

Trabajo Corto (%)

Fecha de Entrega: 19 de Junio 2025

Sistema Multiagente con RAG

Objetivo

Diseñar e implementar un sistema multiagente inteligente que integre técnicas de inteligencia artificial generativa como Retrieval-Augmented Generation (RAG), LangChain, LangGraph y bases de datos vectoriales. El sistema debe mostrar la colaboración efectiva entre agentes especializados para resolver una tarea compleja.

Casos de Uso (escoja uno)

Seleccione uno de los siguientes escenarios para desarrollar su sistema:

- Asistente Legal Inteligente:** Ayuda a redactar respuestas legales a partir de legislación, jurisprudencia y contratos.
- Tutor Académico Personalizado:** Guía el estudio de un estudiante en un área específica (ej. álgebra lineal o historia), sugiriendo recursos y respondiendo dudas.
- Analista de Ciberseguridad:** Detecta amenazas en reportes de logs, consulta patrones conocidos y genera alertas o acciones recomendadas.
- Asistente para Proyectos de Investigación:** Apoya en la redacción de artículos, gestión de referencias y resumen de papers científicos.
- Agente de Soporte Técnico IT:** Responde problemas técnicos a usuarios, recupera soluciones desde manuales o foros, y coordina atención entre agentes.

Entregables

El trabajo incluye una propuesta técnica, una implementación funcional y una presentación en video.

1. Propuesta Técnica

- Nombre del sistema y descripción general

- **Agentes involucrados:** al menos 3, indicando:
 - Nombre, rol, prompts utilizados
 - Cómo interactúa con otros agentes
- **Uso de RAG:** al menos un agente debe utilizar recuperación semántica con base vectorial.
- **Tecnologías utilizadas:** LangChain, LangGraph, modelo base (ej: GPT-4 o Mistral), FAISS/Chroma/Weaviate, etc.
- **Diagrama de arquitectura del sistema:** flujo o grafo de interacción entre agentes y componentes externos.

2. Implementación Funcional

- Implementación en Python, organizada y modular.
- Integración de LangChain y LangGraph con lógica multiagente.
- Base vectorial operativa con un set de datos propio o ejemplo.
- El sistema debe ejecutarse desde un archivo principal con instrucciones claras.

3. Video de Presentación

- Video de 5–8 minutos donde se explique:
 - El caso de uso elegido
 - La arquitectura del sistema
 - Una ejecución real del sistema funcionando
- Puede subirse a YouTube, Drive o plataforma de su elección (con acceso público o por enlace).

Evaluación

Criterio	Puntaje Máx.
Propuesta técnica clara y completa	10
Definición e interacción coherente entre agentes	10
Uso correcto y bien explicado del RAG	10
Diagrama de interacción claro y útil	5
Implementación funcional y modular	10
Video claro y demostración real del sistema	5
Calidad del código, organización y documentación	5
Total	55 puntos

Consideraciones Finales

- El trabajo puede realizarse de forma individual o en parejas.
- La propuesta debe entregarse en PDF.
- El código debe estar en un repositorio (GitHub o similar) con instrucciones claras ('README.md').
- La solución debe ser funcional y reflejar los conceptos aprendidos en clase.