**Nodo 3, como sabría que Recursos disponibles: ¿Qué capacidades específicas de MongoDB Vector Search, índices de texto, y filtros espaciales debes usar?**

**Source: Claude Sonnet 4**

El nodo podría "conocer" las capacidades disponibles de MongoDB y cómo tomar decisiones inteligentes al respecto.

**Entendiendo el Conocimiento del Sistema sobre sus Propias Capacidades**

El nodo de decisión necesita tener un "inventario detallado" de las capacidades de MongoDB y entender cuándo usar cada una.

**Primera Capa: Conocimiento Estático de Capacidades MongoDB**

El nodo tendría acceso a información estructurada sobre las capacidades específicas de la implementación de MongoDB. Esta información podría estar codificada directamente en el prompt o proporcionada como contexto dinámico.

**Capacidades de Vector Search Disponibles:** El sistema podría informar al nodo que tiene acceso a búsqueda vectorial sobre embeddings CLIP de dimensión 512, con capacidad para búsquedas de similaridad coseno, y que puede manejar búsquedas híbridas que combinen similaridad vectorial con filtros de metadatos. El nodo también sabría que las búsquedas vectoriales son más costosas computacionalmente pero muy efectivas para reconocimiento semántico de elementos arquitectónicos.

**Índices de Texto Configurados:** El sistema podría informar que tiene índices de texto completo configurados sobre los campos de OCR extraído, con capacidad para búsquedas de proximidad textual, expresiones regulares para patrones de medidas, y búsquedas fuzzy para manejar posibles errores en la extracción de texto. El nodo entendería que estos índices son muy rápidos para búsquedas textuales exactas pero menos efectivos para búsquedas semánticas.

**Filtros Espaciales Implementados:** El nodo sabría que el sistema puede filtrar eficientemente por metadatos como página, sección, coordenadas de bounding box, y relaciones de proximidad espacial. Entendería que estos filtros son extremadamente rápidos y efectivos para reducir el espacio de búsqueda antes de aplicar técnicas más costosas.

**Segunda Capa: Conocimiento Dinámico del Estado del Sistema**

Pero aquí viene la parte realmente inteligente. El nodo no solo conocería las capacidades teóricas, sino que también podría recibir información sobre el estado actual del sistema:

**Estadísticas de Rendimiento en Tiempo Real:** El sistema podría proporcionar información como "actualmente hay 15,847 documentos en la colección, las búsquedas vectoriales están tomando en promedio 0.8 segundos, los filtros espaciales reducen típicamente el conjunto de datos en un 95%, y las búsquedas de texto están respondiendo en 0.2 segundos promedio".

**Distribución de Datos Relevante:** El nodo podría saber que "para la página 12 específicamente, tenemos 23 segmentos almacenados, 18 de los cuales contienen texto extraído por OCR, y 15 de los cuales tienen embeddings CLIP de alta confianza". Esta información le permitiría tomar decisiones más informadas sobre qué estrategias tienen mayor probabilidad de éxito.

**Tercera Capa: Conocimiento Experiencial y Patrones de Uso**

Lo más sofisticado sería si el nodo pudiera acceder a información sobre patrones históricos de éxito:

**Patrones de Eficacia por Tipo de Consulta:** El sistema podría informar que "para consultas sobre medidas de elementos arquitectónicos como la tuya, la estrategia híbrida visual-textual con filtrado espacial inicial ha tenido una tasa de éxito del 87% en casos similares, mientras que las búsquedas puramente textuales solo alcanzan 34% de éxito debido a la variabilidad en las notaciones arquitectónicas".

**Optimizaciones Específicas del Dominio:** El nodo podría saber que "en planos arquitectónicos, los elementos visuales identificados como puertas tienen una correlación espacial del 78% con anotaciones de medidas dentro de un radio de 50 píxeles, lo que sugiere que la búsqueda de proximidad textual es altamente efectiva después de la identificación visual".

**Cómo Integrar Esta Información en el Prompt**

El prompt del Nodo 3 se enriquecería significativamente con esta información contextual. Aquí te muestro cómo se vería una versión expandida:

**Prompt Mejorado para el Nodo de Decisión:**

"Eres un arquitecto de sistemas de búsqueda multimodal especializado en planos de construcción. Tienes acceso completo a las capacidades y estado actual del sistema MongoDB.

**Capacidades Actuales Disponibles:**

* Vector Search: Embeddings CLIP-512D, similaridad coseno, tiempo promedio 0.8s, efectividad 91% para elementos arquitectónicos
* Índices de Texto: Campo OCR completo, búsqueda de proximidad, regex para patrones de medida, tiempo promedio 0.2s
* Filtros Espaciales: Metadatos de página/sección, bounding boxes, proximidad espacial, reducción promedio del dataset 95%

**Estado Actual del Sistema:**

* Total documentos: {total\_docs}
* Documentos en página {page\_number}: {page\_docs}
* Segmentos con OCR de alta confianza: {high\_conf\_ocr}
* Rendimiento promedio último día: {performance\_stats}

**Patrones de Éxito Históricos:**

* Consultas similares a esta: {success\_rate}% éxito con estrategia {recommended\_strategy}
* Elementos tipo '{target\_element}': mejor identificación via {best\_method}
* Medidas arquitectónicas: {measurement\_success\_rate}% éxito con búsqueda de proximidad

Basándote en esta información completa, diseña la estrategia óptima..."

**Implementación Práctica: Cómo el Sistema "Sabría" Todo Esto**

En la implementación real, esta información podría venir de varias fuentes:

**Configuración Estática del Sistema:** Un archivo de configuración o base de conocimiento que describe las capacidades instaladas y configuradas de MongoDB, incluyendo qué índices existen, qué tipos de búsqueda están habilitados, y qué campos están disponibles para cada tipo de operación.

**Monitoreo en Tiempo Real:** El sistema podría consultar estadísticas de MongoDB directamente antes de cada decisión, obteniendo información sobre el tamaño de las colecciones, el rendimiento reciente de diferentes tipos de consulta, y la distribución de datos relevante para la consulta específica.

**Sistema de Aprendizaje de Patrones:** Una base de datos separada que registra los resultados de consultas anteriores, permitiendo al sistema aprender qué estrategias funcionan mejor para diferentes tipos de consultas. Esto sería como un "historial de éxitos" que el sistema puede consultar para tomar mejores decisiones.

**Análisis Previo de Datos:** Durante el proceso de ingesta, el sistema podría generar metadatos estadísticos sobre los datos almacenados, como "qué porcentaje de puertas identificadas visualmente tienen anotaciones de medida asociadas" o "cuál es la distancia promedio entre elementos visuales y sus anotaciones textuales".

La belleza de este enfoque es que el sistema se vuelve progresivamente más inteligente con el uso, aprendiendo no solo qué capacidades técnicas tiene disponibles, sino también cómo usarlas de manera más efectiva para diferentes tipos de consultas arquitectónicas.

¿Te gustaría que exploremos cómo se podría implementar específicamente alguna de estas fuentes de información, o prefieres que profundicemos en cómo el nodo tomaría decisiones concretas basándose en esta información completa?