*Clínica de VITAL CARE*

*Integrantes: Joaquín Rohland, Jorge Curiqueo  
Sección: INGENIERIA DE SOLUCIONES CON INTELIGENCIA ARTIFICIAL\_002D  
Fecha: 22-09-2025*

*(Nota importante, usted profesor debe crear el entorno virtual ,a ya que github no registro la subida de el ambiente, solo se encuentra el código y las api keys que serán cambiadas a posterioridad por temas de seguridad,*

*En caso de no funcionar o acabarse los tokens de la api presente ahí, envio otra por este medio: ghp\_uxKQm5q2cUAqpf1Tko5vctbK2KfZPb1VFVTf (gpt4o) )*

*Introducción:*

Sistema de consultas médicas y administrativas para la Clínica VITAL CARE (ficticia), con evaluación automática de respuestas y monitoreo de métricas.

*Propósito*

* Permitir a los usuarios (pacientes y personal administrativo) realizar consultas sobre protocolos, guías de síntomas, horarios, procedimientos y políticas de privacidad. El sistema:
* Genera embeddings de documentos internos para búsqueda semántica.
* Recupera documentos relevantes mediante un enfoque híbrido (semántico + palabras clave).
* Genera respuestas basadas en documentos y evalúa su calidad automáticamente.
* Permite monitorear métricas de rendimiento y calidad.
* Facilita la exportación de resultados para análisis o integración con LangSmith.

*Tecnologías utilizadas*

* ***Python 3.10+***
* ***Streamlit:*** Interfaz web interactiva.
* ***LangChain:*** Para generación de embeddings y procesamiento de documentos.
* ***OpenAI API:*** Modelos LLM (gpt-4o) para generación de respuestas y evaluación de métricas.
* ***Plotly Express:*** Visualización de métricas y estadísticas.
* ***Pandas / NumPy***: Manejo de datos y métricas.
* ***dotenv:*** Carga de variables de entorno desde .env.

*Requerimientos*

***Variables de entorno en archivo .env:***  
***GITHUB\_TOKEN***=”…”

***GITHUB\_BASE\_URL***=https://models.inference.ai.azure.com

***OPENAI\_API\_KEY***=”…”

***Librerías Python:***

***pip install streamlit openai langchain scikit-learn plotly pandas numpy python-dotenv***

Estructura del proyecto

* ***main.py:*** Archivo principal que ejecuta la aplicación Streamlit.
* Documentos internos se cargan en memoria y pueden ser agregados, editados o eliminados desde la pestaña “Documentos”.
* Variables de estado principales:
* ***st.session\_state.eval\_rag:*** Contiene documentos, embeddings y configuración de logging.
* ***st.session\_state.interaction\_logs:*** Registra consultas y métricas de rendimiento.
* ***st.session\_state.messages:*** Historial de chat para la pestaña de Chatbot.

Funcionalidades Principales

***Consulta con Métricas (Tab “❓ Consulta”)***

* Entrada de preguntas por parte del usuario.
* Ajuste del número de documentos a evaluar (top\_k).
* Habilitación de evaluación automática de la respuesta.
* Logging de interacciones.
* Botón “Generar Embeddings” para calcular representaciones vectoriales de los documentos.
* Botón “Consultar con Métricas”:
* Realiza búsqueda híbrida (similitud semántica + palabras clave).
* Genera respuesta basada en los documentos recuperados.
* Calcula métricas:
  + Tiempo de recuperación (retrieval\_time)
  + Tiempo de generación (generation\_time)
  + Tiempo total
  + Fidelidad al contexto (1-10)
  + Relevancia (1-10)
  + Precisión del contexto (proporción de documentos relevantes)
* Muestra documentos evaluados y puntuaciones.

***Gestión de Documentos (Tab “📚 Documentos”)***  
***Visualización de documentos actuales.***

***Opciones de:***

* Editar: Modificar contenido de documentos.
* Eliminar: Borrar documentos individuales.
* Agregar: Añadir nuevos documentos mediante textarea o archivos .txt / .md.
* Limpiar Todos: Elimina todos los documentos.

***Estadísticas:***

* Número total de documentos.
* Longitud promedio.
* Total de palabras.
* Estado de embeddings (generados / no generados).

***Dashboard de Métricas (Tab “📊 Métricas”)***

**Visualización interactiva de métricas por consulta.**

**Gráficos:**

* Línea de tiempo de tiempo total por consulta.
* Histograma de fidelidad.
* Scatter plot de tiempo de recuperación vs tiempo de generación.
* Scatter plot de precisión de contexto vs relevancia.

**métricas generales**

* Total de consultas
* Tiempo promedio
* Fidelidad promedio
* Relevancia promedio

***Evaluación Sistemática(Tab “🧪 Evaluación”)***

* Ejecuta un conjunto de consultas predefinidas.
* Recupera documentos, genera respuestas y calcula métricas automáticas.
* Muestra resultados por consulta y métricas promedio.

**Analytics y Exportación (Tab “📈 Analytics”)**

* ***Exportación de datos de interacciones a:***
  + JSON para LangSmith.
  + CSV resumido con métricas.
* ***Análisis de documentos:***
  + Distribución de longitud de documentos.
  + Frecuencia de palabras (Top 10).

**Chatbot (Tab “📱 Chatbot”)**

* Chat interactivo con modelo LLM (gpt-4o).
* Ajuste de **temperatura** para controlar creatividad/respuesta del modelo.
* Registro de conversación en st.session\_state.messages.
* Respuestas dinámicas en la interfaz de Streamlit.

Flujo de Uso:

1. Configurar .env **GITHUB\_TOKEN** y **OPENAI\_API\_KEY**
2. Ejecutar streamlit run main.py
3. En la pestaña **DOCUMENTOS:**

* Revisar y agregar documentos necesarios.
* Generar embeddings

1. En la pestaña **Consulta:**

* Escribir la pregunta.
* Ajustar opciones (top\_k, evaluación automática, logging.)
* Ejecutar consulta para obtener respuesta y métricas.

1. En la pestaña evaluación:

* Ejecutar evaluación sistemática para medir fidelidad, relevancia y precisión.

1. En la pestaña Analytics:

* Revisar gráficos y estadísticas.
* Exportar datos en JSON o CSV.

1. En la pestaña Chatbot:

* Interactuar en tiempo real con el LLM.

Limitaciones Actuales

* El chatbot no está conectado directamente a bases de datos en tiempo real.
* La evaluación de calidad depende únicamente de LLM (gpt-4o) y documentos cargados en memoria.
* El sistema no incluye autenticación de usuarios ni manejo de permisos.
* Los documentos deben ser cargados manualmente por el usuario.

Consideraciones de Seguridad

* No almacenar claves en código fuente; usar siempre .env.
* Las respuestas generadas están basadas solo en los documentos cargados; no reemplazan la orientación médica profesional.
* No se registran datos sensibles fuera del st.session\_state local, salvo si se exportan manualmente.

Posibles Mejoras Futuras (No implementadas)

* Integración con bases de datos de pacientes.
* Autenticación y control de acceso.
* Conexión en tiempo real con protocolos actualizados.
* Integración de RAG avanzado con fuentes externas.