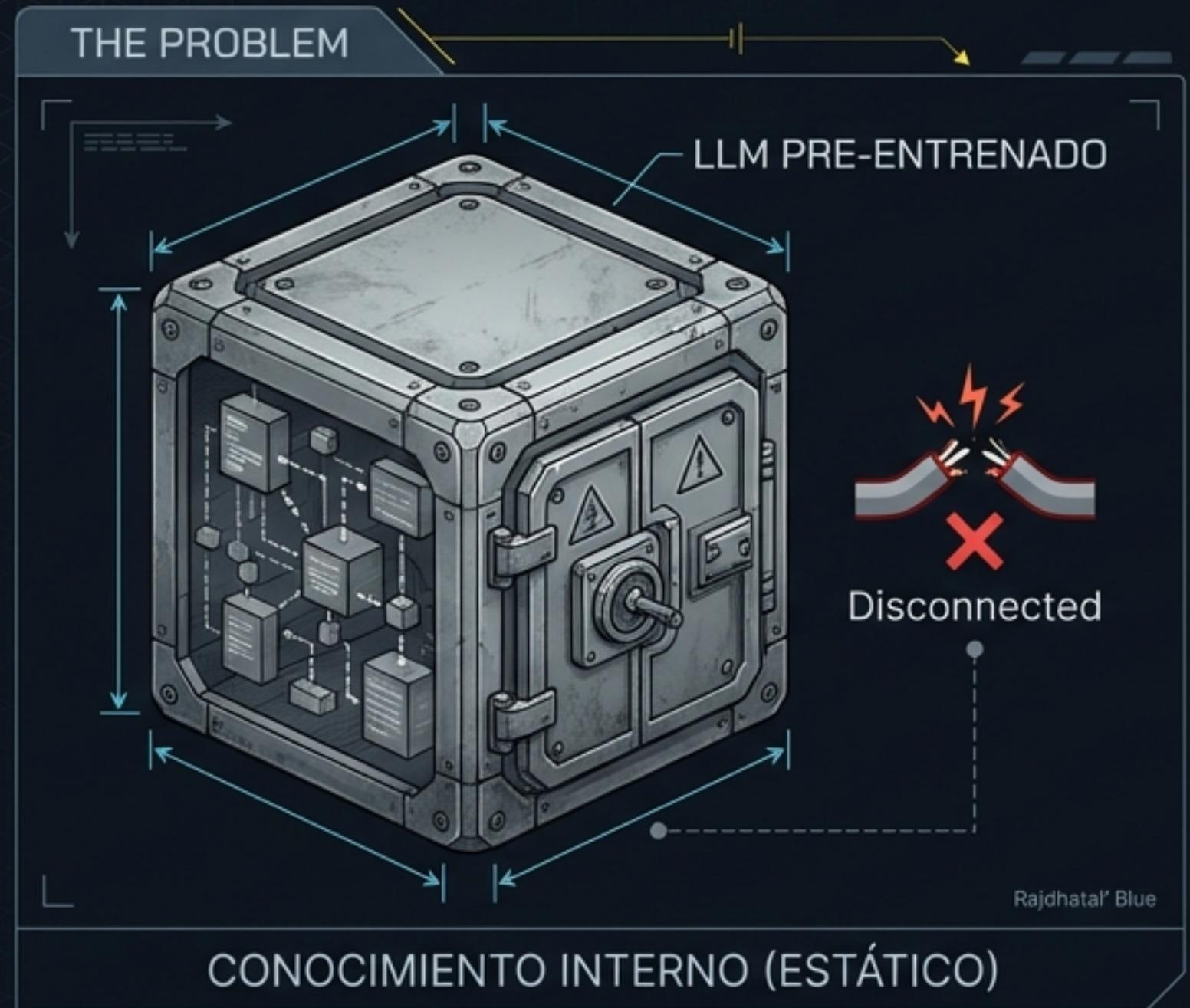


ARQUITECTURA RAG Y MEMORIA VECTORIAL CON GEMINI

EL PLANO DEL ARQUITECTO:
DE LA TEORÍA A LA PRODUCCIÓN



EL DESAFÍO DE LA MEMORIA ESTÁTICA

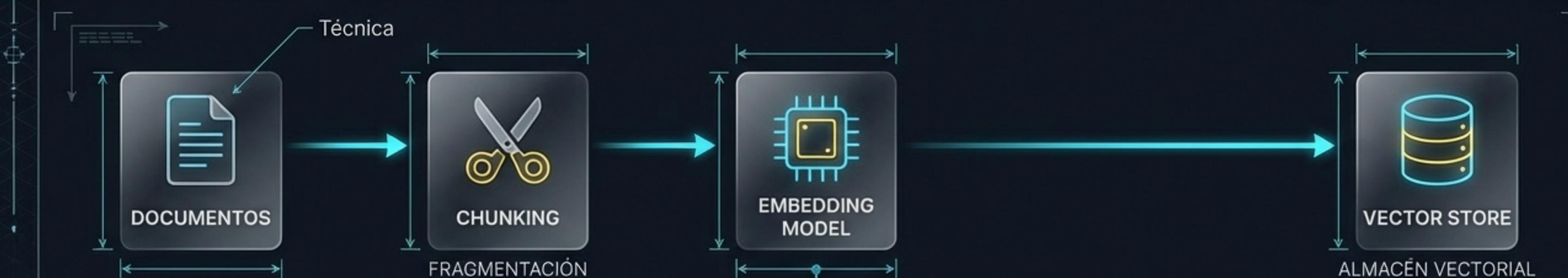


Retrieval-Augmented Generation (RAG) conecta la capacidad generativa de los LLMs con fuentes externas de información actualizada.

ANATOMÍA DEL PIPELINE RAG

JetBrains Mono

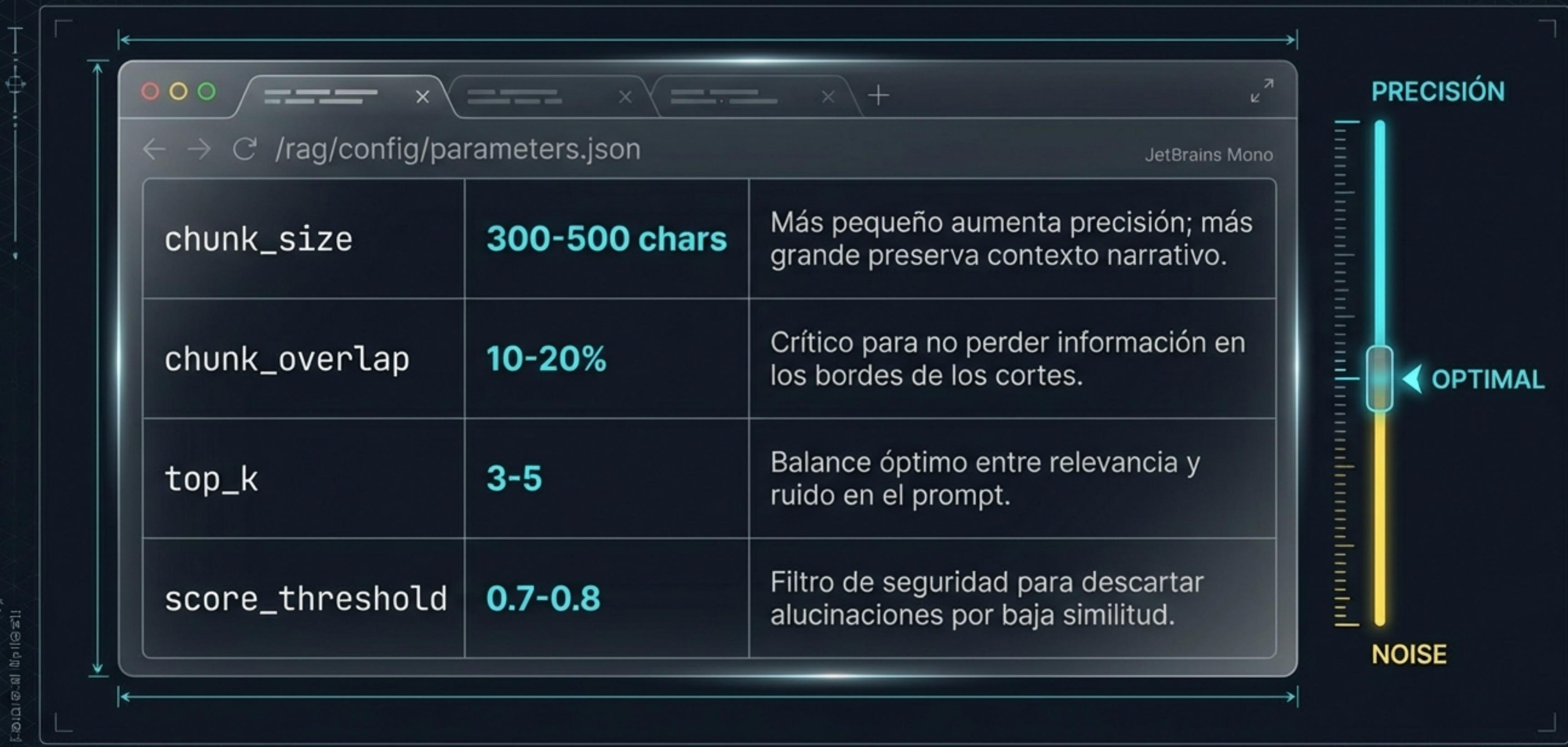
CARRIL 1: INDEXACIÓN (PREPARACIÓN)



CARRIL 2: RETRIEVAL (EJECUCIÓN)



INGENIERÍA DE PRECISIÓN: PARÁMETROS DE CONFIGURACIÓN



EL ÁTOMO SEMÁNTICO: ENTENDIENDO LOS EMBEDDINGS

Representaciones vectoriales densas donde la proximidad matemática equivale a similitud de significado.



SELECCIÓN DE HERRAMIENTAS: MODELOS GEMINI

text-embedding-004

RECOMENDADO

↗ DIMENSIONES: 768

★ CALIDAD: ESTADO DEL ARTE

⚙ USO: PROPÓSITO GENERAL

embedding-001

LEGACY

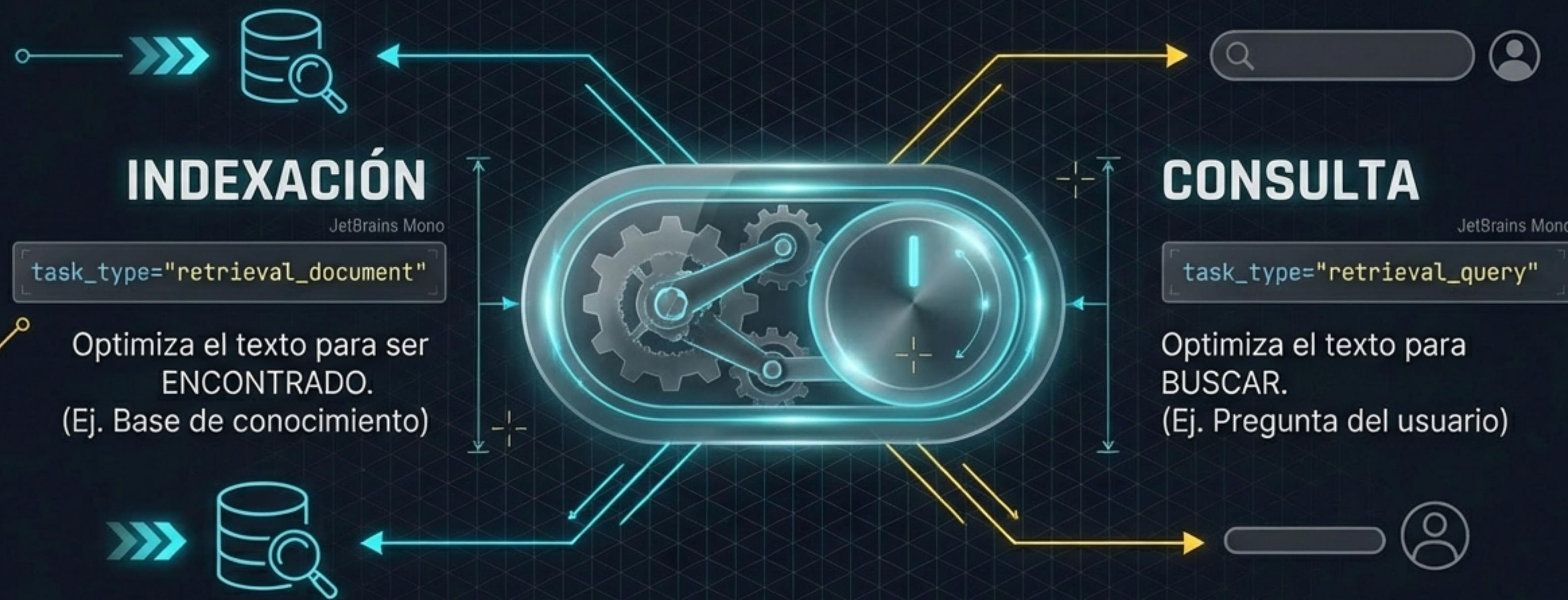
↗ DIMENSIONES: 768

CPU CALIDAD: ESTÁNDAR

⬅ USO: COMPATIBILIDAD RETROACTIVA

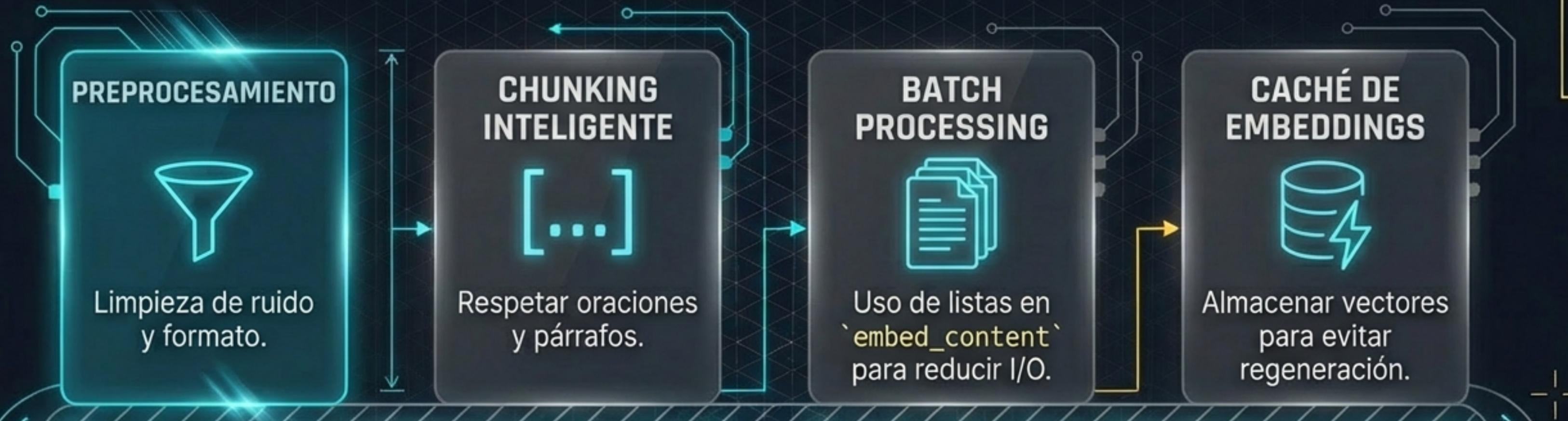
⚠ La consistencia en las dimensiones (768) es crucial para la arquitectura de la base de datos.

TASK TYPES: OPTIMIZANDO LA INTENCIÓN



ADVERTENCIA: Mezclar tipos de tareas resulta en una degradación significativa de la relevancia (baja similitud).

ESTRATEGIAS DE OPTIMIZACIÓN DE RECURSOS



**STATUS: STABLE.
MEMORY: OPTIMIZED.**

DIAGNÓSTICO Y SOLUCIÓN DE FALLOS

SÍNTOMA

Baja relevancia en búsqueda

Alta latencia del sistema

Error en textos largos

CAUSA PROBABLE

task_type incorrecto

Regeneración constante

Límite de tokens excedido

ACCIÓN CORRECTIVA

Verificar "retrieval_query" vs "retrieval_document"



Implementar capa de Caché (Redis/Local)

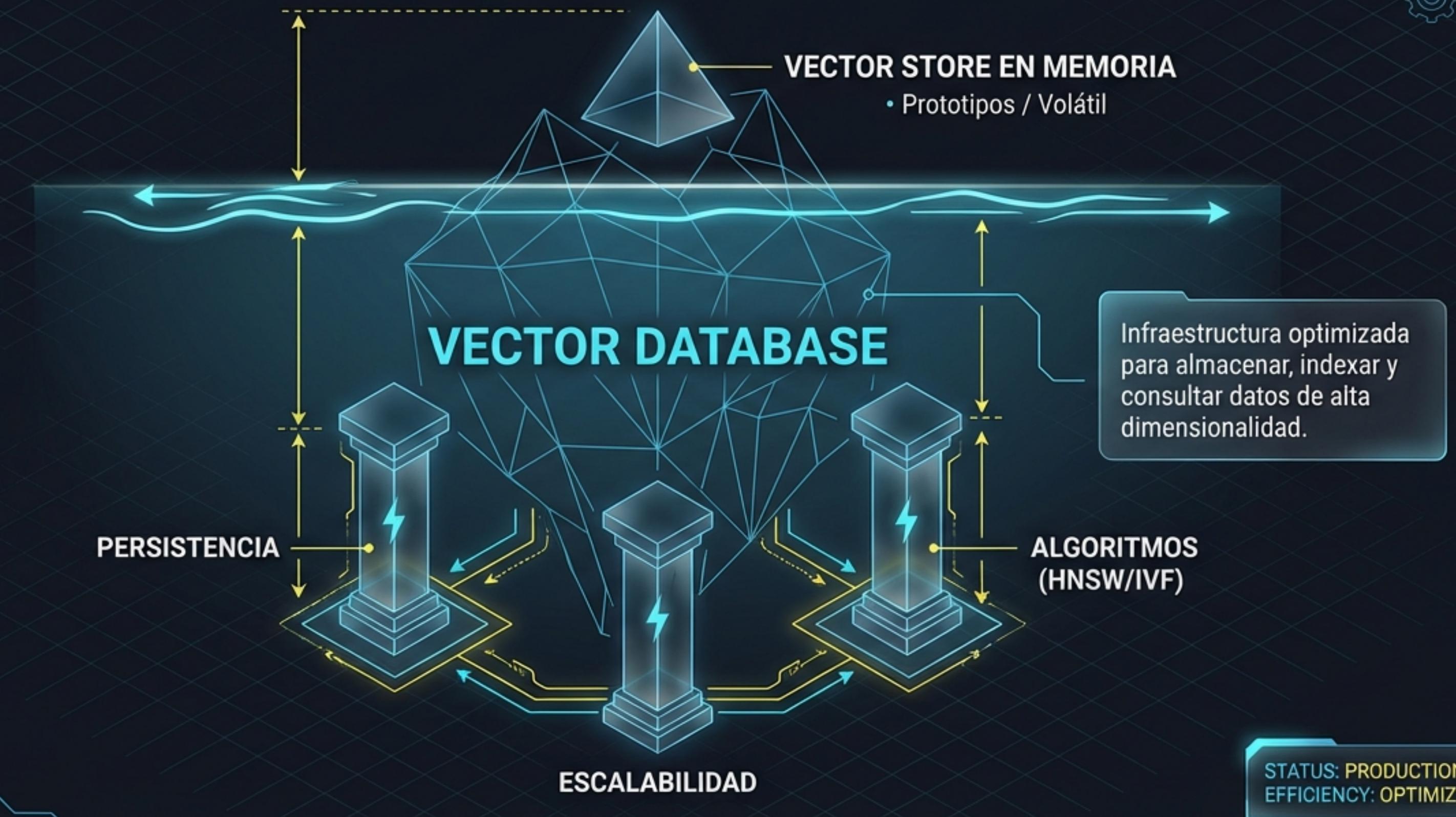


Refinar estrategia de Chunking (< 2048 tokens)

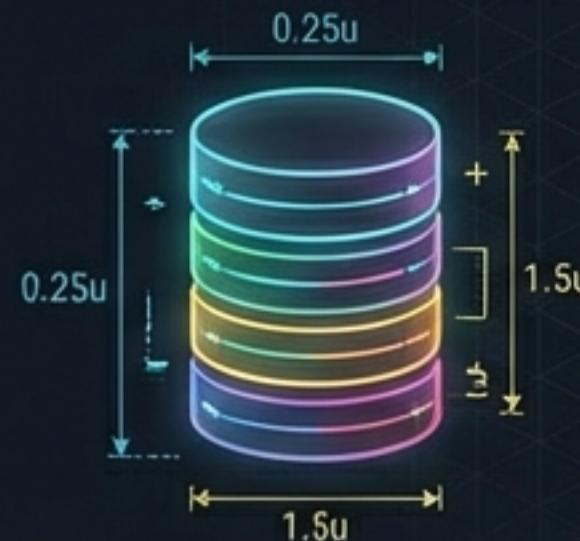


STATUS: DIAGNOSING.
EFFICIENCY: IMPROVING.

DE LA MEMORIA A LA PRODUCCIÓN



PANORAMA DE BASES DE DATOS VECTORIALES



JetBrains Mono
PERFIL:
Desarrollo / Local

NATURALEZA:
Embebida (SQLite)



JetBrains Mono
PERFIL:
Producción / Enterprise

NATURALEZA:
Cloud SaaS (Managed)



JetBrains Mono
PERFIL:
Híbrido / Flexible

NATURALEZA:
Open Source / GraphQL

STATUS: ANALYSIS.
EFFICIENCY: OPTIMAL.



CHROMADB: SIMPLICIDAD Y RAPIDEZ

```
pip install chromadb  
  
collection.query(  
    query_texts=["laptop"],  
    where={"price": {"$lt": 1000}}  
)
```

Instalación en un paso: Sin infraestructura compleja.

Filtrado Avanzado: Uso de metadatos (`\$where`, `\$contains`) antes de la búsqueda vectorial.

Uso Ideal: Validación de conceptos y Apps locales.

ESCALA Y FLEXIBILIDAD: PINECONE Y WEAVIATE



La Nube
PINECONE

- **Namespaces:** Aislamiento lógico (Multi-tenant).
- **Ventaja:** Cero mantenimiento de infraestructura.



La Potencia
WEAVIATE

- **Módulos ML:** Integración nativa de modelos.
- **Ventaja:** Control total y consultas GraphQL.

STATUS: COMPARISON. EFFICIENCY: SCALABLE.

MATRIZ DE DECISIÓN Y CHECKLIST DE DESPLIEGUE

DB	COSTO	ESCALA
Chroma	Gratis	Media
Weaviate	Self-hosted	Alta
Pinecone	Pay-as-you-go	Muy Alta

CHECKLIST DE DESPLIEGUE

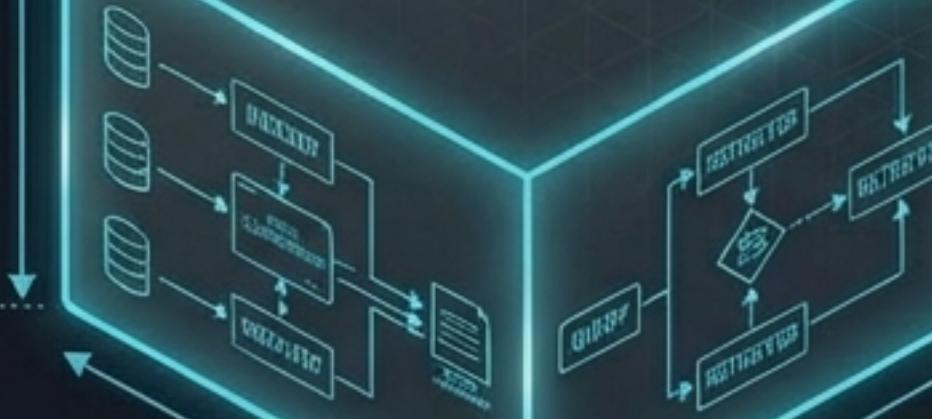
- Elegir DB según presupuesto.
- Configurar índices con métrica cosine.
- Implementar manejo de errores (Retries).
- Definir estrategia de Namespaces.
- Activar Logging y monitoreo.

STATUS: DECISION_MATRIX_DEPLOYMENT_PREP. EFFICIENCY: OPTIMIZED.

RESUMEN DEL PROYECTO

ARQUITECTURA RAG

Flujos de Indexación y Retrieval



EMBEDDINGS

Calidad semántica con Gemini



VECTOR DBs

Persistencia y escala



PRÓXIMO MÓDULO: HYBRID SEARCH

Combinando búsqueda semántica
con la precisión de keywords.

STATUS: PROJECT_SUMMARY. EFFICIENCY: FOUNDATION_BUILD. FUTURE: HYBRID_SEARCH.