ReparaYA

te lo reparamos al toque

Nombres: Vicente Sanchez

Felipe Caceres

Sección : 003D

Profesor : GIOCRISRAI BONILLO

Fecha : 24-09-2025

**Índice**

**Introducción  
 1.1…………………………………………………………………… Objetivo del informe  
 1.2…………………………………………………………………… Alcance del proyecto**

**Análisis del Caso Organizacional  
 2.1…………………………………………. Descripción de la organización (ReparaYA)  
 2.2…………………………………………. Problema o desafío identificado  
 2.3………………………………………… Objetivos de la intervención  
 2.4………………………………………… Datos disponibles y restricciones  
 2.5…………………………………………. Motivación para el uso de IA (LLM + RAG)**

**Formulación de Prompts  
 3.1…………………………………………… Principios de diseño de los prompts  
 3.2. Ejemplos de prompts adaptados al caso**

**Diseño e Implementación del Pipeline RAG  
 4.1……………………………………………. Fuentes de datos internas y externas  
 4.2…………………………………………….. Flujo de información y recuperación  
 4.3…………………………………………….. Integración con GPT-4o**

**Arquitectura de la Solución  
 5.1……………………………………………… Componentes principales  
 5.2……………………………………………….Descripción del flujo del sistema  
 5.3………………………………………………. Diagrama de arquitectura**

**Documentación Técnica y Justificación  
 6.1………………………………………………. Decisiones de diseño tomadas  
 6.2………………………………………………. Esquemas y evidencias técnicas  
 6.3………………..……………………………… Limitaciones y posibles mejoras**

**Conclusiones y Reflexiones  
 7.1………………………………………………… Aprendizajes obtenidos  
 7.2………………………………………………….Contribución individual (reflexión personal)**

**Anexos (opcional)**

* **Código fuente en GitHub**
* **Evidencias de pruebas realizadas**

## 

## 

## **A. Análisis del Caso Organizacional (IE1)**

**ReparaYA** es un taller mecánico ubicado en Santiago, dedicado a mantenimiento preventivo y correctivo de vehículos. Atiende a clientes particulares y flotas pequeñas.

**Problemas identificados:**

1. Reservas desordenadas vía WhatsApp/llamadas → sobrecarga administrativa.
2. Ausencia de historial digital de clientes/vehículos.
3. Atención al cliente limitada, sin respuestas rápidas a consultas frecuentes.

**Objetivo de la intervención con IA:**

* Implementar un **agente virtual inteligente** que gestione citas, responda preguntas y genere presupuestos preliminares.
* Integrar un **pipeline RAG** para acceder a manuales técnicos e historial del cliente.
* Mejorar la satisfacción y fidelización del cliente mediante atención personalizada y disponible 24/7.

## **B. Formulación de Prompts Adaptados al Caso (IE2)**

El diseño de prompts es clave para guiar al modelo. Se consideran principios como **contextualización, claridad y rol definido**.

**Ejemplos de prompts optimizados:**

* **Atención al cliente:** “Eres un asistente del taller mecánico ReparaYA. Responde de forma clara y profesional a esta consulta: *¿Cuánto cuesta un cambio de pastillas de freno para un Toyota Yaris 2018?*”
* **Gestión de citas:** “Agenda una cita para el cliente [nombre] con el servicio [servicio solicitado] el día [fecha], confirmando disponibilidad en el calendario.”
* **Recuperación técnica (RAG):** “Según el manual técnico del vehículo {contexto recuperado}, indica el procedimiento básico para realizar un cambio de aceite en un Nissan Versa 2020.”

## **C. Diseño e Implementación de Pipeline RAG (IE3, IE4)**

El pipeline propuesto combina **fuentes internas** y **fuentes externas**:

* **Fuentes internas:**
  + Historial digital de servicios (PDFs, facturas).
  + Base de datos de clientes y vehículos.
* **Fuentes externas:**
  + Documentación de fabricantes (manuales en línea).
  + Recursos públicos sobre mecánica automotriz.

**Flujo RAG:**

1. Entrada de pregunta del cliente.
2. Recuperación de documentos relevantes (vectorstore FAISS).
3. Envío de contexto + prompt a GPT-4o.
4. Generación de respuesta precisa y justificada.

Esto asegura **coherencia entre datos y respuestas**, fortaleciendo la credibilidad del agente.

## **D. Arquitectura de la Solución con Diagrama (IE5, IE6)**

**Componentes principales:**

* **Frontend:** Web/chatbot para clientes.
* **Backend:** API con LangChain.
* **Motor LLM:** GPT-4o (OpenAI) + opción de GitHub Models.
* **Vectorstore:** FAISS con embeddings de OpenAI.
* **Fuentes de datos:** Base interna + manuales técnicos externos.

**[Cliente Web/App]**

**│**

**▼**

**[Interfaz Chatbot]**

**│**

**▼**

**[Pipeline RAG - LangChain]**

**│**

**┌───────────────┐**

**│ Vectorstore │ ← Datos internos (PDFs, BD clientes)**

**└───────────────┘**

**│**

**▼**

**[LLM GPT-4o / GitHub Models]**

**│**

**▼**

**[Respuesta al Cliente]**

## **E. Documentación Técnica con Justificación y Diagramas (IE7, IE8)**

* **Decisión de usar GPT-4o: ofrece rapidez y precisión en español.**
* **LangChain: facilita la construcción de chains (ingesta, embeddings, consultas).**
* **FAISS: librería eficiente para búsquedas vectoriales.**
* **GitHub Models: asegura flexibilidad y respaldo con otros LLMs alternativos.**

**Limitaciones:**

* **Dependencia de conexión a internet.**
* **Costos de uso de APIs a gran escala.**
* **Necesidad de validación humana para presupuestos complejos.**

## **F. Lenguaje Técnico con Evidencias (IE9)**

**Durante el desarrollo se realizaron pruebas de consultas:**

* **Pregunta: *“¿Cuánto cuesta cambiar frenos en un Nissan Sentra 2017?”***
* **Respuesta del agente: recuperó tarifas internas + repuestos disponibles y entregó un presupuesto preliminar.**

**Esto evidencia la coherencia entre datos almacenados y respuestas generadas, cumpliendo con los objetivos organizacionales.**

## **G. Normativa APA Referencias (ejemplo en APA):**

* **OpenAI. (2025). *GPT-4o: Technical Overview*.** [**https://openai.com/**](https://openai.com/)
* **LangChain. (2025). *Documentation*. https://python.langchain.com/**
* **FAISS. (2025). *Facebook AI Similarity Search*. https://faiss.ai/**
* **Duoc UC. (2025). *Guía de uso ético de la IA*.** [**https://bibliotecas.duoc.cl/ia**](https://bibliotecas.duoc.cl/ia)