Fase 3: Estrategia de Aumentación e Integración: La Inteligencia del Tutor

En esta fase, definiremos cómo el sistema combina la información que ha recuperado con la capacidad de generar respuestas del LLM, haciendo que el tutor sea verdaderamente "inteligente" y adaptativo. Aquí es donde el RAG va más allá de una simple búsqueda y genera respuestas coherentes y pedagógicas.

3.1. Definir Métodos de Fusión: Combinando Conocimiento y Generación

La fusión es el puente entre lo que se recupera y lo que se genera. Decidiremos cómo la información del módulo recuperador (los chunks de texto y los datos del Grafo de Conocimiento) se presenta al Gran Modelo de Lenguaje (LLM) para que este pueda construir la respuesta.

Fusión Basada en Prompt (Concatenación Directa):

Método Principal: La forma más directa es insertar los chunks recuperados directamente en el prompt del LLM, junto con la consulta original del estudiante y las instrucciones específicas del tutor.

Ejemplo de Prompt: Eres un tutor inteligente experto en [Asignatura]. Responde la siguiente pregunta del estudiante de forma clara, didáctica y usando solo la información que te proporciono. Si la información no es suficiente, indícalo.

Pregunta del estudiante: "[Consulta del estudiante]"

Contexto recuperado: [Chunk 1 de texto] [Chunk 2 de texto] [Información clave del KG, si aplica]

Tu respuesta:

Fusión Avanzada (Si se requiere mayor control):

- **Fusión Latente:** Si el LLM tiene capacidades multimodales o de entendimiento de embeddings más allá de solo texto, se podrían fusionar las representaciones vectoriales del contexto con la representación de la consulta antes de la generación. Esto es más complejo y generalmente se considera en etapas de optimización.
- **Fusión Basada en Logits:** Podríamos influir en la generación del LLM en tiempo real, favoreciendo la probabilidad de tokens que estén presentes en el contexto recuperado, para asegurar una mayor fidelidad a la fuente. Esto requiere acceso más profundo a la arquitectura del LLM.

3.2. Elegir Paradigma/Patrón de RAG: La Arquitectura del Tutor

Para un tutor inteligente, un enfoque simple de RAG no es suficiente. Optaremos por una arquitectura que permita un comportamiento más complejo y adaptable, similar al de un tutor humano.

RAG Avanzada o Modular (Enfoque Base):

Esto implica ir más allá de la simple secuencia "recuperar y leer". Incorporaremos pasos adicionales:

- **Re-ranking (Reordenamiento):** Después de la recuperación inicial de chunks, un modelo más pequeño y ligero puede re-ordenar los resultados para seleccionar los más relevantes y pertinentes a la intención específica de la pregunta del estudiante. Esto mejora la calidad del contexto que llega al LLM.
- **Resumen Condensado:** Si los chunks recuperados son muy largos y exceden el límite de tokens del LLM, un módulo de resumen podría condensar la información clave antes de enviarla al generador.
- **Módulos Especializados:** Diseñaremos "sub-módulos" para tareas específicas que el tutor debe realizar:
    - *Módulo de Ejercicios:* Activado cuando el estudiante pide un ejercicio o práctica. Podría generar un ejercicio nuevo o recuperar uno de una base de datos específica.
    - *Módulo de Análisis Literario/Histórico:* Para ayudar a desglosar un texto o un evento.
    - *Módulo de Cálculo (para Matemática):* Aunque el LLM puede hacer cálculos, un módulo externo podría verificar la exactitud de los resultados o realizar operaciones complejas.

Agentic RAG (Clave para la Inteligencia del Tutor):

Este es el componente que dota al tutor de "razonamiento" y capacidad de decisión. Implica la orquestación de múltiples agentes que actúan de forma autónoma.

- **Agente de Planificación (El "Cerebro" del Tutor):**
    - Recibe la consulta del estudiante y, basándose en la intención detectada (Fase 2.1.3), decide la "estrategia" a seguir. Por ejemplo, si es una "pregunta de definición", el plan será "recuperar definición, generar respuesta". Si es "resolver ejercicio", el plan será "recuperar problema similar, recuperar solución paso a paso, guiar al estudiante".
    - Puede descomponer la tarea en sub-tareas (ej., "primero, buscar el concepto X; luego, si el estudiante no entiende, dar un ejemplo").
- **Agentes de Herramientas:** Estos agentes tienen acceso a "herramientas" específicas para ejecutar el plan:

    - *Herramienta de Búsqueda Vectorial:* Para consultar la base de datos de embeddings.
    - *Herramienta de Consulta de Grafo de Conocimiento (KG):* Para realizar consultas complejas sobre relaciones (Cypher o SPARQL).
    - *Herramienta de Generación de Ejercicios:* Si el tutor necesita crear un nuevo ejercicio de un tema específico.
    - *Herramienta de Verificación de Solución (Matemática):* Para comprobar si la respuesta numérica del estudiante es correcta.
    - *Herramienta de Búsqueda Web (para el Crawler Dinámico):* Para la fase dinámica del crawler, cuando se necesite información externa o actualizada.

- **Agente de Reflexión/Evaluación:**
    - Después de que se genera una respuesta, este agente puede evaluarla (internamente o con la ayuda de un LLM más pequeño) para ver si cumple con los criterios de calidad (precisión, pedagogía, coherencia).
    - Si la respuesta no es satisfactoria, el agente de reflexión puede decidir reintentar la generación con un prompt modificado, buscar más contexto (activando nuevamente el recuperador o incluso el crawler dinámico) o pedir aclaración al estudiante.

- **Colaboración Multi-Agente:** Diferentes agentes pueden trabajar en conjunto. Por ejemplo, un agente de historia podría recuperar datos sobre un evento, un agente de literatura podría analizar un texto, y un agente de síntesis podría combinarlos para una respuesta multidisciplinaria si la pregunta lo requiere.

3.3. Selección/Compresión de Contexto: Optimizando el Flujo de Información

El "context window" (ventana de contexto) del LLM es limitado. Es crucial enviar solo la información más relevante para evitar que el LLM se confunda o "alucine" debido a ruido en el contexto.

Estrategias de Selección:

- **Filtrado por Metadatos:** Ya establecido en la Fase 2, se utiliza para reducir la cantidad de chunks antes de la recuperación por similitud.
- **Re-ranking Avanzado:** Seleccionar solo los N mejores chunks después del re-ranking, basándose en su puntuación de relevancia combinada.
- **Diversificación de Resultados:** Asegurarse de que los chunks seleccionados no sean redundantes, pero cubran diferentes aspectos del tema relevante.

Técnicas de Compresión:

- **Compresión de Contexto basada en LLM:** Un LLM más pequeño o una técnica específica (ej., Longformers, Attention-based compressors) puede ser usada para resumir los chunks recuperados antes de pasarlos al LLM principal.
- **Extracción de Información Clave:** En lugar de pasar el chunk completo, el sistema puede extraer solo las oraciones o frases más pertinentes que respondan a la pregunta del estudiante.

Esta fase transforma la capacidad básica de RAG en una inteligencia conversacional y pedagógica para el tutor. La implementación de Agentic RAG será clave para simular un comportamiento de tutor humano, con capacidad de planificación, uso de herramientas y autoevaluación.