Chatbot para Asistencia

de

Soporte Informático

# 

# 

# 

# 

# 

### 

**Organización:** Empresa Servicios Globales Ltda.  
**Rubro:** Prestación de servicios financieros.  
**Tamaño:** Mediana empresa con 150 empleados administrativos y técnicos.  
**Contexto:** La empresa dispone de un área de soporte informático interno que gestiona alrededor de 200 tickets mensuales relacionados con hardware, software y conectividad de red.

### 2. Identificación y descripción del problema/desafío

El área de soporte enfrenta los siguientes problemas:

* Los empleados realizan preguntas repetitivas sobre procedimientos básicos como:  
  + Instalación de software.
  + Reinicio y encendido de equipos.
  + Copias de seguridad.
  + Conexión a la red.
* La atención manual de estas consultas genera:  
  + Retrasos en la resolución de incidentes críticos.
  + Saturación en la mesa de ayuda.
  + Reducción de la productividad de los empleados.

### 3. Objetivos de la intervención

* Implementar un **asistente virtual de soporte** que responda dudas frecuentes a partir de la documentación oficial.
* Reducir en un **50% los tickets repetitivos** en un plazo de 3 meses.
* Permitir a los usuarios resolver problemas básicos sin intervención directa del equipo de soporte.
* Centralizar el acceso a procedimientos y políticas internas de informática.

### 4. Datos disponibles o que se pueden obtener

* Documento interno de **Procedimientos de Soporte Informático – Oficinas** (manual en texto plano).
* Base de **tickets históricos** para identificar preguntas frecuentes (FAQs).
* Políticas internas de TI y seguridad informática.

### 5. Restricciones o requerimientos particulares

* Cumplimiento de **normativas de seguridad corporativa** (no exponer datos sensibles).
* Uso únicamente de **documentación aprobada** como base de conocimiento.
* Integración en el **portal corporativo de autoservicio TI**.
* Presupuesto acotado → uso de **embeddings pequeños** (text-embedding-3-small).

### 6. Análisis del caso organizacional

**Requerimientos específicos:**

* Recuperar fragmentos relevantes de la documentación según la pregunta del usuario.
* Mantener coherencia en interacciones prolongadas mediante memoria de conversación.
* Integración con interfaz web interna (Streamlit).

**Objetivos a resolver con IA:**

* Reducir tiempo de atención para consultas frecuentes.
* Garantizar precisión y cumplimiento de políticas internas.
* Optimizar la eficiencia del equipo de soporte.

### 7. Formulación de prompts

**Prompt del sistema (System Prompt):**

* Define al modelo como **asistente de soporte informático**.
* Instruye al modelo a usar únicamente la información recuperada de la base vectorial.
* Incluye directrices de tono, claridad y pasos prácticos.

**Justificación de elección:**

* No se usa **Zero-Shot** porque puede generar respuestas inventadas fuera del material oficial.
* No se usa **Few-Shot** porque no hay necesidad de entrenar con ejemplos; RAG ya proporciona contexto suficiente.
* No se usa **Chain-of-Thought** porque los problemas no requieren razonamiento complejo paso a paso.
* Se eligió **vectorstore + RAG** para garantizar respuestas precisas basadas en la documentación interna.

### 8. Diseño e implementación del pipeline RAG

1. **División de documentos en chunks** con RecursiveCharacterTextSplitter.
2. **Creación de embeddings** usando OpenAIEmbeddings(text-embedding-3-small).
3. **Almacenamiento en base vectorial** con FAISS.
4. **Búsqueda por similitud** al recibir una pregunta del usuario.
5. **Generación de respuesta** con ChatOpenAI(gpt-4o-mini), utilizando los chunks recuperados y el system prompt.
6. **Memoria de conversación** con ConversationBufferMemory para mantener contexto.

### 9. Arquitectura de la solución

**Componentes clave:**

* **Frontend:** Streamlit para la interfaz de chat web.
* **Base de conocimiento:** Documento de soporte dividido en chunks, indexado en FAISS.

### **Embeddings:** OpenAIEmbeddings para vectorizar los textos.

### **Modelo LLM:** ChatOpenAI para generar respuestas.

### **Memoria:** ConversationBufferMemory para mantener historial de conversación.

**Flujo de información:** Usuario → Streamlit → Base vectorial (FAISS) → Modelo LLM → Memoria → Respuesta al usuario

### 10. Documentación técnica

**Decisiones clave:**

* **Vectorstore + RAG:** Garantiza precisión y coherencia de respuestas basadas en documentación oficial.
* **System prompt:** Controla comportamiento, tono y directrices del asistente.
* **No usar Zero/Few-Shot ni Chain-of-Thought:** Evita errores y costos innecesarios.

**Conclusión**

El diseño asegura un chatbot confiable, eficiente y seguro, capaz de atender consultas frecuentes de manera inmediata, cumpliendo con políticas internas y buenas prácticas de soporte informático.