**1.- Análisis del Caso Organizacional (IE1)** Sodimac es una empresa líder en el rubro de la construcción y mejoramiento del hogar. La empresa enfrenta un problema recurrente en la atención a clientes, quienes consultan sobre disponibilidad de productos, compatibilidad de materiales, manuales de instalación y políticas de servicio.  
 La información necesaria para responder a estas consultas se encuentra dispersa en múltiples fuentes, lo que genera respuestas lentas, sobrecarga en los centros de atención y baja satisfacción del cliente.  
 El objetivo es diseñar e implementar un agente basado en LLM y RAG para centralizar la información y mejorar la rapidez, coherencia y trazabilidad de las respuestas.

**2.- Formulación de Prompts (IE2)** Se crearon prompts optimizados para controlar el comportamiento del LLM.  
 Ejemplo de prompt para clientes:

Eres un asistente experto en productos de construcción de Sodimac. Usa únicamente la información del contexto proporcionado. Si no encuentras la respuesta, responde “No tengo la información requerida en la base de datos de Sodimac”

Contexto recuperado: {retrieved\_chunks}

Pregunta del cliente: {user\_query}

Responde en un máximo de 5 líneas, citando la fuente del documento.

**Se aplicaron las siguientes estrategias:**

**Context grounding:** obligar al modelo a usar solo la información recuperada.  
**Control de formato:** respuesta breve con cita de la fuente.  
**Fallback:** evitar invenciones del modelo con un mensaje estándar.

**3.- Diseño e Implementación del Pipeline RAG (IE3, IE4)** El pipeline RAG se implementó con LangChain y FAISS.  
 El flujo del pipeline es el siguiente:

**Ingesta de documentos:** se cargaron fichas técnicas, catálogos, manuales y políticas de Sodimac.

**Preprocesamiento:** se limpió el texto y se dividió en fragmentos de 600 tokens con un solapamiento del 20%.

**Embeddings:** se generaron con el modelo all-minilm-16-v2

**Almacenamiento:** se indexaron los fragmentos en FAISS con sus metadatos.

**Consulta:** se convirtió la pregunta del usuario en un embedding y se buscaron los 5 fragmentos más relevantes.

**Prompt injection:** los fragmentos seleccionados se enviaron al LLM en el prompt estructurado.

**Generación de respuesta:** el LLM respondió con la información encontrada en los documentos.

**Validación:** se registró cada consulta, los fragmentos usados y el tiempo de respuesta.

**4.- Arquitectura de la Solución (IE5, IE6)** Los componentes principales de la solución son:

**Usuario:** realiza la consulta a través de una interfaz web o un chatbot.

**Frontend:** la interfaz de usuario.

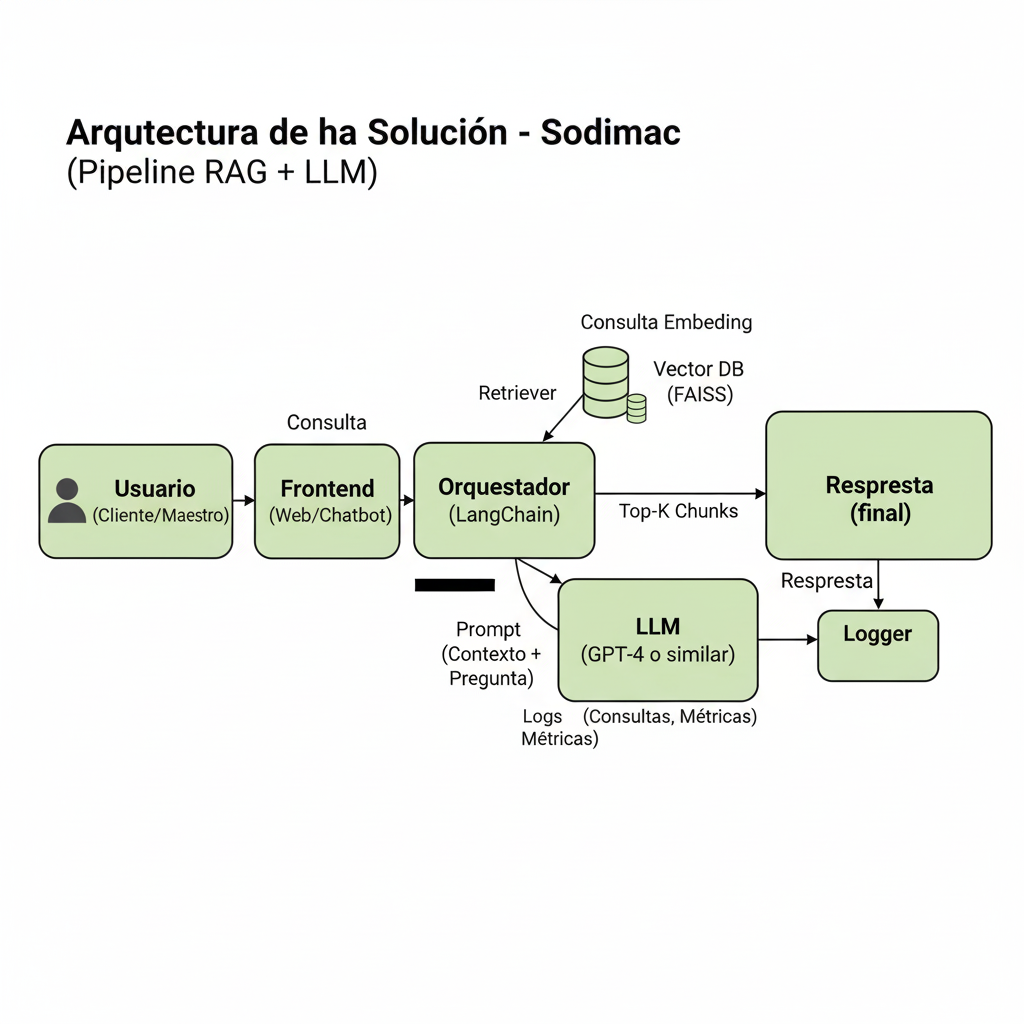
**Orquestador (LangChain):** gestiona el flujo entre el usuario, el recuperador de datos y el LLM.

**Vector DB (FAISS):** almacena los embeddings de los documentos.

**LLM (GPT-4 o similar):** genera la respuesta final.

**Logger:** almacena los registros de las consultas.

**Diagrama de arquitectura:**



**5.- Documentación Técnica (IE7, IE8)** Decisiones clave y su justificación:

Se eligió RAG sobre un LLM puro para evitar respuestas inventadas y asegurar la fuente.

FAISS se seleccionó por ser una opción gratuita, local y eficiente para prototipos.

El modelo de embeddings MiniLM se eligió por su buen balance entre precisión y costo.

El tamaño de los fragmentos de 600 tokens se definió para abarcar la información técnica.

La temperatura del LLM se ajustó a 0.1 para que las respuestas fueran basadas en hechos.

**6.- Redacción Técnica, Coherente y con Evidencias (IE9)** El informe utiliza un lenguaje técnico, apoyado en ejemplos de prompts, flujos de datos y diagramas de arquitectura.

**7.- Referencias (APA)** Sodimac. (2025). *Catálogo y fichas técnicas de productos.* Recuperado de<https://sodimac.falabella.com> LangChain. (2025). *Documentation.* Recuperado de<https://docs.langchain.com> Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *EMNLP.*

**7.- Reflexiones finales**Claudio Castillo: En el desarrollo de este proyecto, inicialmente necesitamos encontrar un caso de negocio con el cual nos sintiéramos cómodos para desarrollar, el cual fue el caso de Sodimac, debido a que este mismo, requiere de documentos básicos y de fácil acceso, de esta forma reuniendo en un solo punto, distinta variedad de información relacionada a herramientas, ya sea fichas técnicas, gracias a este proyecto, aprendí a diseñar prompts optimizados para definir el comportamiento y las acciones del modelo y también evitando la entrega de respuestas inventadas

Fabrizzio Mego: Durante el desarrollo que realice en este proyecto, fui capaz de aprender de forma general, el funcionamiento y la implementación del pipeline RAG, utilizando herramientas como LangChain y FAISS, desarrollando código para la ingesta de documentos, además fui capaz de entender la importancia de la gestión de versiones y colaboración mediante Github, realizando commits y elaborando la creación del archivo README que me permitió estructurar la información de ejecución del proyecto, lo que mas destaco fue aprender a como estructurar el