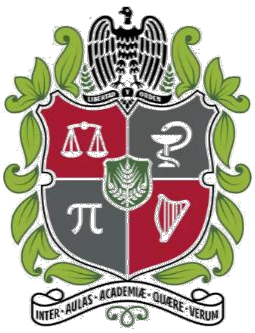

Universidad Nacional de Colombia

Sede Medellín

Procesamiento del Lenguaje Natural – 3011176

Trabajo Práctico 3 - Valor: 25%

Agentic AI: LLMs y Langchain 1.0

Profesor: Jaime Alberto Guzmán Luna

Fecha de Entrega: miércoles 10 de diciembre, Hora de cierre: 12:00 meridiano.
Fecha de Sustentación: miércoles 10 de diciembre de 4 a 8 pm.

INTRODUCCIÓN

Esta práctica tiene como fin la aplicación de los dos temas básicos vistos en clase: Transformers/LLMs y Agentic AI con LangChain 1.0. El estudiante debe seleccionar un dominio de aplicación (salud, agricultura, turismo, arte, educación, historia, etc) y construir un sistema Agentic AI basado en agentes orientado al análisis inteligente de documentos.

OBJETIVO GENERAL

Desarrollar un sistema Agentic AI multi-agente capaz de procesar, indexar, recuperar y analizar documentos mediante un modelo RAG, empleando al menos cinco agentes especializados implementados con LangChain 1.0 e integrando los modelos LLM Gemini y Groq de forma diferenciada.

OBJETIVOS ESPECÍFICOS

1. Diseñar e implementar un flujo Agentic AI multi-agente que integre de manera orquestada al menos cinco agentes funcionales basados en LangChain 1.0

2. Implementar un agente especializado para consumo, limpieza, chunking e indexación de documentos usando embeddings y un vector store (Faiss).
3. Construir un agente clasificador de intención del usuario capaz de reconocer cuatro tipos de consultas: búsqueda de información en el vector store (Faiss), resumen de documentos, comparación de documentos y consulta general (diferente a la almacenada en el sistema). Para ello se usará el LLM apropiado para identificar la intención de la solicitud del usuario.
4. Implementar un agente recuperador semántico basado en embeddings para localizar los documentos más relevantes frente a una consulta.
5. Desarrollar un agente generador de respuestas mediante RAG utilizando en LLM seleccionado para respuestas contextuales rápidas.
6. Implementar un agente crítico/verificador utilizando un LLM para validar coherencia, evitar alucinaciones y garantizar el uso apropiado del contexto.
7. Implementar al menos 5 herramientas (Tools) apropiadas para ser usadas en las actividades de los agentes
8. Integrar mecanismos de trazabilidad para registrar decisiones, rutas ejecutadas y documentos utilizados.
9. Elaborar un informe técnico del sistema completo.

DESCRIPCIÓN DETALLADA DEL SISTEMA A DESARROLLAR

1. **Agente de consumo / Indexador**
   - Cargar documentos (PDF/TXT/HTML). Al menos 100 documentos en el dominio seleccionado
   - Limpiar texto, segmentar en chunks y generar embeddings.
   - Indexar información en un vector store (FAISS).
2. **Agente Orquestador (Orquestador)**

   • Administrar el flujo completo del sistema.

   • Determinar qué agente debe ejecutarse según la consulta del usuario.

   • Debe usar un LLM para decisiones rápidas y eficientes.

3. **Agente Clasificador de Consultas (Clasificador)**
   - Clasificar la consulta del usuario en cuatro categorías:
     - Búsqueda de consumo: Solicita hechos o datos específicos contenidos en los documentos mediante lenguaje natural.
     - Resumen: Requiere hacer un resumen de uno o varios documentos.
     - Comparación: Solicita contrastar secciones/documentos.

- General: No requiere acceso al corpus ni a la recuperación de la base vectorial. Se realizará una pregunta directa a un LLM seleccionado en diseño.

- Alcance del agente:
  - Determinar si la consulta requiere búsqueda semántica.
  - Identificar si el usuario desea un resumen o comparación de documentos.
  - Detectar si la consulta puede resolverse sin RAG llamando al LLM seleccionado (intención de respuesta general).

• Este agente deberá usar un LLM, el cual permita una capacidad de interpretación profunda del lenguaje y comprensión contextual.

4. **Agente de Búsqueda Semántica (Recuperador)**
   - Ejecutar la búsqueda de similaridad semántica usando el vector store.
   - Seleccionar los documentos más relevantes.
   - Este agente deberá utilizar un LLM para optimizar la velocidad de recuperación.
5. **Agente de Respuesta con RAG (Agente de Respuesta)**
   - Construye una respuesta combinando:
     - La consulta del usuario.
     - Los fragmentos recuperados.
   - Produce respuestas justificadas con citas.
   - Este agente deberá utilizar un LLM para generar respuestas rápidas basadas en contexto.
6. **Agente Verificador / Crítico (Evaluador)**
   - Evalúa si la respuesta generada:
     - Está respaldada por el contexto recuperado.
     - Es coherente y libre de alucinaciones.
     - Cumple con los requerimientos del usuario.
   - En caso de fallo, solicita nueva respuesta al agente RAG (loop controlado).
   - Este agente deberá utilizar un LLM, para las tareas de razonamiento y validación compleja que se requiere.

USO DIFERENCIADO DE LLMs

Se hará uso de los LLMs Gemini y Groq en los agentes donde se solita el uso de un LLM. Se deberá hacer un análisis y presentar la respectiva justificación de cual LLM se usará y el porque es mejor para esa actividad.

FLUJO GENERAL DEL SISTEMA

1. Usuario → Orquestador.
2. Orquestador → Clasificador (Gemini).
3. Si intención ∈ {búsqueda, resumen, comparación}:
   - Recuperador → Agente de Respuesta → Evaluador.
   - Si la respuesta no es adecuada → regeneración de respuesta.
4. Si intención = general:
   - Responder directamente con el LLM del respectivo agente (clasificador).
5. Se retorna la respuesta final con trazabilidad completa.

REQUISITOS ESPECÍFICOS

1. Mantener trazabilidad explícita del flujo.
2. Procesar y analizar documentos reales.

ENTREGABLES

1. Código fuente de la implementación Agentic AI. Deberá incluir comentarios por función y explicación del flujo general.
2. Carpeta con documentos (PDF/TXT/HTML) utilizados: 100.
3. Informe técnico del sistema Agentic AI que incluya su diseño y registros de ejecución. Se incluirán la documentación de al menos 10 casos de uso que muestren las funcionalidades solicitadas del sistema. Igualmente deberá incluir la explicación del porqué se seleccionó Gemini y Groq en cada uno de los agentes donde se indicaba el uso de un LLM. Deberá ser un documento PDF.

SUSTENTACIÓN

1. Se realizará de manera presencial durante el horario de la clase. Durante esta se debe mostrar el funcionamiento completo del sistema y explicar:
   - La explicación clara del problema
   - Cómo se estructuraron los componentes del sistema Agentic AI en LangChain (agentes y sus requerimientos)
   - La demostración funcional del sistema con sus casos de uso.
- Se debe hacer una pequeña presentación en PowerPoint para soportar la presentación
- Se realizará una sesión de preguntas (10 min de sustentación y 5 min de preguntas).

- Todos los integrantes deben estar presentes en la sustentación y se podrá seleccionar uno cualquiera para responder preguntas.

METODOLOGÍA DE EVALUACIÓN

1. Código y documentación: 70%
2. Sustentación: 20%
3. Atención a preguntas: 10 %

MATERIAL POR ENTREGAR

Se deberá entregar en Google Classroom un archivo ZIP (con el siguiente nombre del archivo: practica3-grupo-XX–equipo-YY.zip, donde XX representa el número del grupo (1 o 2) y YY el número del equipo) que contenga lo siguiente:

- **Informe técnico:** Descripción detallada del sistema Agentic AI, junto con la justificación de las decisiones tomadas en cada parte del programa.
- **Código Fuente:** Todo el programa en Python, debidamente organizado y documentado.
- **Carpeta con documentos:** Documentos (PDF/TXT/HTML) utilizados: 100.

ANEXO 1

INSTRUCCIONES GENERALES PARA LA PRESENTACIÓN DE LOS TRABAJOS

- Los trabajos serán presentados en grupos de 4 personas definidos al inicio del semestre (ver grupos en el material de la semana 2).
- **No se admiten trabajos similares (incluyendo el planteamiento del dominio)**, en caso de presentarse programas con códigos similares serán anulados. La totalidad del trabajo solicitado al alumno, en esta evaluación práctica, deberá ser original y propio del autor que lo presenta. El estudiante es responsable de evitar que su material evaluable (código, solución al problema, memorias, etc.) sea accesible a estudiantes de otros grupos. En caso de que se detecten copias por incumplimiento de estas reglas, por acción o inacción, la sanción afectará a todos los estudiantes involucrados: quienes copien y quienes hayan sido copiados.
- Los integrantes de cada grupo deberán ser capaces de explicar y responder preguntas sobre el trabajo realizado durante la sustentación del trabajo.
- La evaluación de la práctica está asociada con la sustentación respectiva programada. Para tal fin, la persona que no asista a la sustentación, se considerará que no presentó la práctica en su totalidad (**"no se tendrá en cuenta el código, ni la documentación para la evaluación"**).
- El plazo de entrega del material asociado a la práctica es estricto y se indica al inicio de este documento. No se admiten trabajos después de esta fecha. No se considera como entrega aquella que solo contiene código o solo contiene el vídeo o solo contiene el documento de diseño o cualquier forma incompleta del trabajo.
- El código entregado en el archivo .zip en Google Classroom, debe ser el mismo que se ilustra en la sustentación. En caso de no cumplirse, se considera que el código no fue entregado y, por tanto, no se realizó la práctica por parte del grupo.

NOTA1: Si en la fecha de la sustentación no se presenta el material a entregar total, el trabajo de la práctica se considerará no entregado.

NOTA 2: La sustentación del trabajo se realizará en la fecha indicada por parte de todos los estudiantes del grupo y las preguntas orales deben ser atendidas por los estudiantes a criterios del profesor.

NOTA 3: La redacción de este documento de práctica puede presentar algunos errores o la ausencia de algunos requisitos específicos, los cuales deberán ser informados por parte de los estudiantes al profesor, para ser aclarados de manera colectiva durante la clase. Estas aclaraciones harán parte de los requisitos de la práctica.