

Propuesta de Proyecto

AI Generativa para Arquitectura de Software

Integrantes:

Juan Pablo Castro - 202222086

Manuel Gómez - 202020415

Santiago Casasbuenas - 202214932

Introducción:

Este documento explica el proyecto que se quiere realizar como equipo, nuestras motivaciones y su funcionamiento. Adicionalmente, se propone un cronograma de trabajo. Es importante resaltar que en este documento se abarca la propuesta de uno de los dos equipos de trabajo que están planteados para el proyecto, por lo que nuestra propuesta de trabajo y cronograma pueden diferir considerablemente.

Motivación

El proyecto surge como continuación del desarrollo de ArchIA, un asistente conversacional especializado en arquitectura de software que integra modelos de lenguaje de gran escala con herramientas especializadas para apoyar la toma de decisiones en el diseño de sistemas. Actualmente el sistema cuenta con una arquitectura basada en un Grafo de Estados implementado con LangGraph, un módulo RAG soportado por ChromaDB y modelos de OpenAI, y nodos especializados para estilos, tácticas, ASR y generación de diagramas. Aunque el sistema es funcional, su estado actual requiere mejoras estructurales en tres frentes principales: el módulo RAG, los prompts de los agentes y el flujo del grafo, así como en la calidad del producto final generado por el agente. El propósito es consolidar un MVP robusto y, de forma paralela, ampliar sus capacidades incluyendo soporte más explícito para atributos de calidad como seguridad y disponibilidad.

Definición del Problema

ArchIA implementa un backend que orquesta agentes especializados mediante un supervisor central que enruta solicitudes hacia nodos como ASR, Style, Tactics y Diagram Agent. Sin embargo, el desempeño y la coherencia del sistema dependen críticamente de la calidad del módulo RAG, de la precisión de los prompts que guían el razonamiento de los agentes y de la correcta transición entre estados del grafo. El problema central radica en que el estado actual no garantiza un nivel óptimo de fundamentación técnica, consistencia en las respuestas ni calidad estructurada del output, lo que limita su consolidación como MVP orientado a arquitectos de software. Además, aunque el sistema contempla atributos como escalabilidad y latencia, se requiere fortalecer el soporte sistemático para seguridad y disponibilidad como capacidades explícitas dentro del flujo completo de decisión arquitectónica. El manejo de contextos también es un problema al no mantenerse a lo largo de una conversación de forma correcta.

Objetivos

El objetivo general es evolucionar el estado actual de ArchIA hacia un MVP funcional y validado que apoye efectivamente a arquitectos de software en la toma de decisiones, promoviendo altos estándares de calidad arquitectónica como seguridad, escalabilidad, mantenibilidad y desempeño. Específicamente, se busca mejorar el módulo de recuperación de información (RAG) basado en ChromaDB y embeddings de OpenAI, optimizar los prompts que estructuran el comportamiento de los nodos especializados y del Supervisor, refinar el flujo del Grafo de Estados para garantizar coherencia en el razonamiento multi-paso, y elevar la calidad y estructura del output final generado por el sistema. De manera paralela, se pretende incorporar nuevas funcionalidades que refuercen el soporte para atributos de calidad como seguridad y disponibilidad.

Justificación

ArchIA está diseñado para intervenir en fases tempranas del ciclo de vida del desarrollo de software, particularmente en análisis de requisitos, diseño arquitectónico, documentación y validación. Las decisiones arquitectónicas tomadas en estas etapas impactan directamente atributos de calidad como disponibilidad, seguridad y desempeño, por lo que es fundamental que la herramienta ofrezca recomendaciones fundamentadas, coherentes y técnicamente sólidas. El proyecto adopta una metodología ágil basada en iteraciones quincenales de diseño y validación, lo cual permite introducir mejoras incrementales y evaluar el comportamiento del sistema frente a escenarios

representativos. La mejora del RAG, del flujo del grafo y del diseño de prompts es necesaria para garantizar trazabilidad entre requisitos arquitectónicos significativos (ASR), estilos seleccionados, tácticas propuestas y diagramas generados, fortaleciendo la confiabilidad del sistema como herramienta de apoyo profesional.

Alcance

El proyecto incluye la mejora del backend que orquesta agentes mediante LangGraph, la optimización del módulo RAG soportado por ChromaDB y embeddings de OpenAI, el refinamiento de los nodos especializados (ASR, Style, Tactics y Diagram Agent) y la consolidación del flujo completo hasta la generación de la respuesta final. También comprende la ejecución de pruebas automatizadas E2E para verificar el correcto funcionamiento del sistema y pruebas de calidad con arquitectos de software de distintos niveles de experiencia, quienes evaluarán la herramienta mediante formularios estructurados. Por el tiempo que se lleva desarrollando el proyecto (varios semestres) se quiere enfocar en lograr un producto mínimo viable en este, y continuar el desarrollo en semestres posteriores. El proyecto se desarrolla con fines académicos y contempla la anonimización de los trabajos utilizados como fuente de información en el RAG.

Restricciones del Proyecto

El proyecto debe cumplir con restricciones éticas y técnicas. Los trabajos académicos utilizados como fuente para el sistema RAG deben ser anonimizados para proteger datos personales y evitar problemas de propiedad intelectual. La información que los usuarios carguen en la plataforma debe almacenarse de forma encriptada, con controles estrictos de acceso, y no podrá utilizarse para entrenar el modelo ni para fines distintos a la consulta puntual del usuario. Además, debe dejarse explícito que ArchIA es una herramienta de apoyo y no reemplaza el criterio profesional del arquitecto de software.

Principales Requisitos

El sistema debe ser capaz de analizar la intención del usuario mediante un nodo Supervisor, enrutar la solicitud hacia nodos especializados y mantener un estado tipado del grafo durante toda la conversación. Debe permitir la extracción de Requisitos Arquitectónicamente Significativos (ASR) estructurados como escenarios de atributo de calidad, recomendar estilos arquitectónicos y sugerir tácticas alineadas con atributos

como escalabilidad, disponibilidad, latencia y seguridad. Asimismo, debe generar diagramas como lo son los de despliegue y componentes, soportar persistencia de estado, almacenar decisiones arquitectónicas y registrar retroalimentación del usuario. El sistema debe ser validado mediante pruebas automatizadas E2E y pruebas de calidad con usuarios finales.

Cronograma de trabajo

El proyecto se desarrolla bajo una metodología ágil basada en iteraciones (sprints) quincenales de diseño y validación. En cada ciclo se definen objetivos específicos, se implementan mejoras funcionales incluyendo optimización del RAG, refinamiento de prompts y ajustes en el flujo del grafo y se evalúa el comportamiento del sistema frente a escenarios de uso representativos. Las mejoras se validan mediante pruebas automatizadas E2E para garantizar estabilidad del producto y pruebas de calidad con arquitectos de software, quienes utilizan la herramienta durante un periodo determinado y registran su experiencia en un formulario estructurado. Este enfoque iterativo permite avanzar progresivamente hacia la consolidación del MVP y la incorporación de nuevas funcionalidades como soporte reforzado para seguridad y disponibilidad.

Se tiene entonces este cronograma planteado:

			Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8
Tarea	Categoría	Duración	19/02/2026	05/03/2026	19/03/2026	02/04/2026	16/04/2026	30/04/2026	14/05/2026	28/05/2026
Organizar Repositorio	Fix	2	x	x						
Implementación Reflexion Agent	Feat	2	x	x						
Extensión del RAG	Feat	4		x	x	x	x			
Implementación diagramas	Fix	3		x	x	x				
Mejora de prompts	Feat	3			x	x	x			
Mejora de manejo de contextos	Feat	4			x	x	x	x		
Mejora de contexto con RAG	Feat	2				x	x			
Unificación de respuestas de varios nodos	Feat	3					x	x	x	
Mejoras de latencia de respuesta	Fix	2					x	x		
Deep Research dentro del RAG	Feat	2						x	x	
Mejoras UI/UX	Feat	2						x	x	
Integración con Seguridad	Feat	2							x	x
Integración con Disponibilidad	Feat	2							x	x
Finalización MVP	Goal	2							x	x
Despliegue del proyecto	Feat	1								x