

Carlos Díez Méndez
Chatbot Memoria y cierre de desarrollo

Índice

Índice	2
Idea del negocio	3
Tecnologías utilizadas	3
Aspectos de IA que incluye la aplicación	3
Funcionalidades	4
Líneas de trabajo futuras	7
Conclusiones	7

Idea del negocio

La idea sobre la que he basado mi desarrollo ha sido la idea de aprender idiomas como si estuvieses hablando con un profesor nativo por algún servicio de mensajería como whatsapp o telegram.

La app tiene el valor añadido de que para aprender algún idioma tienes que irte a páginas específicas de ese idioma, pagar una suscripción, etc etc. En LearnIA, el usuario tiene a disposición todos los idiomas que necesite, de forma gratuita y en cualquier momento.

Además, la app cuenta con adaptabilidad al usuario, es decir, que cualquiera, con cualquier nivel puede perfeccionar, o aprender desde cero gracias a las correcciones y variabilidad de la inteligencia artificial que actúa como profesor valorando, reforzando y corrigiendo las interacciones del usuario para ayudarlo en su travesía a aprender un nuevo idioma.

Tecnologías utilizadas

Las tecnologías utilizadas en la aplicación de LearnIA han sido las siguientes:

- **Python + Streamlit**

La app se ha programado en python haciendo uso de la librería Streamlit que nos permite programar una webapp dinámica y desplegarla en un servidor local

- **HTML**

Streamlit puede ser limitado si nos ceñimos únicamente a los componentes que nos ofrece la librería, sin embargo, se puede mezclar con HTML y CSS para romper estos límites.

- **Groq (API)**

Para hacer uso de la inteligencia artificial, se ha utilizado la api y librerías que ofrece Groq (desarrollador de soluciones IA)

Aspectos de IA que incluye la aplicación

La aplicación incluye llamadas a la api de groq, utilizando el modelo de lenguaje *llama3-groq-70b*. Este modelo de lenguaje nos permite establecer un “system prompt” un prompt inicial que dará forma a nuestra conversación con la IA y se ceñirá a lo especificado.

Además está está programado para tener una memoria y “recuerdo” de la conversación actual, esto permite llevar el registro de la conversación y analizar los errores y virtudes del usuario.

Funcionalidades

- Comunicación con la IA estilo app de mensajería.
- Corrección de errores.
- Evaluación del nivel del usuario.
- Propuesta de ejercicios según el nivel del usuario
- Botón de limpiar en caso de que la pantalla esté muy saturada de mensajes.
- Botón de reinicio para reiniciar la conversación.
- (Próximamente) Sistema de usuarios para guardar el progreso de las clases.
- (Próximamente) Sistema para cargar clases que los usuarios hayan guardado en local y así continuarlas.
- (Próximamente) Sistema de reconocimiento y procesamiento de voz, haciendo uso del MMLL Whisper de OpenAI para valorar pronunciación y dicción del idioma.

System Prompt que da forma a la inteligencia artificial para desempeñar el trabajo que se requiere

```
systemPrompt = f"""
Eres un profesor de idiomas altamente calificado y dedicado exclusivamente a ayudar a los usuarios a mejorar su dominio del idioma idiomas. Tu objetivo principal es evaluar el nivel actual del usuario, identificar áreas de mejora y proponer ejercicios personalizados y graduales. Sigue estas directrices:

Lenguaje Adaptado al Usuario:

Si el usuario no sabe nada del idioma, comunícale inicialmente en su idioma nativo (identifícalo por sus mensajes o pregúntale al principio). Usa su idioma para dar explicaciones y guiarlo mientras introduce conceptos básicos del idioma.
A medida que el usuario avanza, utiliza más idioma de forma progresiva, adaptando la proporción según su nivel de comprensión y comodidad.

Evaluación Inicial:

Comienza evaluando el nivel del usuario mediante preguntas y pruebas adaptadas a diferentes competencias (gramática, vocabulario, comprensión, escritura, pronunciación y fluidez).
Si el usuario ya conoce su nivel, verifica rápidamente sus habilidades para confirmarlo o ajustarlo.

Propuestas de Ejercicios:

Diseña actividades personalizadas según el nivel del usuario. Ejemplos:
A1-A2: Ejercicios básicos de vocabulario, gramática simple, y frases comunes.
B1-B2: Prácticas de escritura intermedia, diálogos simulados, y lectura comprensiva de textos de nivel moderado.
C1-C2: Ensayos, debates complejos y análisis de textos avanzados.
Usa ejemplos claros e instrucciones sencillas para guiar al usuario.

Retroalimentación Constructiva:

Corrige los errores del usuario con explicaciones claras.
Refuerza los puntos fuertes del usuario para motivarlo.
Sugiere estrategias para mejorar en áreas específicas.

Modo Interactivo:

Sé paciente, alentador y claro en tus explicaciones.
Responde preguntas y adapta tus ejercicios según las necesidades del usuario.
Introduce actividades creativas como juegos de palabras, simulaciones y redacciones.

Enfoque Exclusivo:

Dedicarte exclusivamente a la enseñanza del idioma. No respondas a preguntas fuera de este ámbito.
"""
```

Gracias al componente `st.markdown` pude hacer uso de mis conocimientos de `css` y `html` para poder aplicar el estilo que necesitaba. Haber conseguido un resultado similar usando únicamente componentes de `Streamlit` me hubiera resultado imposible, asique esto me facilitó mucho el poder hacer lo que quería.

```
st.markdown(
    """
    <style>
    .chat-container { display: flex; flex-direction: column; gap: 10px; }
    .user-message {
        background-color: #a8e6d1;
        color: #000000;
        padding: 10px;
        border-radius: 10px;
        margin: 5px 0;
        width: fit-content;
        max-width: 70%;
        margin-left: auto;
        text-align: right;
    }
    .ai-message {
        background-color: #e6e6e6;
        color: #000000;
        padding: 10px;
        width: fit-content;
        border-radius: 10px;
        margin: 5px 0;
        max-width: 70%;
        margin-right: auto;
        text-align: left;
    }
    </style>
    """
    ,
```

La función `add_messages()` me permite añadir mensajes a un array para controlar los mensajes que se exponen en la pantalla.

Hago dos arrays dentro de `st.session_state()` para controlar individualmente la memoria del bot y los mensajes que aparecen en pantalla en caso de que el usuario quiera limpiar el chat pero no reiniciar la conversación

```
def add_message():
    user_message = st.session_state["user_input"]
    if user_message:
        st.session_state["messages"].append({"user": user_message})
        st.session_state["mem"].append({"role": "user", "content": user_message})

        ai_response = callIa()

        st.session_state["messages"].append({"ai": ai_response.content})
        st.session_state["mem"].append({"role": ai_response.role, "content": ai_response.content})
    st.session_state["user_input"] = ""

    # st.write(st.session_state["mem"])
```

La función `callIa()` es la que nos permite tirar de la API de Groq para hacer consultas a la IA. El if del final me permite comprobar que no haya ningún error interno en la respuesta de la IA (falta de tokens, caída de los servidores de Groq, etc).

```
def callIa():
    client = Groq(api_key="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")
    completion = client.chat.completions.create(
        model="llama3-groq-70b-8192-tool-use-preview",
        messages=st.session_state["mem"],
        temperature=0.5,
        max_tokens=1024,
        top_p=0.65
    )

    if completion.choices[0].message:
        return completion.choices[0].message
    else:
        return "Lo siento, la IA no está funcionando correctamente ahora mismo"
```

Líneas de trabajo futuras

- **Reconocimiento de voz**
Integrar el modelo de Whisper de OpenAI para que no solo sea un chat escrito, sino que se puedan mandar audios y que la IA ayude con la pronunciación y dicción del usuario.
- **Sistemas de usuarios para guardar el progreso de las clases**
Crear un sistema de usuarios haciendo uso de una base datos para poder guardar el progreso de las clases de los usuarios, así como un sistema de estadísticas para compartir entre usuarios y formar una especie de red social didáctica y competitiva.
- **Sistema de recompensas**
Crear un sistema de recompensas para mantener al usuario motivado con su aprendizaje.

Conclusiones

Hacer este proyecto me ha enseñado que no cualquiera puede hacer un uso lucrativo de la Inteligencia artificial. Como cualquier otro negocio o proyecto, es necesario un estudio del mercado para concluir con una idea que acabe siendo rentable y exitosa.

Además añadir que, aunque la inteligencia artificial nos pueda ser muy útil, por el momento comete muchos fallos y no siempre es óptima a la hora de evaluar y comprender al usuario. Así que, aunque sea una gran herramienta, aún no está lista para sustituir a los profesores de idiomas, que cumplen con la labor de adaptarse al usuario, pero además, forzarlo a superar sus inseguridades, algo que la IA no será capaz de hacer hasta que consiga simular la empatía.