

# Proyecto Final Integrador

## Descripción General

Este proyecto consiste en el **desarrollo y la mejora de un Sistema de Agentes Inteligentes** partiendo del código base de "[Bruno, el Mozo Virtual](#)" proporcionado en el curso. El objetivo es evolucionar este agente conversacional simple hacia un sistema más complejo, capaz de integrar una base de conocimiento especializada y registrar su actividad en plataformas externas.

El proyecto requiere no solo el agente con **LangGraph** sino también la implementación de la observabilidad (*tracing*) y la persistencia de datos.

---

## Objetivos de Aprendizaje

Además de los objetivos fundamentales del curso, este proyecto busca:

- **Evolución Arquitectónica:** Transformar un agente base para demostrar una comprensión profunda de las transiciones y estados de LangGraph.
  - **Gestión de Conocimiento (RAG avanzado):** Implementar y optimizar una **Base de Conocimiento**, asegurando que el agente utilice el *retrieval* para fundamentar respuestas específicas.
  - **Integración de Sistemas Externos:** Conectar el agente con un servicio de persistencia externo (**Notion**) para grabar resultados estructurados, reportes o decisiones clave.
  - **Dominio de la Observabilidad:** Implementar **LangSmith** para el *tracing*, *debugging* y la evaluación del rendimiento del agente, justificando las decisiones de diseño a partir de los registros de trazas.
- 

## Requisitos Técnicos

El sistema desarrollado debe cumplir con los siguientes requisitos funcionales y técnicos:

1. **Punto de Partida:** El proyecto puede iniciarse a partir del **código base del "Mozo Virtual"** proporcionado en el curso.
2. **Operaciones con LangGraph:** El sistema debe demostrar el uso de un grafo de **LangGraph** con al menos **dos agentes conectados y en colaboración** (ej: un agente "Investigador" y un agente "generador de informe").
3. **Base de Conocimiento (RAG):**
  - Diseñar y construir una **Base de Conocimiento** propia que contenga la información necesaria para el tema elegido.
  - El agente debe demostrar la capacidad de **buscar y recuperar información de**

- esta base** para responder a consultas.
4. **Persistencia Externa (Notion):**
- Implementar una herramienta (Tool) que se conecte a la **API de Notion**.
  - El sistema debe ser capaz de **grabar un informe estructurado, un resumen de la tarea o una decisión clave** en una base de datos o página de Notion al finalizar una tarea principal.
5. **Observabilidad y Tracing (LangSmith):**
- Configurar el proyecto para utilizar **LangSmith** en el seguimiento de todas las trazas de ejecución (LLM calls, *tool calls*, estados del grafo).
- 

## Entregables

Cada grupo deberá presentar los siguientes elementos:

1. **Repositorio de Código Público (GIT):**
- El proyecto finalizado debe ser entregado en un **repositorio de Git (GitHub/GitLab) de acceso PÚBLICO**.
  - Debe incluir un archivo requirements.txt y un archivo README.md claro con instrucciones para la instalación, configuración (variables de entorno) y ejecución.
2. **Documentación :**
- Un documento que incluya:
    - **Tema seleccionado:** Explicar el tema de interés seleccionado.
    - **Arquitectura (LangGraph):** Diagrama de la arquitectura del grafo/MCP, explicando los nodos, las funciones de transición y la lógica de control.
    - **Prompts:** Ejemplos de *prompts* clave.
    - **Análisis de Observabilidad:** Enlace a la *Project Run* de LangSmith.
    - **Links Repo git:** Link del repositorio que usaron.
3. **URL de Persistencia:** Enlace público a la página o base de datos de **Notion** donde el agente graba sus informes de manera autónoma.
- 

## Criterios de Evaluación y Rúbrica

Para la evaluación del proyecto se seguirá la siguiente rúbrica:

Criterio	Descripción Detallada
<b>I. Solidez Técnica y Funcionalidad</b>	Robustez del código, evolución del proyecto base. Correcto diseño de estados y transiciones en LangGraph. Implementación de las integraciones obligatorias (RAG,

	Notion, LangSmith).
<b>II. Diseño Conceptual y Arquitectura</b>	Claridad y justificación de la arquitectura. Calidad de los <i>prompts</i> y las herramientas ( <i>tools</i> ).
<b>III. Integración y Observabilidad</b>	Uso efectivo de <b>LangSmith</b> para <i>debugging</i> y optimización. Persistencia exitosa y estructurada de datos en <b>Notion</b> .
<b>IV. Calidad de los Entregables</b>	Claridad, estructura del informe. Código bien comentado y organizado.