ANH
AGENCIA NACIONAL DE HIDROCARBUROS

Contratar el análisis, diseño, implementación y puesta en funcionamiento del Banco de Información Energética (BIEN(Banco de Información Energética)) que contemple la gobernanza del dato y permita la carga de información, la estandarización, el acceso y el manejo integral y sistemático de la información gestionada por la Vicepresidencia Técnica de la ANH(Agencia Nacional de Hidrocarburos).



Informe de Base de Datos e Indexación agente BIEN IA

Bogotá, Colombia

DOCUMENTO CONFIDENCIAL

UT ANH PWD – NX 2025

Unión Temporal ANH PWD-NX 2025

Historial de versiones

Id	Fecha (dd/mm/aa)	Versión	Descripción	Elaborado por	Revisado por
1	23/01/2026	1.0	Versión inicial del documento	UT ANH PWD-NX 2025	

CONFIDENCIAL

Unión Temporal ANH PWD-NX 2025

Contenido

Introducción 5
1. Alcance del Documento 6
1.1 Visión Integrada del Ecosistema de Almacenamiento 7
2. Marco Conceptual: Bases de Datos para RAG 8
2.1 Taxonomía de Bases de Datos 9
2.2 Comparación de Paradigmas de Almacenamiento 10
2.3 Patrones de Integración para RAG 11
3. Alineación con Requerimientos Contractuales 12
3.1 Mapeo de Requerimientos a Componentes de Base de Datos 12
4. Arquitectura de Almacenamiento del BIEN 14
4.1 Arquitectura Física de Almacenamiento 14
4.2 Arquitectura Lógica de Datos 16
4.3 Flujo de Datos entre Componentes 17
5. Base de Datos Vectorial: Milvus + cuVS 18
5.1 Arquitectura de Milvus 18
5.2 Configuración de Colecciones para BIEN 20
5.3 Aceleración GPU con cuVS 21
5.4 Configuración de Índices Vectoriales 22
5.5 Schema de Colección Principal 23
6. Estrategias de Indexación 23
6.1 Pipeline de Indexación de Documentos 23
6.2 Configuración de Chunking por Tipo de Documento 25
6.3 Métricas de Rendimiento de Indexación 25
7. Conclusiones y Recomendaciones 26
7.1 Resumen de Capacidades Implementadas 26
7.2 Alineación con NVIDIA RAG Blueprint 27
7.3 Recomendaciones 27

CONFIDENCIAL

8.lReferencias 28
Recursos Principales 28

CONFIDENCIAL

Unión Temporal ANH PWD-NX 2025

INTRODUCCIÓN

El Banco de Información Energética (BIEN) es una iniciativa estratégica de la Agencia Nacional de Hidrocarburos (ANH) de Colombia que busca centralizar, organizar y democratizar el acceso a la información técnica del sector energético nacional. Este repositorio digital los datos de exploración y producción de hidrocarburos en Colombia, incluyendo informes geológicos, registros de pozos, datos sísmicos, estudios petrofísicos y documentación contractual.

La gestión eficiente de este volumen de información requiere una arquitectura de bases de datos híbrida que combine lo mejor de los paradigmas relacional y no relacional. Las bases de datos relacionales aportan consistencia ACID, integridad referencial y capacidades de consulta SQL para metadatos estructurados. Las bases de datos no relacionales—particularmente las bases de datos vectoriales—habilitan la búsqueda semántica mediante embeddings de alta dimensionalidad, fundamental para el sistema RAG del Agente BIEN.

La implementación propuesta, alineada con el NVIDIA RAG Blueprint v2.3.0, integra Milvus como base de datos vectorial acelerada por GPU mediante cuVS, Elasticsearch para búsqueda full-text, MinIO para almacenamiento de objetos y Redis para caché de alta velocidad. Adicionalmente, PostgreSQL con la extensión PostGIS se implementa como un agente externo de consultas GIS que puede ser invocado para enriquecer respuestas con información geoespacial sin formar parte del núcleo de la arquitectura RAG. Esta combinación proporciona el rendimiento, escalabilidad y flexibilidad necesarios para soportar consultas en

CONFIDENCIAL

tiempo real sobre el corpus documental del BIEN, con latencias objetivo-inferiores a 1.5 segundos para el 95% de las consultas.

El presente informe detalla la arquitectura, configuración y optimización de cada componente de almacenamiento de datos, alineando las decisiones técnicas con los requerimientos contractuales y las capacidades del servidor Dell PowerEdge R760xa con 4×H100 NVL disponible para el proyecto.

1. Alcance del Documento

El presente informe detalla la implementación de bases de datos relacionales y no relacionales para la indexación eficiente de datos técnicos y documentales del proyecto BIEN. Esta implementación responde directamente a los requerimientos del Anexo B del contrato, que especifica la necesidad de "funcionalidades de búsquedas avanzadas y rápidas, incluyendo capacidades de búsqueda full-text" y "priorización de resultados basada en criterios de relevancia".

La arquitectura propuesta implementa un ecosistema de almacenamiento de datos alineado con el NVIDIA RAG Blueprint v2.3.0, compuesto por cuatro componentes internos especializados: Milvus para búsqueda vectorial semántica con aceleración cuVS, Elasticsearch para búsqueda full-text, MinIO para almacenamiento de objetos, y Redis para caché de alta velocidad. Complementariamente, PostgreSQL + PostGIS se implementa como un agente externo de GIS que puede ser consultado por el sistema para enriquecer

CONFIDENCIAL

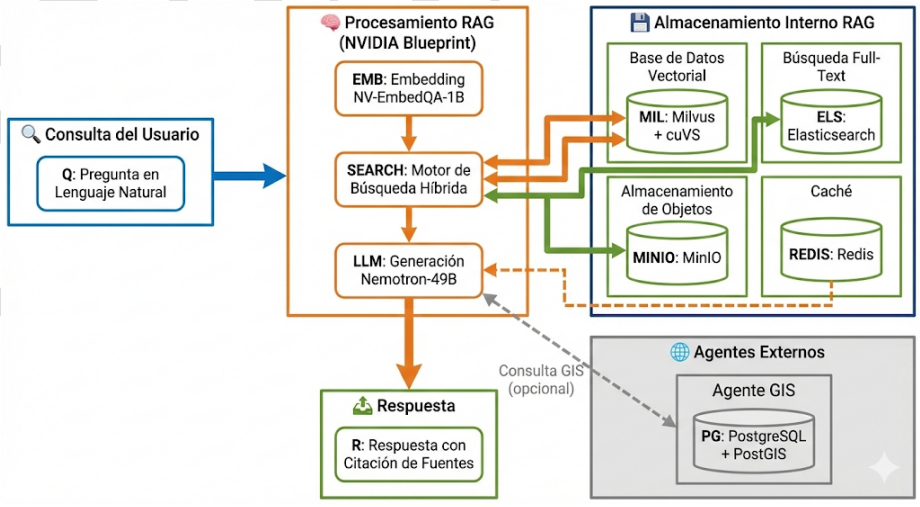
respuestas con datos geoespaciales y metadatos estructurados del dominio de hidrocarburos.

Nota Arquitectónica - Agente Externo GIS: Siguiendo el NVIDIA RAG Blueprint v2.3.0, PostgreSQL + PostGIS no forma parte del pipeline de retrieval interno del sistema RAG. Se implementa como un agente externo especializado que el LLM puede consultar opcionalmente para enriquecer respuestas con información geoespacial del dominio de hidrocarburos (coordenadas de pozos, cuencas, bloques exploratorios).

1.1 Visión Integrada del Ecosistema de Almacenamiento

El siguiente diagrama presenta la visión de alto nivel del ecosistema de almacenamiento del Agente BIEN, mostrando la interacción entre los diferentes tipos de bases de datos y su rol en el flujo de procesamiento RAG.

Diagrama 1.1: Ecosistema de Almacenamiento del Agente BIEN



El ecosistema implementa una arquitectura de almacenamiento alineada con el NVIDIA RAG Blueprint, donde el núcleo del sistema está compuesto por cuatro componentes internos optimizados: Milvus maneja la búsqueda vectorial por similitud semántica con aceleración GPU cuVS, Elasticsearch proporciona búsqueda full-text para términos exactos, MinIO almacena los documentos originales para citación, y Redis acelera operaciones frecuentes mediante caché. PostgreSQL + PostGIS opera como un agente externo de GIS que el LLM puede consultar opcionalmente para enriquecer respuestas con información geoespacial (coordenadas de pozos, cuencas, ubicaciones de yacimientos) sin formar parte del pipeline de retrieval principal. Esta arquitectura sigue el principio de separación entre almacenamiento interno RAG y servicios externos especializados.

2. Marco Conceptual: Bases de Datos para RAG

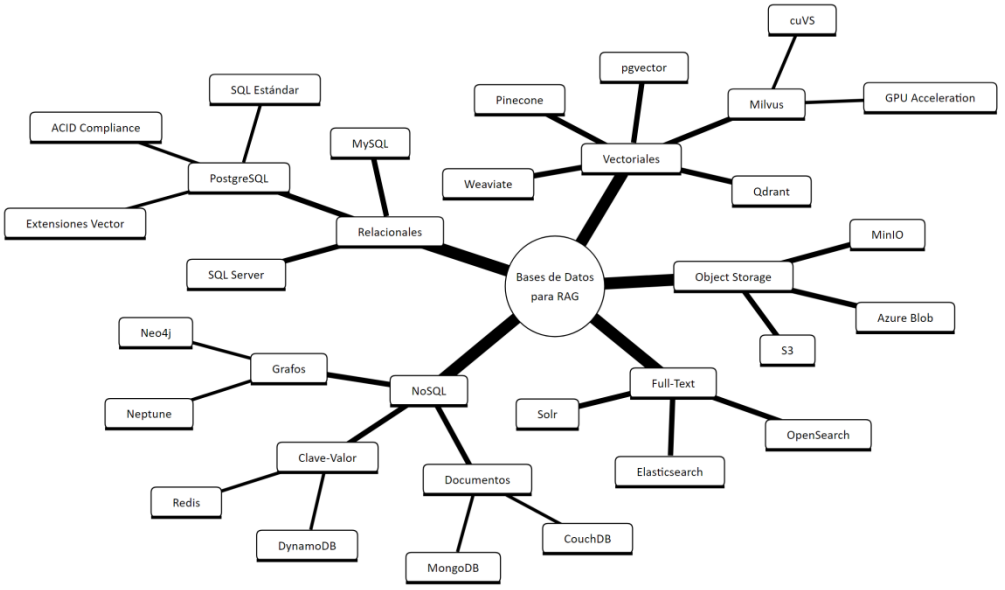
La implementación de un sistema RAG empresarial requiere comprender las características distintivas de cada tipo de base de datos y cómo se complementan para formar un ecosistema de almacenamiento coherente. Esta sección establece los fundamentos conceptuales que guían las decisiones arquitectónicas del proyecto BIEN.

CONFIDENCIAL

2.1 Taxonomía de Bases de Datos

El universo de sistemas de gestión de bases de datos se ha diversificado significativamente para atender requerimientos especializados. El siguiente diagrama clasifica los tipos de bases de datos relevantes para sistemas RAG.

Diagrama 2.1: Taxonomía de Bases de Datos para Sistemas RAG



La taxonomía revela la especialización creciente del mercado de bases de datos. Para el proyecto BIEN, la selección prioriza soluciones open-source con soporte empresarial, aceleración GPU cuando disponible, y compatibilidad con el stack NVIDIA RAG Blueprint. Milvus destaca como base de datos vectorial por su integración nativa con cuVS para aceleración GPU en las H100, mientras que PostgreSQL se mantiene como estándar de facto para datos relacionales en ambientes Linux.

2.2 Comparación de Paradigmas de Almacenamiento

La siguiente tabla contrasta las características fundamentales de cada paradigma de almacenamiento, facilitando la comprensión de por qué el proyecto BIEN requiere múltiples sistemas complementarios.

Tabla 2.1: Comparación de Paradigmas de Almacenamiento

Característica	Vectorial (Milvus)	Objetos (MinIO)	Caché (Redis)	Relacional (PostgreSQL)*
Modelo de Datos	Vectores N-dimensionales	Blobs binarios	Key-Value	Tablas con esquema fijo
Tipo de Consulta	k-NN por similitud	Get/Put por clave	Get/Set	SQL con JOINs
Consistencia	Eventual	Strong	Eventual	ACID completo
Escalabilidad	Horizontal sharding	Horizontal	Replicación	Vertical + Replicación
Latencia Típica	10-50 ms	5-50 ms	<1 ms	1-10 ms
Rol en BIEN	Búsqueda semántica	Documentos originales	Caché queries	Agente GIS externo

CONFIDENCIAL

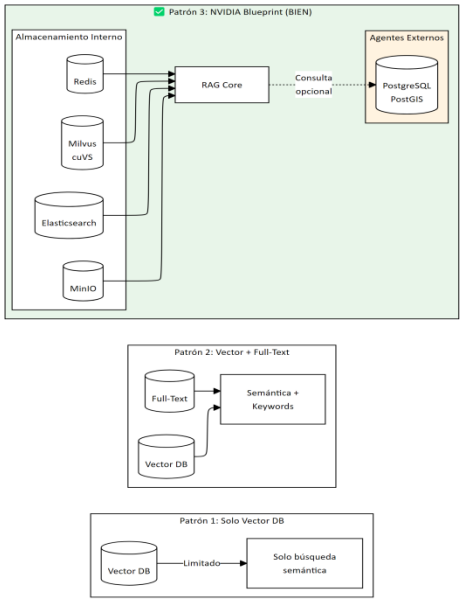
*Nota: PostgreSQL + PostGIS no forma parte del pipeline RAG interno según el NVIDIA RAG Blueprint. Se implementa como un agente externo de consultas GIS que puede ser invocado opcionalmente para obtener información geoespacial.

Los componentes internos del RAG (Milvus, Elasticsearch, MinIO, Redis) trabajan en conjunto para el flujo de retrieval y generación. PostgreSQL opera de forma independiente como servicio de consultas geoespaciales que enriquece respuestas cuando es necesario.

2.3 Patrones de Integración para RAG

El siguiente diagrama ilustra los patrones de integración más comunes entre bases de datos en arquitecturas RAG, destacando el patrón implementado para el BIEN.

Diagrama 2.2: Patrones de Integración de Bases de Datos para RAG



CONFIDENCIAL

El Patrón 3 (NVIDIA RAG Blueprint) representa la arquitectura recomendada para sistemas RAG empresariales. El núcleo del sistema utiliza Milvus+cuVS para búsqueda vectorial, Elasticsearch para full-text, MinIO para objetos y Redis para caché. PostgreSQL + PostGIS opera como un agente externo que puede ser consultado opcionalmente para enriquecer respuestas con información geoespacial, sin participar en el pipeline de retrieval principal.

3. Alineación con Requerimientos Contractuales

3.1 Mapeo de Requerimientos a Componentes de Base de Datos

El Anexo B del contrato ANH-002-CM-2025 establece requisitos específicos relacionados con almacenamiento e indexación de datos. La siguiente tabla presenta el mapeo detallado entre cada requerimiento contractual y el componente de base de datos que lo implementa.

Tabla 3.1: Trazabilidad Requerimientos vs Componentes de Almacenamiento

Requerimiento Contractual	Sección	Componente Implementador	Justificación Técnica
"Funcionalidades de búsquedas avanzadas y rápidas"	Pág. 9	Milvus + cuVS	Búsqueda vectorial GPU-acelerada con latencia <50ms

CONFIDENCIAL

Unión Temporal ANH PWD-NX 2025

Requerimiento Contractual	Sección	Componente Implementador	Justificación Técnica
"Capacidades de búsqueda full-text"	Pág. 9	Elasticsearch	Índices invertidos optimizados para texto
"Priorización de resultados basada en criterios de relevancia"	Pág. 9	Milvus + NV-RerankQA	Reranking por similitud coseno + cross-encoder
"Identificación de información repetida"	Pág. 5	Milvus	Detección de duplicados por similitud vectorial >0.95
"Almacenamiento seguro de datos energéticos"	Pág. 12	MinIO (interno) + Elasticsearch	Almacenamiento de objetos con encriptación en reposo
"Trazabilidad y auditoría"	Pág. 14	Elasticsearch + Logs	Índice de auditoría con timestamps y usuarios
"Escalabilidad para crecimiento del corpus"	Pág. 8	Milvus (sharding)	Arquitectura distribuida horizontal

CONFIDENCIAL

Requerimiento Contractual	Sección	Componente Implementador	Justificación Técnica
"Consultas geoespaciales de pozos/cuencas"	Pág. 10	PostgreSQL + PostGIS (externo)	Agente GIS externo para enriquecimiento de respuestas

La matriz de trazabilidad demuestra cobertura completa de los requerimientos contractuales mediante la arquitectura alineada con NVIDIA RAG Blueprint. Los componentes internos (Milvus, Elasticsearch, MinIO, Redis) satisfacen los requerimientos de búsqueda y almacenamiento, mientras que PostgreSQL + PostGIS opera como agente externo para consultas geoespaciales especializadas del dominio de hidrocarburos.

4. Arquitectura de Almacenamiento del BIEN

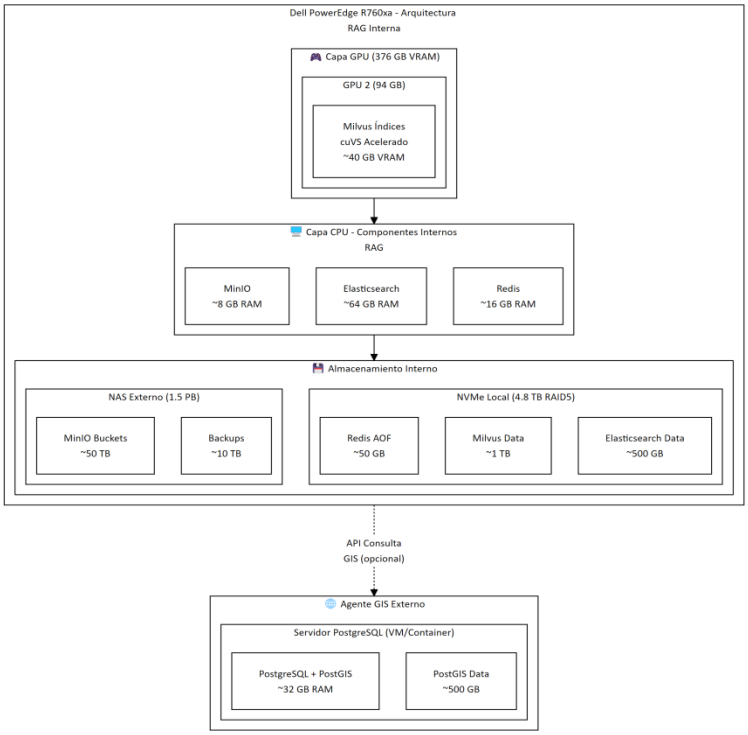
La arquitectura de almacenamiento del proyecto BIEN implementa un diseño en capas que separa las responsabilidades de acceso, procesamiento y persistencia. Esta sección detalla la configuración física y lógica del ecosistema de bases de datos.

4.1 Arquitectura Física de Almacenamiento

El siguiente diagrama presenta la distribución física de los componentes de almacenamiento en el servidor Dell PowerEdge R760xa, mostrando la asignación de recursos GPU, CPU y almacenamiento.

CONFIDENCIAL

Diagrama 4.1: Arquitectura Física de Almacenamiento (NVIDIA RAG Blueprint)

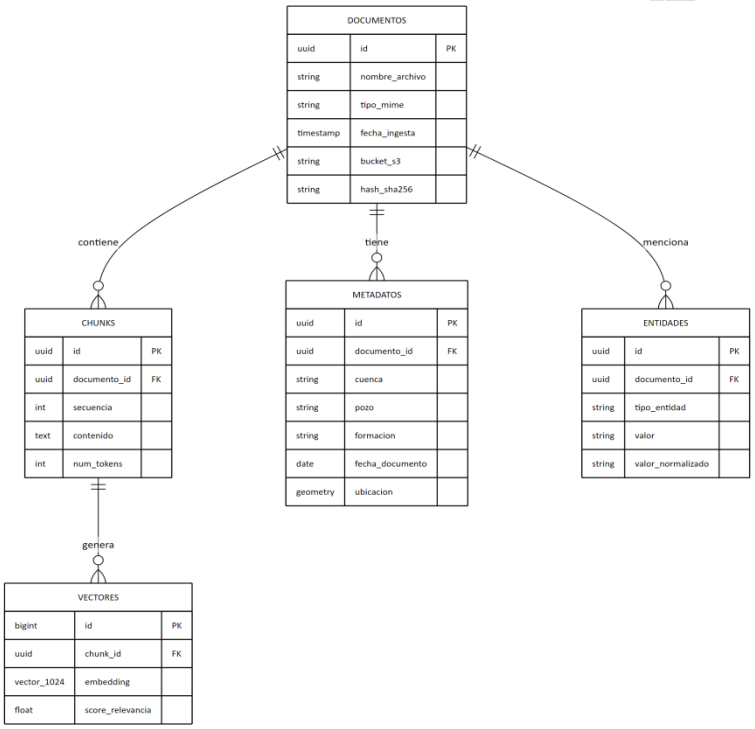


La arquitectura física sigue el NVIDIA RAG Blueprint, separando el núcleo RAG del servidor Dell PowerEdge de los servicios externos. Los componentes internos (Milvus+cuVS, Elasticsearch, MinIO, Redis) residen en el servidor principal, mientras que PostgreSQL + PostGIS opera como un agente GIS externo que puede ejecutarse en una VM o contenedor separado. Esta separación permite escalar el servicio GIS independientemente y mantiene la arquitectura RAG alineada con el blueprint de NVIDIA.

4.2 Arquitectura Lógica de Datos

La organización lógica de datos separa el contenido por dominio funcional y tipo de acceso. El siguiente diagrama muestra las colecciones, tablas e índices principales del sistema.

Diagrama 4.2: Modelo Lógico de Datos del BIEN

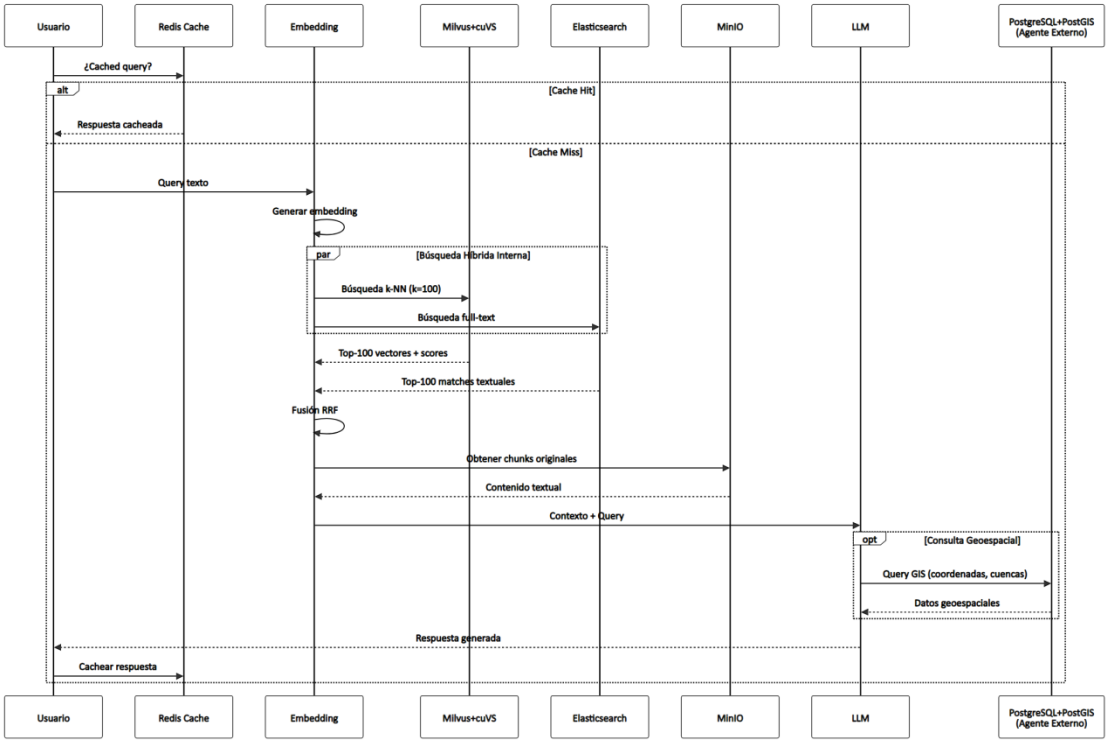


El modelo lógico establece relaciones claras entre documentos, sus fragmentos (chunks), las representaciones vectoriales y los metadatos estructurados. Esta normalización permite consultas eficientes por cualquier dimensión: búsqueda semántica sobre vectores, filtrado por metadatos geográficos/temporales, o navegación por entidades extraídas (pozos, cuencas, formaciones).

4.3 Flujo de Datos entre Componentes

El siguiente diagrama de secuencia ilustra el flujo de datos durante una consulta típica, mostrando la interacción entre todos los componentes de almacenamiento.

Diagrama 4.3: Flujo de Datos en Consulta RAG (NVIDIA Blueprint)



El flujo ilustra la arquitectura NVIDIA RAG Blueprint: Redis intercepta consultas repetidas, las nuevas activan búsqueda híbrida paralela en Milvus (semántica) y Elasticsearch (textual) como componentes internos, fusión RRF, recuperación de contenido desde MinIO, y generación con el LLM. La consulta al agente GIS externo (PostgreSQL+PostGIS) es opcional y ocurre solo cuando el LLM determina que se requiere información geoespacial para enriquecer la respuesta.

5. Base de Datos Vectorial: Milvus + cuVS

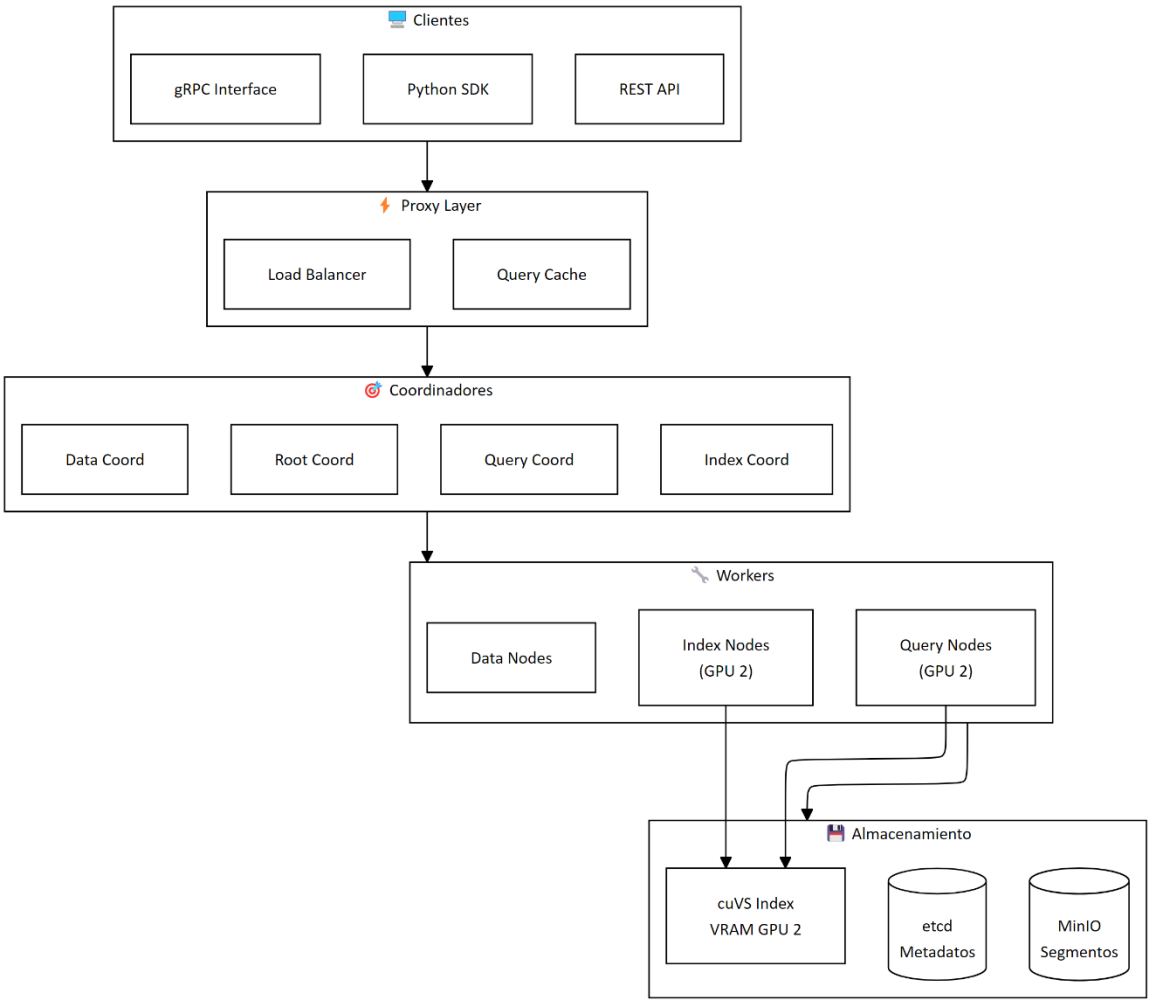
La base de datos vectorial Milvus constituye el componente central de la arquitectura de búsqueda semántica del BIEN. Esta sección detalla su configuración, optimización y operación acelerada por GPU mediante NVIDIA cuVS.

5.1 Arquitectura de Milvus

Milvus implementa una arquitectura distribuida diseñada para búsqueda vectorial a escala. El siguiente diagrama presenta los componentes internos de una instancia Milvus standalone optimizada para el BIEN.

CONFIDENCIAL

Diagrama 5.1: Arquitectura Interna de Milvus para BIEN



La arquitectura de Milvus separa coordinación (metadatos, scheduling) de ejecución (query, indexación). Los Query Nodes y Index Nodes se configuran para utilizar GPU 2 del servidor, aprovechando la aceleración cuVS para operaciones vectoriales. etcd gestiona el estado del cluster mientras MinIO almacena los segmentos de datos persistentes.

5.2 Configuración de Colecciones para BIEN

La siguiente tabla especifica las colecciones de Milvus configuradas para el proyecto BIEN, cada una optimizada para un tipo específico de contenido.

Tabla 5.1: Colecciones Milvus del BIEN

Colección	Dimensiones	Índice	Métrica	Contenido	Volumen Estimado
bien_documents	1024	IVF_FLAT	Cosine	Chunks de documentos PDF/DOC	~7M vectores
bien_images	1024	IVF_FLAT	Cosine	Captions de imágenes técnicas	~200K vectores
bien_tables	1024	IVF_FLAT	Cosine	Contenido de tablas Excel	~500K vectores
bien_wells	1024	IVF_FLAT	Cosine	Registros de pozos LAS/DLIS	~300K vectores
bien_queries	1024	FLAT	Cosine	Historial de	~100K vectores

CONFIDENCIAL

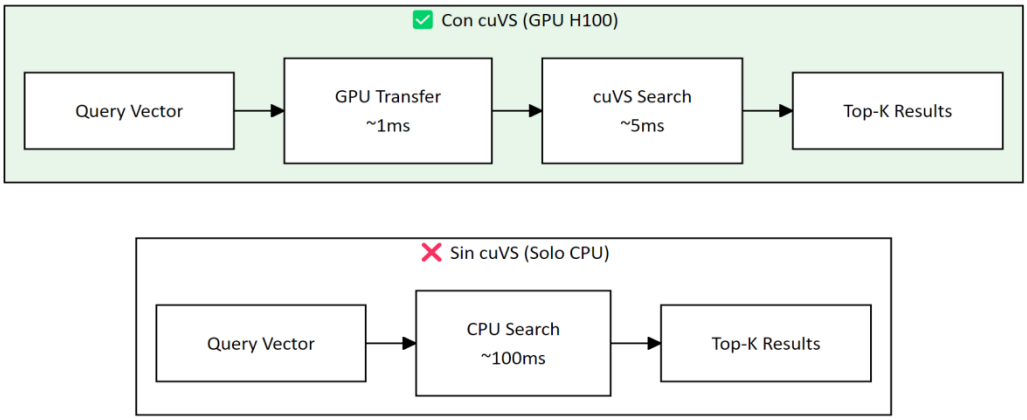
Colección	Dimensiones	Índice	Métrica	Contenido	Volumen Estimado
				consultas (caché)	

Cada colección implementa un índice IVF_FLAT con 4096 clusters (nlist), proporcionando balance óptimo entre precisión y velocidad para los volúmenes estimados. La colección de queries utiliza índice FLAT (fuerza bruta) dado su menor volumen, priorizando precisión absoluta para detección de consultas similares.

5.3 Aceleración GPU con cuVS

NVIDIA cuVS (CUDA Vector Search) proporciona aceleración de hardware para operaciones vectoriales en GPUs H100. El siguiente diagrama compara el flujo de búsqueda con y sin aceleración GPU.

Diagrama 5.2: Aceleración cuVS vs CPU



CONFIDENCIAL

La aceleración cuVS reduce la latencia de búsqueda de ~100ms (CPU) a ~6ms (GPU), un speedup de 16×. Para el BIEN, donde se esperan ~100 consultas/minuto, esta reducción es crítica para mantener la experiencia interactiva del usuario.

5.4 Configuración de Índices Vectoriales

La siguiente tabla detalla las configuraciones de índice recomendadas para diferentes escenarios de uso del BIEN.

Tabla 5.2: Configuración de Índices Vectoriales

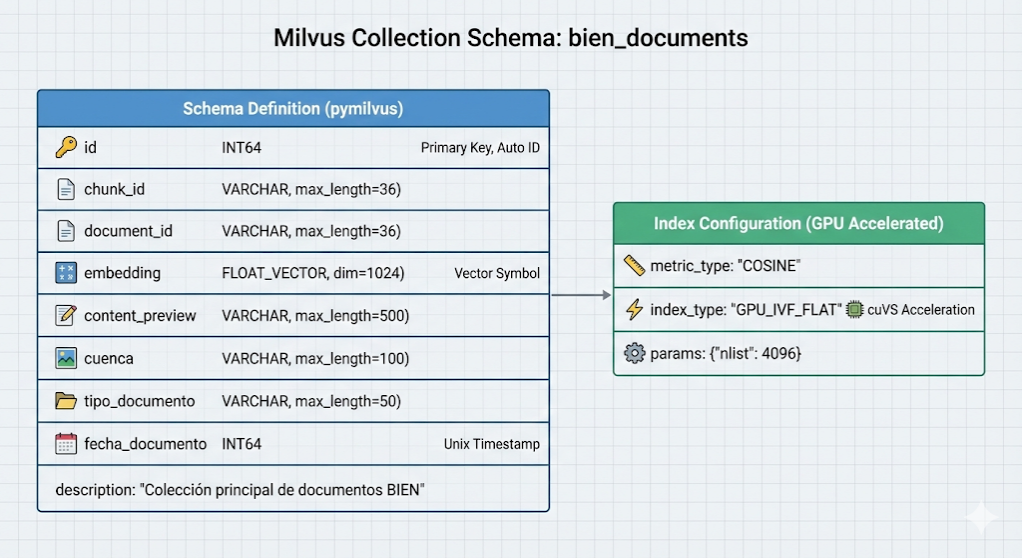
Escenario	Tipo de Índice	Parámetros	Precisión	Latencia	Memoria
Producción (Default)	IVF_FLAT	nlist=4096, nprobe=128	99.5%	~30ms	Media
Alta Precisión	FLAT	N/A	100%	~100ms	Alta
Alta Velocidad	IVF_PQ	nlist=4096, m=64, nprobe=64	97%	~10ms	Baja
Memoria Limitada	HNSW	M=16, ef=128	98%	~20ms	Media

La configuración de producción (IVF_FLAT) proporciona el balance óptimo para el BIEN: precisión cercana al 100% con latencia aceptable. El parámetro

CONFIDENCIAL

nprobe=128 explora el 3% de los clusters, suficiente para encontrar vecinos más cercanos sin escanear la totalidad del índice.

5.5 Schema de Colección Principal



6. Estrategias de Indexación

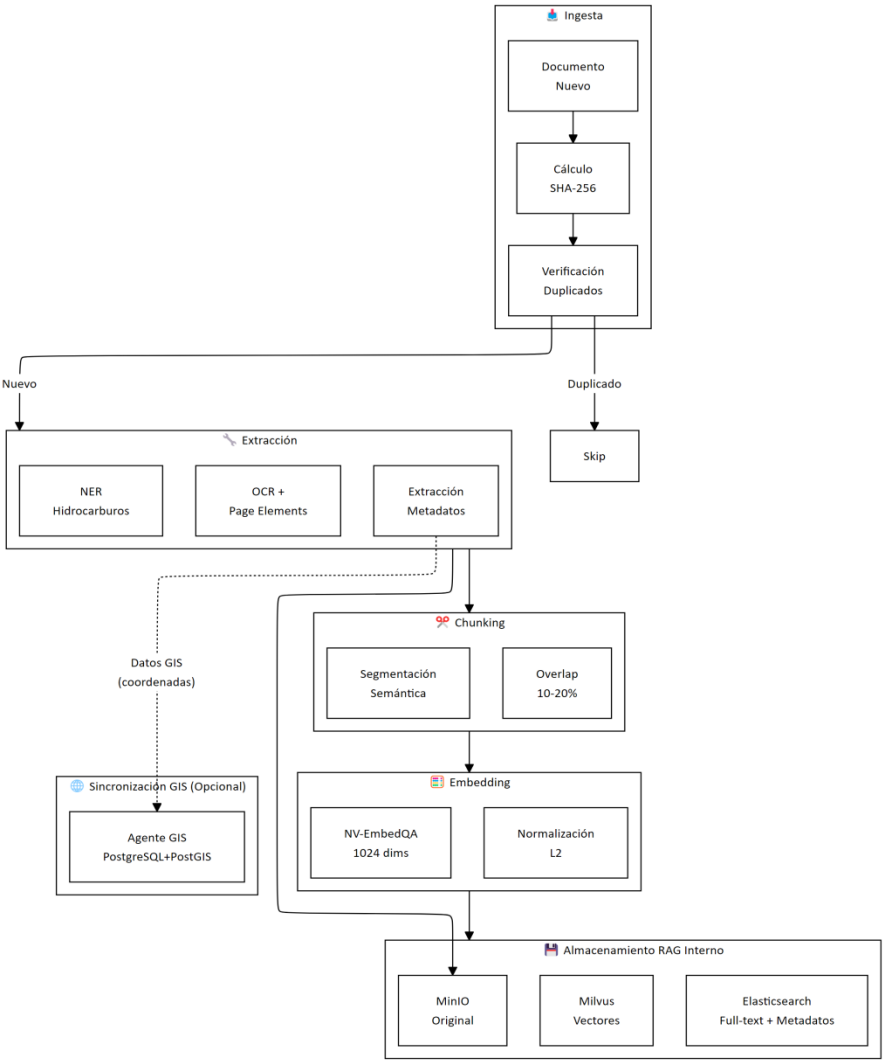
La indexación eficiente es fundamental para el rendimiento del sistema. Esta sección detalla las estrategias implementadas para cada tipo de dato.

6.1 Pipeline de Indexación de Documentos

El siguiente diagrama presenta el pipeline completo de indexación para documentos nuevos.

Diagrama 6.1: Pipeline de Indexación de Documentos (NVIDIA Blueprint)

Unión Temporal ANH PWD-NX 2025



El pipeline implementa un flujo de procesamiento end-to-end: verificación de duplicados mediante hash SHA-256, extracción de contenido con OCR y NER especializado para hidrocarburos, segmentación semántica en chunks con overlap, generación de embeddings con NV-EmbedQA, y almacenamiento distribuido en los componentes internos RAG. Los metadatos geoespaciales (coordenadas de pozos, cuencas) pueden sincronizarse opcionalmente con el Agente GIS externo para consultas especializadas.

6.2 Configuración de Chunking por Tipo de Documento

Tabla 6.1: Configuración de Chunking

Tipo de Documento	Tamaño Chunk	Overlap	Estrategia
Informes técnicos PDF	1000 tokens	200 tokens	Semántico por secciones
Registros de pozos LAS	500 tokens	100 tokens	Por intervalos de profundidad
Hojas de cálculo Excel	Por fila/grupo	Sin overlap	Preservar estructura
Imágenes técnicas	N/A	N/A	Caption único
Contratos y legales	800 tokens	150 tokens	Por cláusulas

6.3 Métricas de Rendimiento de Indexación

Tabla 6.2: Objetivos de Métricas de Rendimiento

Métrica	Objetivo	Componente	Medición
Latencia búsqueda vectorial P50	< 30ms	Milvus + cuVS	Prometheus
Latencia búsqueda full-text P50	< 20ms	Elasticsearch	Prometheus
Throughput de indexación	> 3000 docs/hora	Pipeline completo	Logs Ingestor

CONFIDENCIAL

Métrica	Objetivo	Componente	Medición
Hit rate de caché	> 80%	Redis	Redis INFO
Disponibilidad RAG interno	> 99.5%	Milvus, ES, MinIO, Redis	Health checks
Latencia Agente GIS P95	< 50ms	PostgreSQL + PostGIS (externo)	pg_stat_statements

Nota: El Agente GIS (PostgreSQL + PostGIS) se monitorea por separado como servicio externo, con objetivos de disponibilidad independientes del pipeline RAG interno.

7. Conclusiones y Recomendaciones

7.1 Resumen de Capacidades Implementadas

La implementación de bases de datos para el proyecto BIEN, alineada con el NVIDIA RAG Blueprint v2.3.0, cumple integralmente con los requerimientos contractuales mediante una arquitectura que separa claramente los componentes internos RAG de los servicios externos:

Componentes Internos del Pipeline RAG:

1. Búsqueda Vectorial Semántica: Milvus + cuVS proporciona búsqueda por similitud con aceleración GPU y latencia <30ms.
2. Búsqueda Full-Text: Elasticsearch con analizadores técnicos para terminología de hidrocarburos.

CONFIDENCIAL

3. Almacenamiento Documental: MinIO para preservación de documentos originales con compatibilidad S3.
4. Caché de Alto Rendimiento: Redis para reducción de latencia en consultas frecuentes.

Agentes Externos Especializados:

5. Agente GIS (PostgreSQL + PostGIS): Servicio externo para consultas geoespaciales que el LLM puede invocar opcionalmente mediante function calling para enriquecer respuestas con información de ubicación de pozos, cuencas y bloques.

7.2 Alineación con NVIDIA RAG Blueprint

La arquitectura implementada sigue fielmente el NVIDIA RAG Blueprint al:

- Utilizar Milvus + cuVS como Vector Database central con aceleración GPU
- Implementar MinIO como Object Store para documentos originales
- Mantener PostgreSQL como servicio externo, no como parte del pipeline de retrieval
- Permitir escalabilidad independiente de cada componente

7.3 Recomendaciones

1. Monitoreo Continuo: Implementar dashboards Grafana con alertas para métricas críticas de los componentes internos RAG.

CONFIDENCIAL

2. Backup Automatizado: Configurar snapshots diarios de Milvus y Elasticsearch (componentes internos), con backup separado para el Agente GIS.
3. Tuning Periódico: Revisar configuración de índices cuVS trimestralmente según patrones de uso.
4. Escalabilidad RAG: Planificar migración a Milvus distributed cuando el corpus supere 50M vectores.
5. Escalabilidad GIS: El Agente GIS puede escalarse independientemente del pipeline RAG según la demanda de consultas geoespaciales.

CONFIDENCIAL

8. Referencias

Las referencias bibliográficas completas se encuentran en bibliografia.xml.

Recursos Principales

- Milvus Documentation - https://milvus.io/docs
- NVIDIA cuVS Documentation - https://docs.nvidia.com/cuvs/
- PostgreSQL Documentation - https://www.postgresql.org/docs/
- Elasticsearch Documentation - https://www.elastic.co/guide/
- MinIO Documentation - https://min.io/docs/
- Redis Documentation - https://redis.io/docs/

NVIDIA RAG Blueprint - https:/