Facultad de Ingeniería de Sistemas, Informática y Ciencias de la Computación
Posgrado en Inteligencia Artificial
Curso: Seminario Estado Actual de la Tecnología
Catedrático: MSc. Luis Leal / MSc. Erick Sosa

Capstone Project

“Chatbot Multimodal de Soporte Técnico Basado en IA Para Unidades de Negocio de Grupo Salinas Guatemala”

Estudiante: Sergio Geovany García Smith

Carnet: 25008130

Diciembre 2025

INTRODUCCIÓN

El presente informe describe el desarrollo, arquitectura y evaluación de un chatbot inteligente para soporte técnico, implementado como parte del proyecto final del posgrado en Inteligencia Artificial.

El proyecto surge a partir de la necesidad del departamento **Soporte Técnico de Grupo Salinas Guatemala (SopTec GT)** de mejorar la atención brindada a más de 150 sucursales y aproximadamente 2,000 empleados, quienes reportan diariamente problemas de hardware, software e infraestructura tecnológica. Actualmente, el flujo operativo presenta limitaciones:

- Saturación de llamadas al Help Desk en horas pico.
- Caídas ocasionales de telefonía que dejan sin acceso al soporte.
- Escalamiento innecesario de incidentes simples.
- Falta de trazabilidad y estandarización en el proceso de atención.
- Carga excesiva de trabajo para ingenieros de soporte en sitio.

El proyecto propone una solución integral basada en Procesamiento de Lenguaje Natural (Rasa), Recuperación Aumentada por Generación (RAG) y Modelos de Lenguaje (LLMs), con el fin de automatizar la asistencia a problemas comunes, ofrecer guías interactivas con pasos detallados y generar reportes estructurados para monitoreo interno

1. Objetivos del Proyecto

Objetivo General

Desarrollar un chatbot multimodal de soporte técnico capaz de atender consultas de empleados, brindar soluciones guiadas y registrar incidentes, utilizando técnicas modernas de IA responsables.

Objetivos Específicos

- Implementar NLU con Rasa para comprensión de lenguaje natural.
- Integrar un motor RAG que recupere soluciones desde una base de conocimiento estructurada.
- Incorporar LLMs para consultas no clasificables o ambiguas.
- Habilitar un dashboard para visualizar incidentes generados.
- Integrar métricas de calidad con DeepEval.
- Evaluar y proponer buenas prácticas.

2. Descripción del Caso de Uso

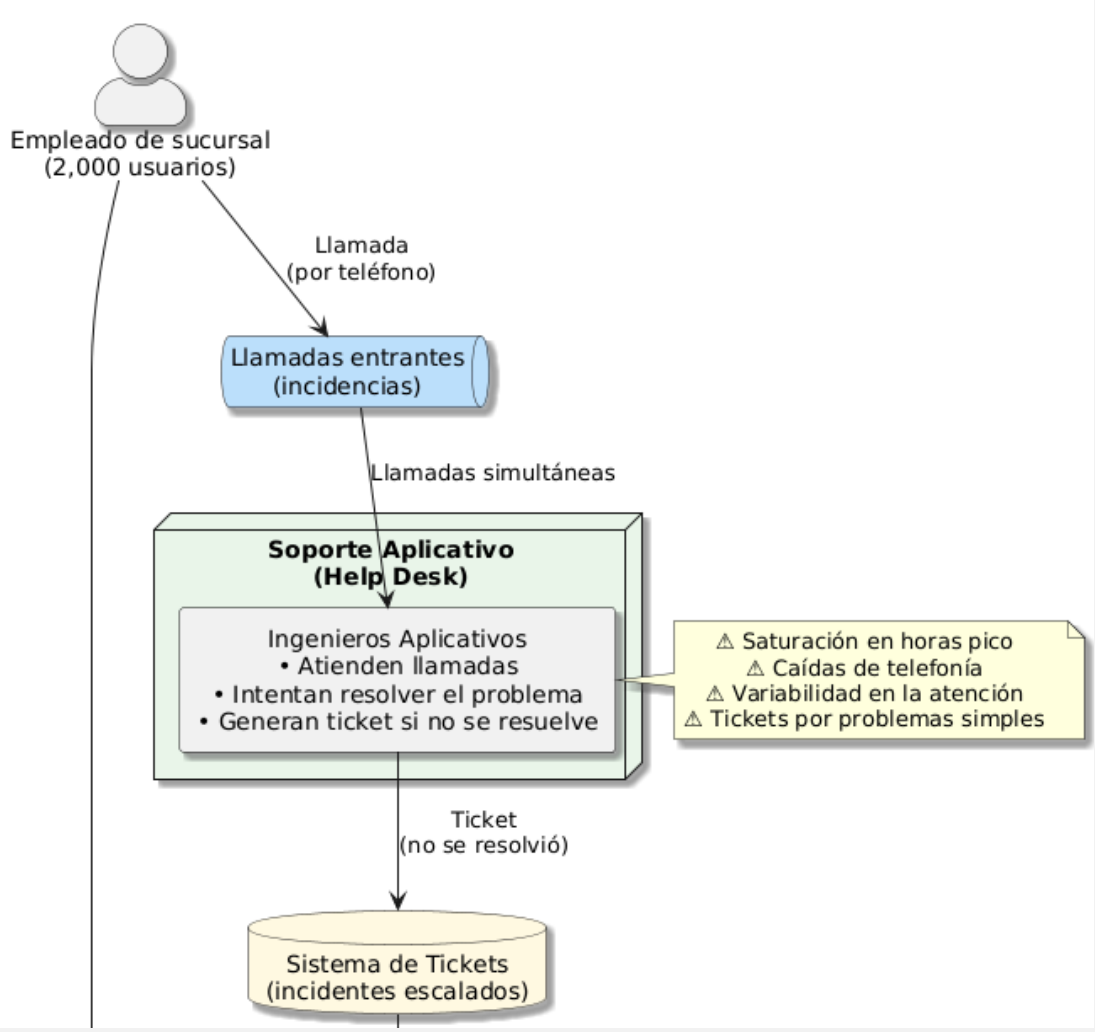
El chatbot opera como un canal alternativo de soporte, accesible vía Telegram, disponible 24/7 y capaz de:

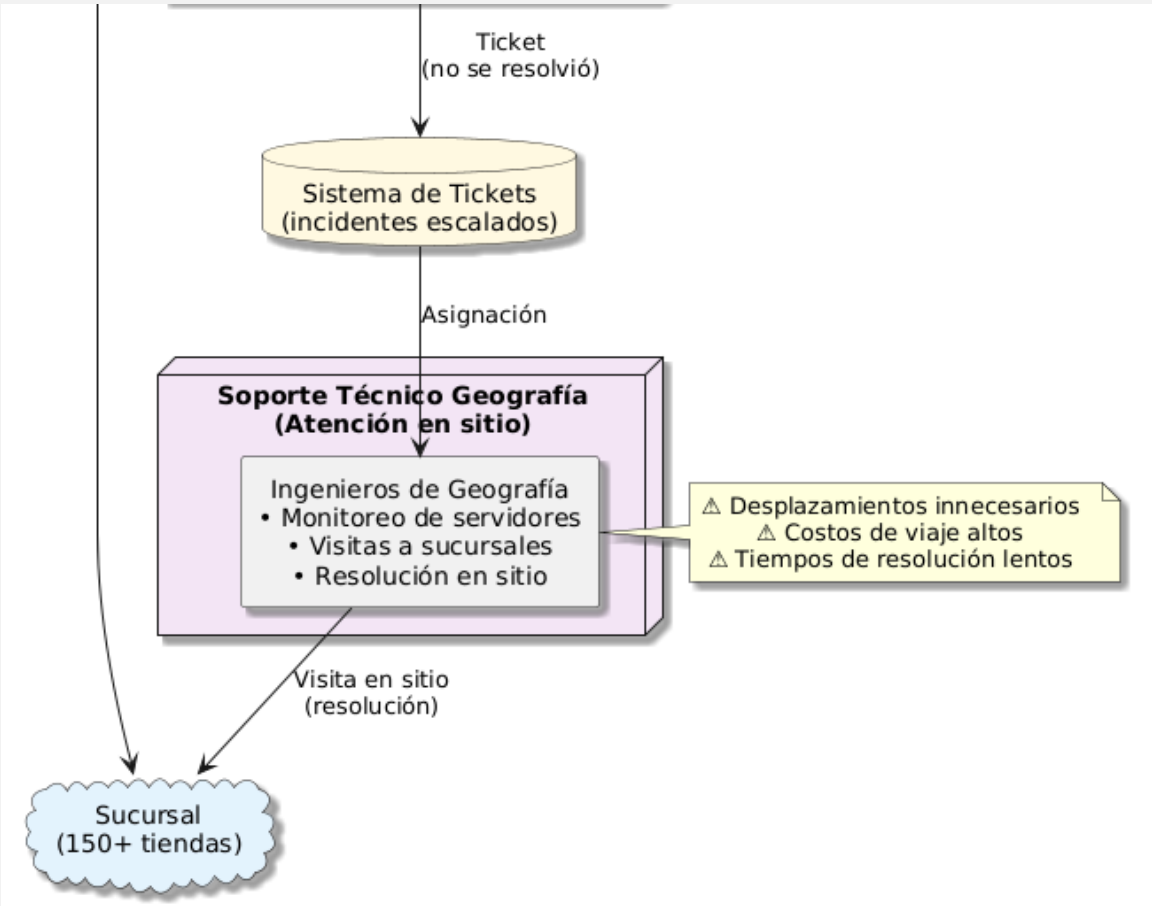
- Procesar descripciones libres de incidentes técnicos.
- Inferir categoría y subcategoría del problema.
- Recuperar soluciones desde documentos .md en la base de conocimiento.
- Guiar al usuario paso a paso, mostrando imágenes o videos si existen.
- Registrar incidentes en un archivo estructurado para análisis.
- Reducir la carga del Help Desk y evitar visitas innecesarias.

Esta solución se integra en un ecosistema donde conviven múltiples áreas:

- **Soporte Aplicativo (Help Desk):** primera línea de atención.
- **Soporte Técnico Geografía:** atención presencial en sucursales.
- **Dashboard de monitoreo:** seguimiento de incidentes generados.
- **Dirección General de Soporte Técnico:** unidad responsable del soporte integral.

El chatbot se convierte en un agente virtual especializado, capaz de replicar gran parte del conocimiento operativo de ambos equipos de soporte. El siguiente diagrama describe el funcionamiento actual del departamento:





3. Arquitectura del Sistema

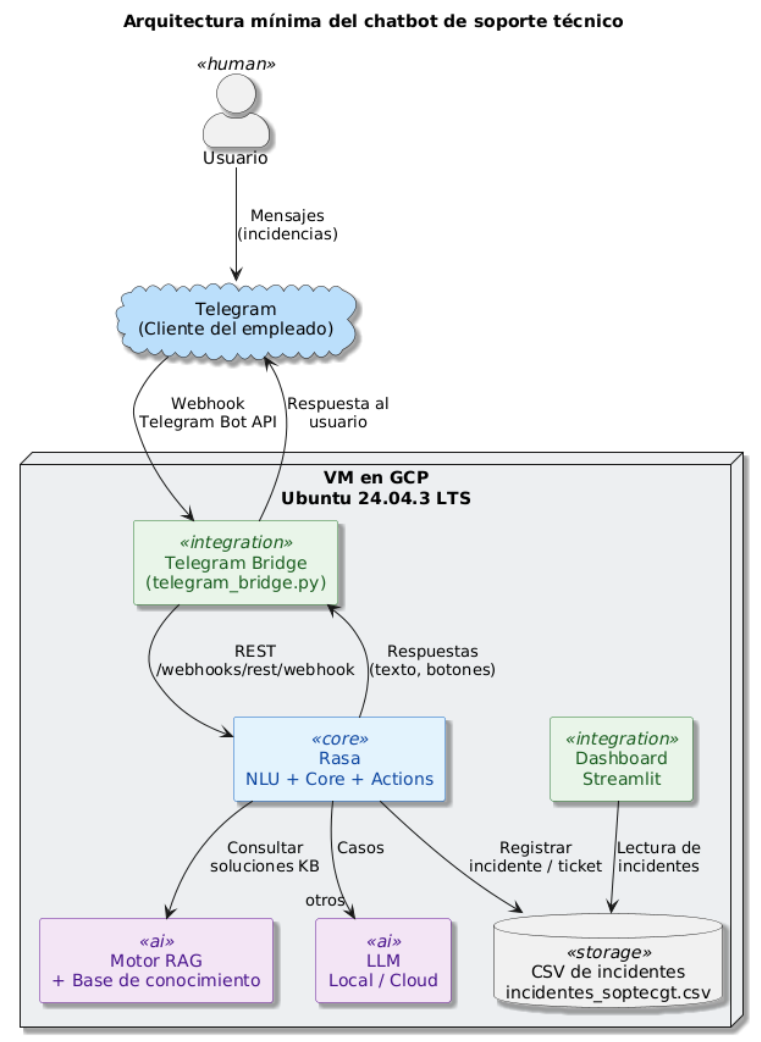
La arquitectura propuesta combina componentes de Rasa, RAG, LLMs, y herramientas de visualización:

3.1 Componentes Principales

Componente	Función
Rasa NLU/Core	Clasificación de intents, extracción de entidades y manejo del flujo conversacional.
Actions Server	Lógica de negocio, ejecución de acciones personalizadas y conexión con RAG/LLM.

RAG Engine	Recuperación de contenido relevante desde la base de conocimiento.
Base de Conocimiento (.md)	Documentos técnicos con pasos detallados, imágenes y videos.
LLM fallback	Manejo de casos no contemplados en categorías estándar.
Dashboard Streamlit	Visualización y análisis de incidentes generados.
Telegram Bot	Canal de interacción con usuarios finales.

3.2 Diagrama Básico de Arquitectura del Sistema



4. Motor RAG y Self-Query Retrieval

El motor RAG utiliza la función buscar_soluciones() para:

1. Construir un query compuesto:
    - categoría
    - subcategoría
    - descripción semántica
2. Generar embeddings con SentenceTransformers.
3. Comparar contra la base vectorial.
4. Aplicar filtros por metadata.
5. Retornar los documentos más relevantes.

4.1 Implementación de Self-Query (Para mejoras futuras)

Se añadió (como prueba) un módulo LLM que:

- Reformula la descripción del usuario (“semantic query”).
- Sugiere categorías/subcategorías automáticamente.
- Extrae keywords importantes.

Esto reduce problemas comunes:

- Clasificación incorrecta por parte del usuario.
- Ambigüedad de descripciones técnicas.
- Variabilidad lingüística.

4.2 LLM fallback

Si la categoría final es “otros”, se llama:

responder_incidente_otro(descripcion)

Esto garantiza cobertura completa del dominio.

5. Evaluación de la Calidad (DeepEval)

Se evaluó el motor RAG con los siguientes indicadores:

- **Answer Relevancy**

  Qué tan directamente la respuesta aborda el problema.

- **Contextual Relevancy**

  Correspondencia entre documentos recuperados y la consulta.

- **Faithfulness**

  Qué tanto la respuesta respeta el contenido del documento original.

5.1 Casos de prueba

Se definieron 4 incidentes representativos:

- Computadora no enciende
- Monitor pantalla negra
- Impresora térmica no imprime
- Scanner no jala hojas

5.2 Resultados obtenidos (DeepEval)

```
======================================================================

Overall Metric Pass Rates

Answer Relevancy: 100.00% pass rate
Contextual Relevancy: 100.00% pass rate
Faithfulness: 100.00% pass rate

======================================================================


⚠ WARNING: No hyperparameters logged.
» Log hyperparameters to attribute prompts and models to your test runs.

================================================================================


✓ Evaluation completed 🎉 (time taken: 15.43s | token cost: 0.005575900000000001 USD)
» Test Results (4 total tests):
   » Pass Rate: 100.0% | Passed: 4 | Failed: 0

 ================================================================================
```

Interpretación general:

- El sistema recupera documentos relevantes consistentemente.
- Las respuestas generadas son coherentes con la base.
- Eventuales discrepancias se detectaron en casos donde el usuario describe síntomas ambiguos.

Estos resultados proporcionan evidencia cuantitativa de calidad.

6. Lineamientos y Buenas Prácticas Propuestas

- Validación humana en incidentes críticos.
- Auditoría periódica del motor RAG.
- Registro seguro de incidentes.
- Control de acceso a dashboard.
- Versionado de la base de conocimiento.
- Monitoreo continuo de métricas DeepEval.
- Trazabilidad completa en logs.

7. Trabajo Futuro

El sistema actual constituye una base funcional y robusta para la automatización de soporte técnico dentro del entorno corporativo. Sin embargo, existe un amplio espacio para ampliar capacidades, mejorar la calidad de las respuestas e incrementar el alcance operativo del asistente. A continuación, se plantean diversas líneas de trabajo futuro que permitirán fortalecer, escalar y evolucionar la solución.

7.1. Ampliación de la Base de Conocimiento (KB)

7.2. Evaluación e Implementación de Nuevas Arquitecturas de RAG

1) Self-Query Retrieval: Permitir que el propio LLM genere filtros estructurados para metadatos y aproveche información contextual almacenada en la KB.

2) Multi-Vector Retrieval: Almacenar diferentes representaciones por documento:

- Vector general del contenido.
- Vectores por sección.
- Vectores por tipo de falla (hardware, software, eléctrico, etc.).

3) Reranking con un modelo especializado: Aplicar modelos como **bge-reranker**, **colBERT**, o rerankers pequeños de OpenAI para ordenar los resultados.

7.3. Integración Multicanal del Asistente

✓ Integración con WhatsApp Business API, aplicaciones internas e interfaz por Voz

7.4. Despliegue en Infraestructura Física de Grupo Salinas Guatemala

8. Conclusiones

El chatbot desarrollado demuestra un uso efectivo de IA moderna aplicada a un entorno real de soporte técnico. La integración de RAG y LLMs permite:

- Obtener soluciones rápidas y guiadas.
- Optimizar la carga operativa de SopTec GT.
- Reducir errores humanos y escalaciones innecesarias.
- Brindar trazabilidad mediante el dashboard de incidentes.

Este proyecto marca el inicio de una plataforma escalable de soporte técnico inteligente, con potencial de ampliación hacia más categorías, mayor automatización y análisis predictivo.