Challenge - Atom

Ejercicio Propuesto: AI Agent de Voz para Nutrición de Leads

Descripción: Desarrollar un **AI Agent de voz** (utilizando Python) que emplee un modelo de lenguaje preentrenado para construir un asistente virtual capaz de:

- **Interactuar con prospectos** a través de voz para recopilar información relevante sobre sus necesidades y perfil.
- **Gestionar y analizar datos de los leads** para mejorar la segmentación y personalización.
- **Manejar entradas en lenguaje natural**, identificando información clave y respondiendo de manera útil y precisa.

Requisitos:

1. **Integración de un LLM:**
    - Utilizar un modelo de lenguaje como GPT o Whisper (para conversión de voz a texto) para procesar y responder preguntas.
    - Demostrar cómo cargar el modelo, utilizarlo para generar respuestas y personalizarlo si es necesario.
    - Se recomienda el uso de **Langchain** o un framework similar para facilitar la orquestación de los modelos y la gestión del flujo de conversación.
2. **Procesamiento de lenguaje natural (NLP):**
    - Implementar funcionalidades para entender la intención del usuario (por ejemplo, interés en un producto, dudas sobre servicios, requerimientos específicos, etc.).
    - Extraer información clave como nombre, empresa, necesidades, presupuesto estimado, etc.
3. **Interacción por voz:**
    - Implementar **reconocimiento de voz (ASR)** para convertir las respuestas habladas del prospecto en texto.
    - Implementar **síntesis de voz (TTS)** para generar respuestas habladas y guiar la conversación.
    - Mantener el contexto en conversaciones de múltiples turnos.
4. **Flujo de diálogo inteligente:**
    - Desarrollar un flujo de conversación que **nutra al lead** guiándolo en la interacción.

    - Integrar información relevante a lo largo de la conversación de manera eficaz para enriquecer el perfil del prospecto.
5. **Integración con CRM o Base de Datos:**
    - Implementar la capacidad de almacenar y actualizar información en un CRM o base de datos basada en las interacciones con los leads.
    - Manejar autenticación, interacción con la API del CRM y sincronización de datos.
6. **Manejo de errores y ambigüedades:**
    - Implementar estrategias para manejar errores en la entrada del usuario o respuestas ambiguas.
    - Solicitar clarificaciones de manera natural cuando sea necesario.
7. **Testing:**
    - Escribir pruebas unitarias para verificar la funcionalidad de los diferentes componentes del sistema.
    - Incluir pruebas para la integración con el CRM o base de datos.

Entregables:

- Código fuente en Python con sus pruebas unitarias y documentación correspondiente (Repositorio en Github o similar. Recordar que el repositorio debe ser público).
- Ideal: Presentar una **UI mínima** para interacción por voz (Ejemplo: utilizando Reetol, Streamlit, o una API con FastAPI/Flask y una interfaz web).
- Ejemplos de sesiones de diálogo por voz demostrando la funcionalidad, incluyendo la interacción con el CRM o base de datos.

Bonus:

- Implementar la opción de cambiar entre texto y voz en la interacción.
- Permitir personalización del asistente según usuario (preferencias, historial de interacción, etc.).