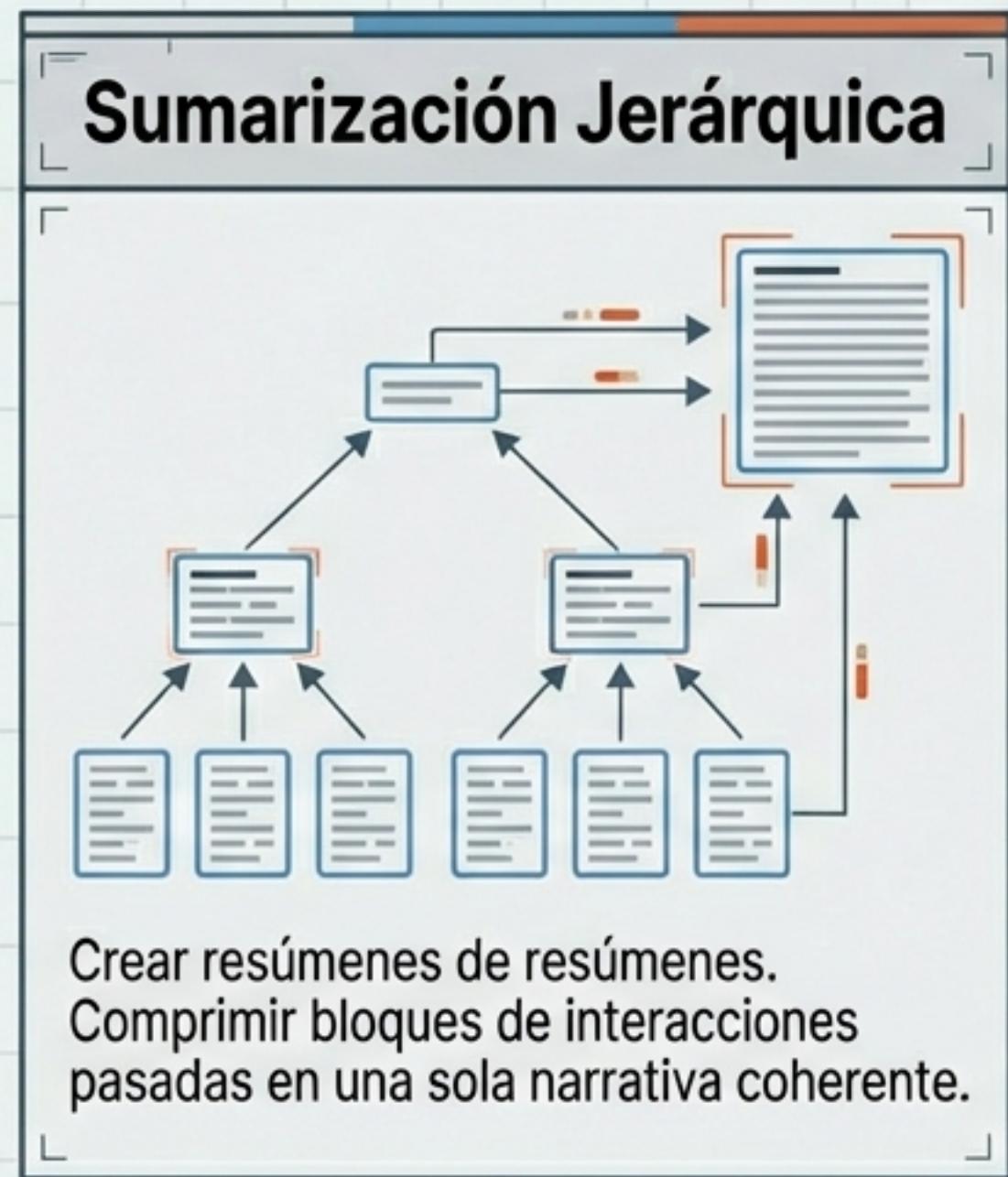


PROJECT: INFINITE GROWTH PROBLEM & COMPRESSION

DRAWING NC: GRW-CMP-001-V1

DATE: OCT 25, 2024

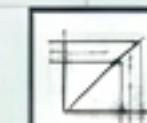
Estrategias de Compresión y Sumarización



* TECH NOTE: RECURSIVE ABSTRACTING

* TECH NOTE: TEMPORAL FADING

* TECH NOTE: ATOMIC TUPLE EXTRACTION

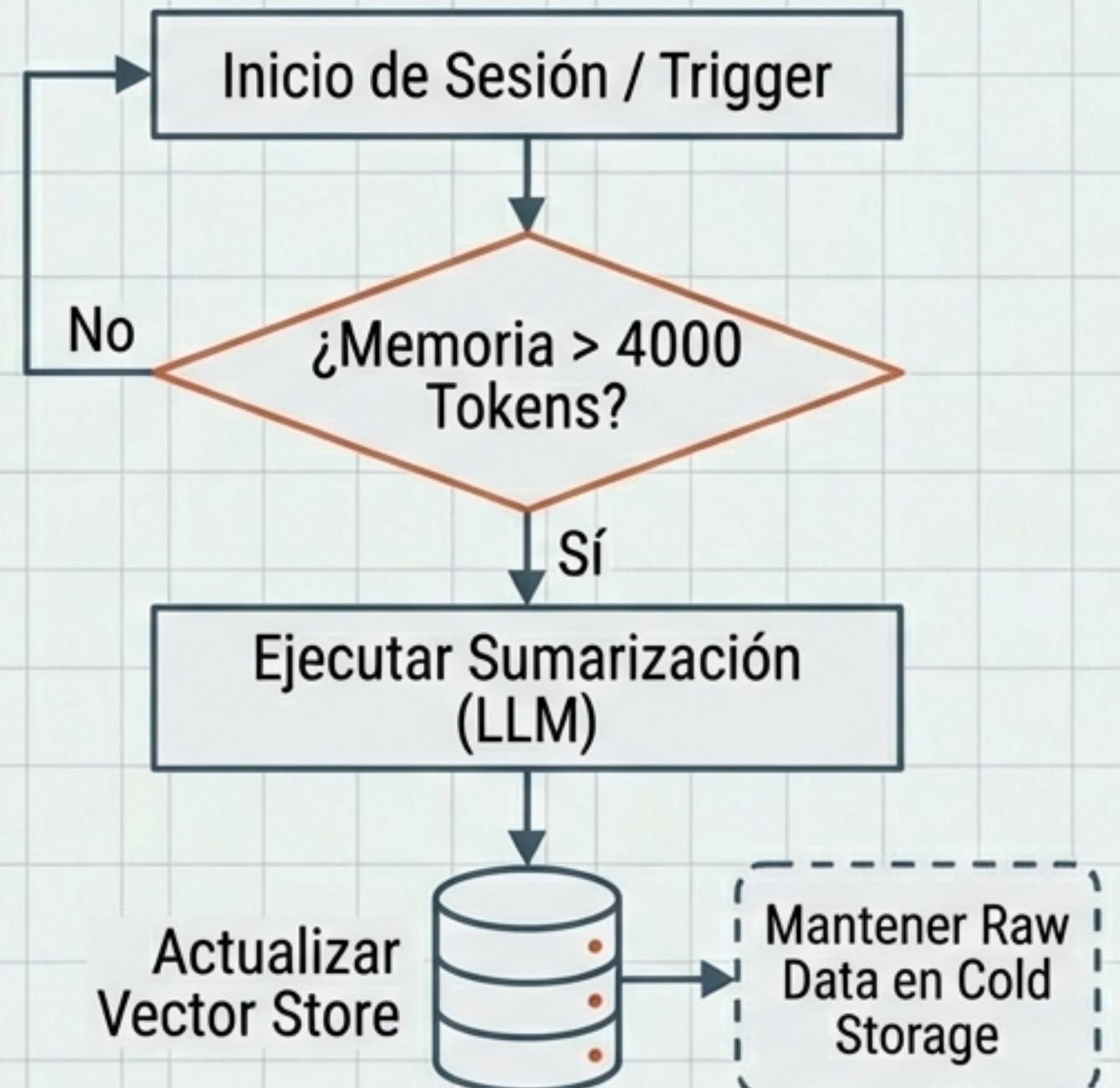


PROJECT: STRATEGIES FOR COMPRESSION & SUMMARIZATION

DRAWING NC: STR-CMP-SUM-001-V1

DATE: OCT 25, 2024

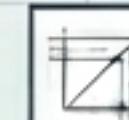
Implementando el Sistema de Compresión



Workflow:

- **Trigger (Disparador):** Ejecutar cuando la memoria alcanza X tokens.
- **Action (Acción):** Sumarizar el N% más antiguo o agrupar (cluster) memorias similares.
- **Resultado:** Carga de contexto reducida, IQ del agente mantenido, costos controlados.

* TECH NOTE: LOGIC FLOW



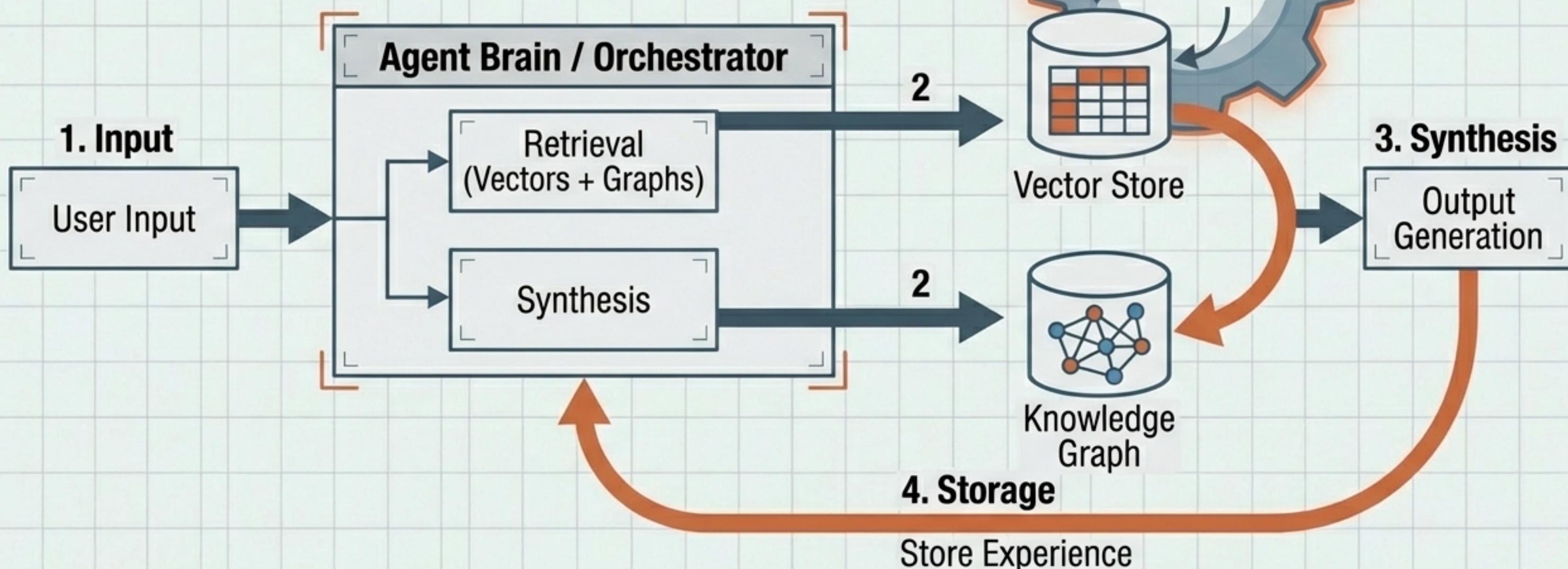
PROJECT: IMPLEMENTING COMPRESSION SYSTEM

DRAWING NC: FLOW-CMP-001-V1

DATE: OCT 25, 2024

La Arquitectura Unificada

Diagrama de Sistema Completo



Resumen y Conclusiones Clave

VECTOR STORES

Cimientos Semánticos.
(La "Vibra" y el contexto general).

GRAFOS

Precisión Estructurada.
(La Lógica y los hechos duros).

COMPRESIÓN

Eficiencia y Longevidad.
(El Mantenimiento y gestión de costos).



Pensamiento Final:

Construir memoria no se trata solo de almacenamiento; se trata de curar inteligencia. Un agente inteligente es aquel que sabe qué recordar y qué olvidar.

