

Propuesta Técnica – Sistema Multiagente con RAG

Autores: Dominic Casares Aguirre c.2022085016 & Mariana Viquez Monge c.2022029468

SMART-TUTOR: Sistema Multiagente de Tutoría Académica con RAG

Descripción General:

SMART-TUTOR es un sistema multiagente inteligente diseñado para proporcionar tutoría académica personalizada en materias STEM (Ciencia, Tecnología, Ingeniería y Matemáticas). El sistema integra técnicas avanzadas de inteligencia artificial generativa, incluyendo Retrieval-Augmented Generation (RAG), para ofrecer una experiencia educativa adaptativa y contextualizada.

El sistema analiza automáticamente el nivel de conocimiento del estudiante, recupera recursos educativos relevantes mediante búsqueda semántica en una base vectorial, y genera explicaciones personalizadas que se adaptan al perfil y necesidades específicas de cada estudiante. La arquitectura multiagente permite una especialización eficiente de tareas, donde cada agente se enfoca en un aspecto específico del proceso educativo, trabajando en colaboración para lograr un resultado óptimo.

Objetivos principales:

- Evaluar automáticamente el nivel de conocimiento del estudiante
- Proporcionar recursos educativos contextualizados mediante RAG
- Generar explicaciones personalizadas y adaptativas
- Mantener un historial de aprendizaje para mejora continua
- Ofrecer una experiencia educativa interactiva y motivadora

AGENTES INVOLUCRADOS

Agente Evaluador

Especialista en evaluación educativa que analiza las consultas de los estudiantes para determinar su nivel de conocimiento, tipo de dificultad y necesidades de aprendizaje.

Prompt Utilizado:

```
Eres un Agente Evaluador experto en educación. Tu función es evaluar el nivel de conocimiento de un estudiante.
```

```
Consulta del estudiante: {consulta}
```

```
Materia: {materia}
```

Analiza la consulta y determina:

1. Nivel de conocimiento aparente (básico, intermedio, avanzado)
2. Tipo de dificultad (conceptual, procedimental, aplicación)
3. Áreas de conocimiento previo necesarias
4. Recomendaciones para el enfoque de enseñanza

Proporciona una evaluación estructurada en formato JSON con las claves:

- nivel_estimado
- tipo_dificultad
- conocimientos_previos
- recomendaciones_ensenanza
- confianza_evaluacion (0-1)

Interacciones:

- **Entrada:** Recibe la consulta del estudiante y la materia de estudio
- **Proceso:** Analiza la complejidad lingüística, términos técnicos utilizados y estructura de la pregunta
- **Salida:** Proporciona evaluación estructurada al Agente Recuperador
- **Flujo:** Evaluador → Recuperador → Tutor

Agente Recuperador RAG

Especialista en recuperación de información que utiliza técnicas de RAG para encontrar recursos educativos relevantes en la base de datos vectorial.

Prompt Implícito: Utiliza búsqueda semántica basada en embeddings para recuperar documentos relevantes. No usa prompt textual directo, sino algoritmos de similitud coseno en el espacio vectorial.

Proceso de RAG:

1. **Embedding de consulta:** Convierte la consulta del estudiante en vector numérico
2. **Búsqueda semántica:** Utiliza FAISS para encontrar documentos similares
3. **Ranking de relevancia:** Ordena resultados por similitud semántica
4. **Filtrado contextual:** Considera nivel de conocimiento y materia

Interacciones:

- **Entrada:** Recibe consulta original, materia y nivel de conocimiento del Agente Evaluador
- **Proceso:** Ejecuta búsqueda semántica en base vectorial con ~500+ documentos educativos
- **Salida:** Entrega top-3 recursos más relevantes al Agente Tutor
- **Flujo:** Evaluador → **Recuperador** → Tutor

Agente Tutor Coordinador

Tutor principal que coordina la información de los otros agentes y genera respuestas educativas personalizadas y coherentes.

Prompt Utilizado:

Eres un Tutor Académico Personalizado experto en {materia}. Tu función es proporcionar explicaciones claras y personalizadas.

INFORMACIÓN DEL ESTUDIANTE:

- Consulta: {consulta}
- Nivel estimado: {nivel_conocimiento}
- Evaluación: {evaluacion}

RECURSOS DISPONIBLES:

{recursos}

INSTRUCCIONES:

1. Proporciona una explicación clara y adaptada al nivel del estudiante
2. Usa los recursos encontrados para enriquecer tu respuesta
3. Incluye ejemplos prácticos cuando sea apropiado
4. Sugiere pasos siguientes para el aprendizaje
5. Mantén un tono amigable y motivador

Genera una respuesta educativa completa y personalizada:

Interacciones:

- **Entrada:** Recibe evaluación del Agente Evaluador y recursos del Agente Recuperador
- **Proceso:** Sintetiza información, adapta explicación según nivel, genera respuesta coherente
- **Salida:** Proporciona respuesta final personalizada al estudiante
- **Flujo:** Evaluador → Recuperador → **Tutor** → Usuario

USO DE RAG (RETRIEVAL-AUGMENTED GENERATION)

Implementación RAG en el Agente Recuperador

Base de Datos Vectorial:

- **Tecnología:** FAISS (Facebook AI Similarity Search)
- **Contenido:** 500+ documentos educativos en Álgebra Lineal, Cálculo, Probabilidad
- **Embeddings:** Ollama Embeddings con modelo Llama 3.1
- **Estructura:** Documentos segmentados con metadatos (materia, nivel, tipo)

Proceso RAG Detallado:

Preparación de Datos:

```
Document(page_content="Materia: algebra_lineal\nTítulo: Vectores\nContenido: ...", metadata={"materia": "algebra_lineal", "nivel": "basico", "tipo": "concepto"})
```

Creación de Embeddings:

```
embeddings = OllamaEmbeddings(model="llama3.1")\n\nvector_store = FAISS.from_documents(docs_divididos, embeddings)
```

Búsqueda Semántica:

```
query_busqueda = f"{consulta} {materia} nivel {nivel_conocimiento}"\n\ndocumentos_relevantes = vector_store.similarity_search(query_busqueda, k=3)
```

Ventajas del RAG:

- Encuentra recursos específicos para la consulta exacta
- Considera nivel de conocimiento y materia

- Tiene una base de conocimiento extensible sin reentrenamiento
- Ranking automático por similitud semántica

TECNOLOGÍAS UTILIZADAS

Componente	Tecnología	Versión	Función
Modelo Base	Ollama Llama 3.1	8B	Generación de texto y razonamiento
Framework LLM	LangChain	Latest	Integración y manejo de modelos
Orquestación	LangGraph	Latest	Coordinación multiagente
Base Vectorial	FAISS	CPU	Almacenamiento y búsqueda semántica
Embeddings	Ollama Embeddings	Llama 3.1	Vectorización de texto
Lenguaje	Python	3.8+	Implementación principal

Dependencias

- langchain-community 0.0.38
- langgraph 0.0.62
- faiss-cpu 1.7.4
- ollama >= 0.1.0
- bash

Justificación de Selección

Ollama + Llama 3.1:

- Ejecución local para privacidad de datos estudiantiles

- Rendimiento optimizado para tareas educativas
- Sin dependencia de APIs externas

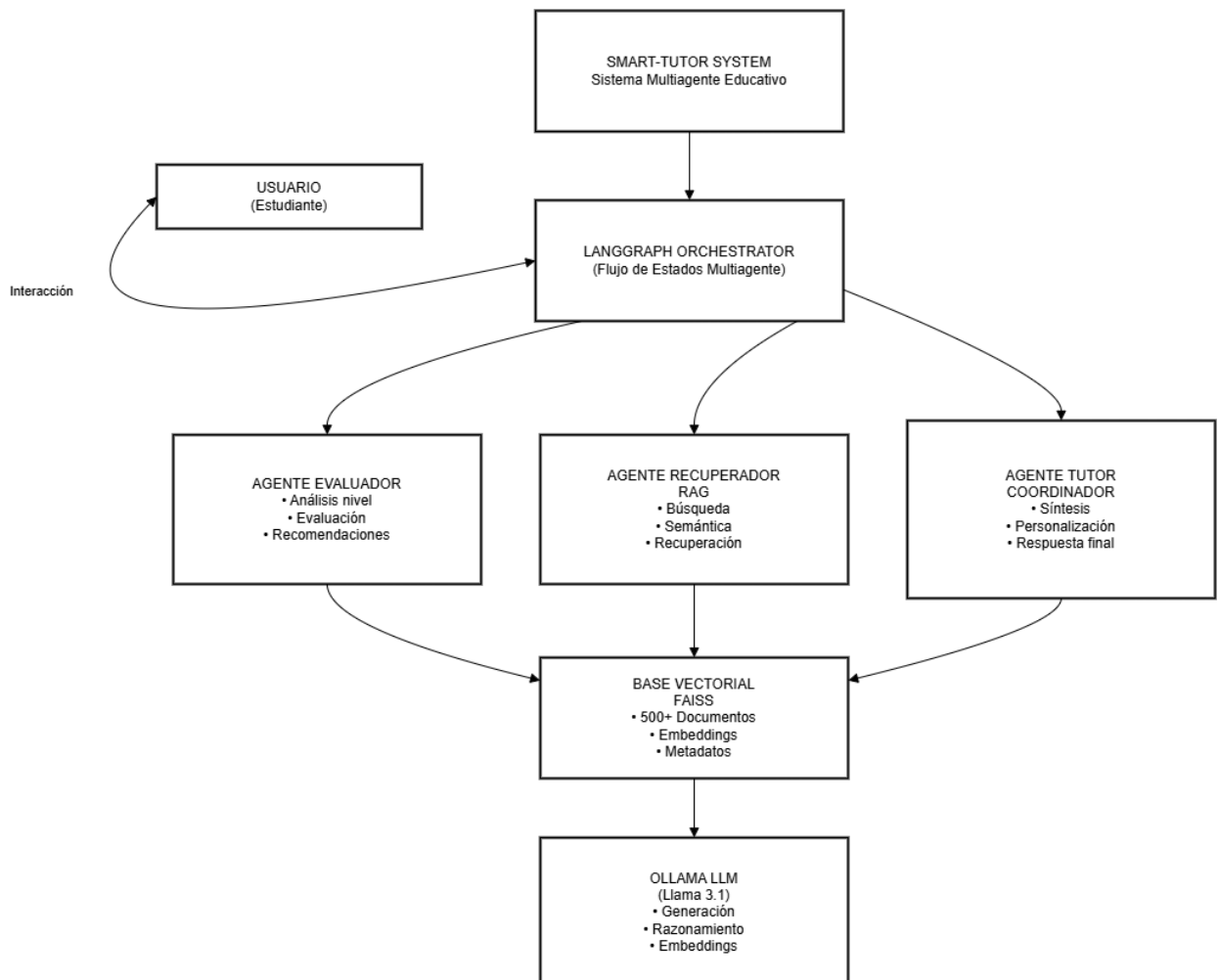
LangChain + LangGraph:

- Abstracción robusta para manejo de LLMs
- Flujos de trabajo complejos multiagente
- Integración nativa con bases vectoriales

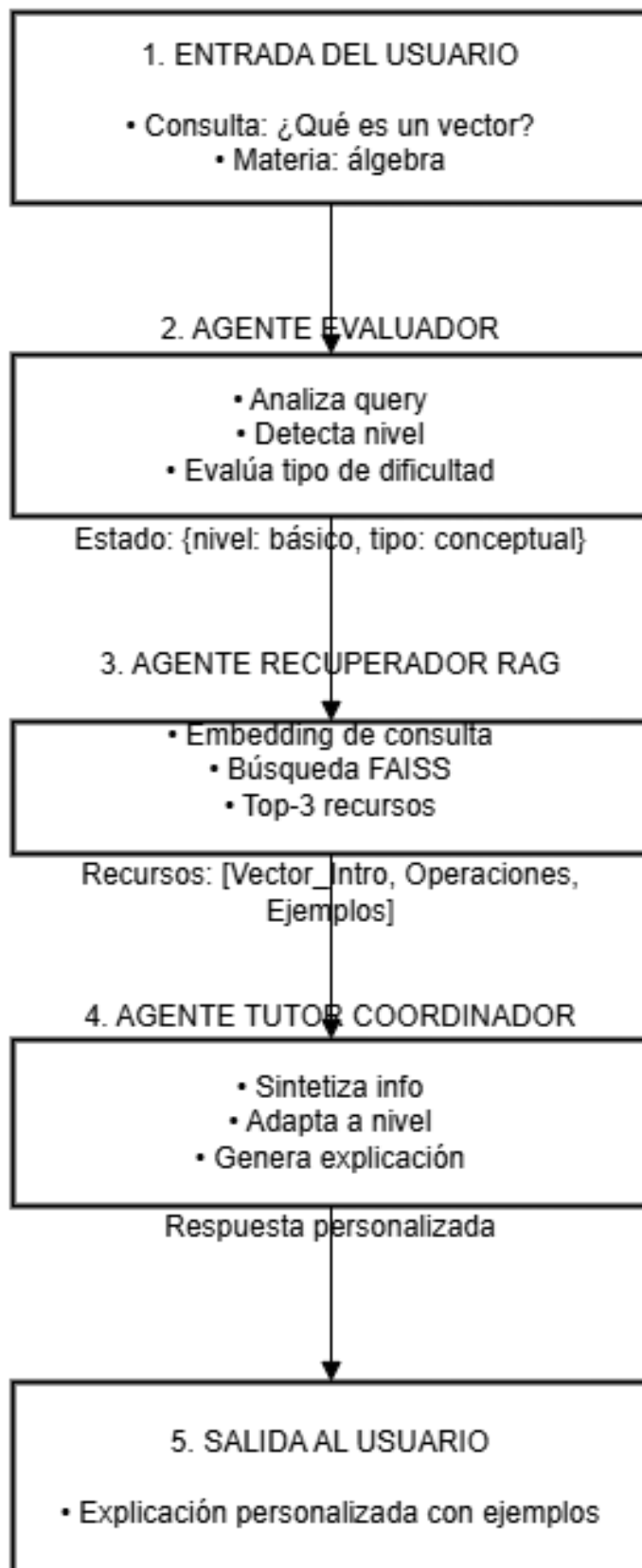
FAISS:

- Búsqueda vectorial ultrarrápida
- Escalabilidad para grandes datasets
- Integración perfecta con embeddings

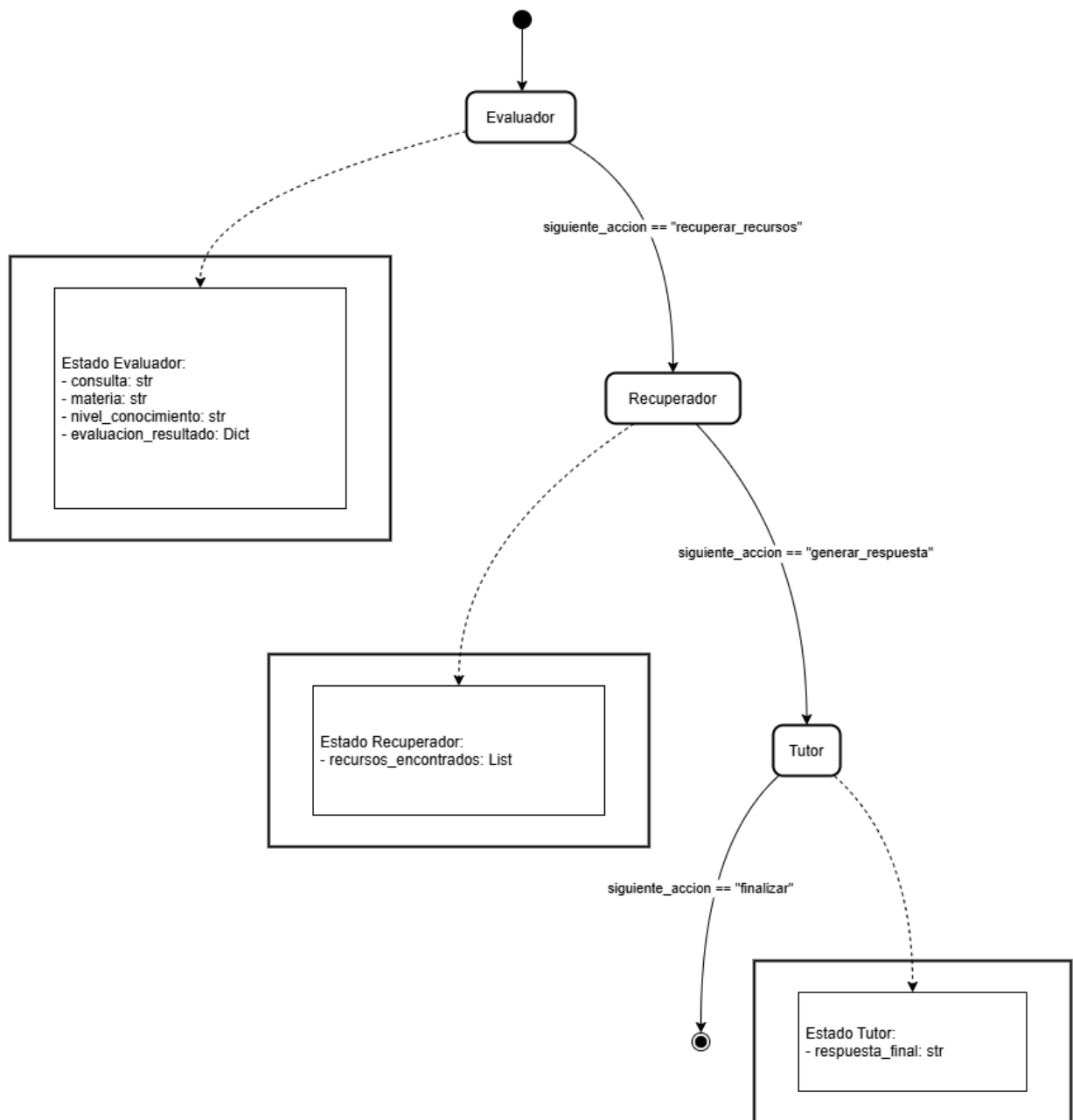
Diagrama de arquitectura



Flujo de Interacción Detallado



Grafo de Estados LangGraph



COMPONENTES EXTERNOS E INTEGRACIONES

Componentes Principales

- **Ollama Server:** Servidor local para modelos LLM
- **Base de Datos Educativa:** Contenido estructurado por materia y nivel
- **Sistema de Embeddings:** Vectorización en tiempo real
- **Memory Saver:** Persistencia de estados de conversación

Interfaces de Datos


```
# Formato de documentos en base vectorial
{
  "page_content": "Contenido educativo...",
  "metadata": {
    "materia": "algebra_lineal",
    "titulo": "Vectores",
    "nivel": "basico",
    "tipo": "concepto"
  }
}
```

CARACTERÍSTICAS TÉCNICAS DESTACADAS

Escalabilidad

- Arquitectura modular con agentes independientes
- Base vectorial extensible sin límites predefinidos
- Procesamiento asíncrono para múltiples estudiantes

Personalización

- Evaluación automática de nivel de conocimiento
- Adaptación dinámica de explicaciones
- Historial de aprendizaje personalizado

Robustez

- Manejo de errores en cada agente
- Fallbacks para garantizar respuestas coherentes
- Validación de estados en cada transición

Privacidad y Seguridad

- Procesamiento local con Ollama
- Sin envío de datos a servidores externos
- Control total sobre información estudiantil