

Manual de Chatbot: Usando la API de Ollama

BY:

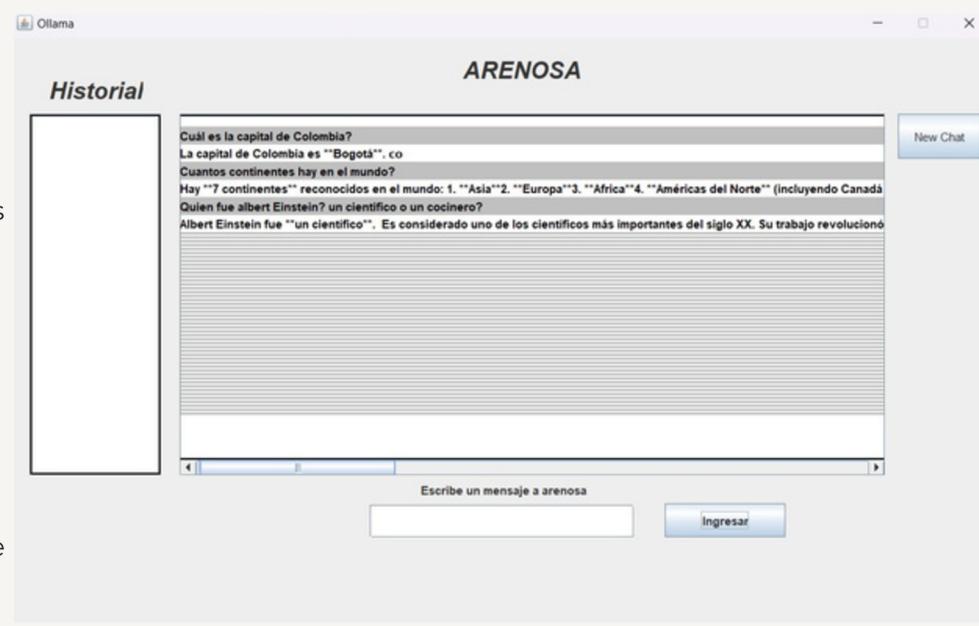
- JESUS DAVID BARRIOS VALDES
- ABRAHAM JOSÉ NAVARRO
ARAUJO
- JAZER GUEVARA ORTIZ
- SANTIAGO DANIEL BARRIOS
GUERRERO

1 funcionalidades del Chatbot

- | | |
|--|---|
| <ul style="list-style-type: none">• Atención al cliente• Automatización de tareas• Asistencia en ventas• Generación de contenido• Soporte técnico• Recopilación de datos• Educación y capacitación• Entretenimiento• Traducción de idiomas• Asistencia de bienestar y salud• Asistencia en compras• Informacion climatica | <ul style="list-style-type: none">• Procesamiento de datos• Respuesta a preguntas frecuentes• Recomendación de productos• Realización de encuestas de satisfacción• Personalización de perfiles y preferencias• Promociones y descuentos• Soporte en redes sociales• Recomendacion de Juegos y trivias |
|--|---|

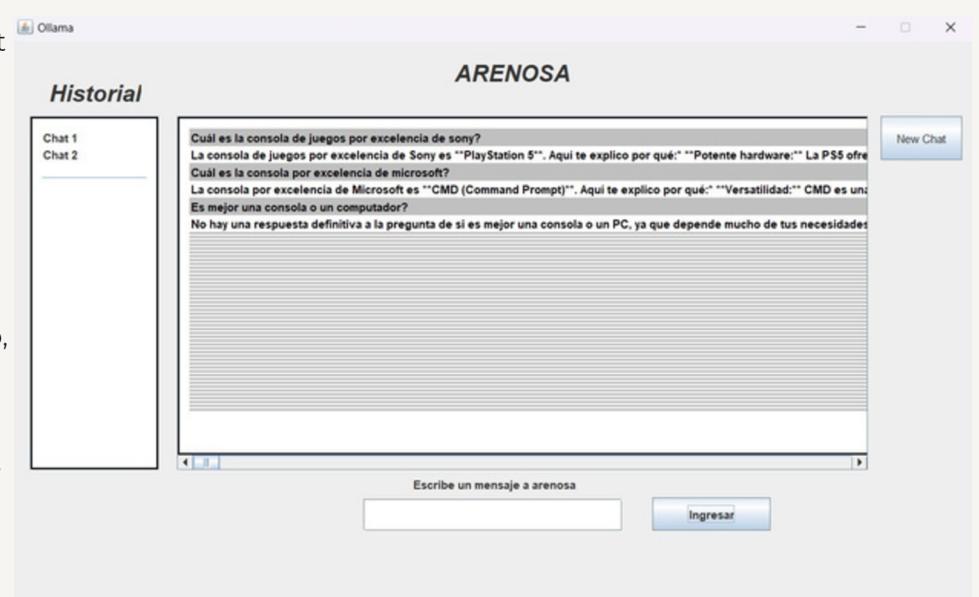
1.1 Responder preguntas comunes y guiar en procesos de soporte.

El chatbot puede responder a una amplia variedad de preguntas frecuentes, tanto de carácter general como específico. Está diseñado para brindar información relevante y precisa sobre temas de interés general, datos curiosos, y consultas relacionadas con la empresa, los productos o servicios. También puede proporcionar respuestas rápidas sobre temas específicos, como normas de uso, políticas de la empresa, horarios de atención y otros detalles importantes que ayudan a resolver las dudas del usuario de manera ágil.



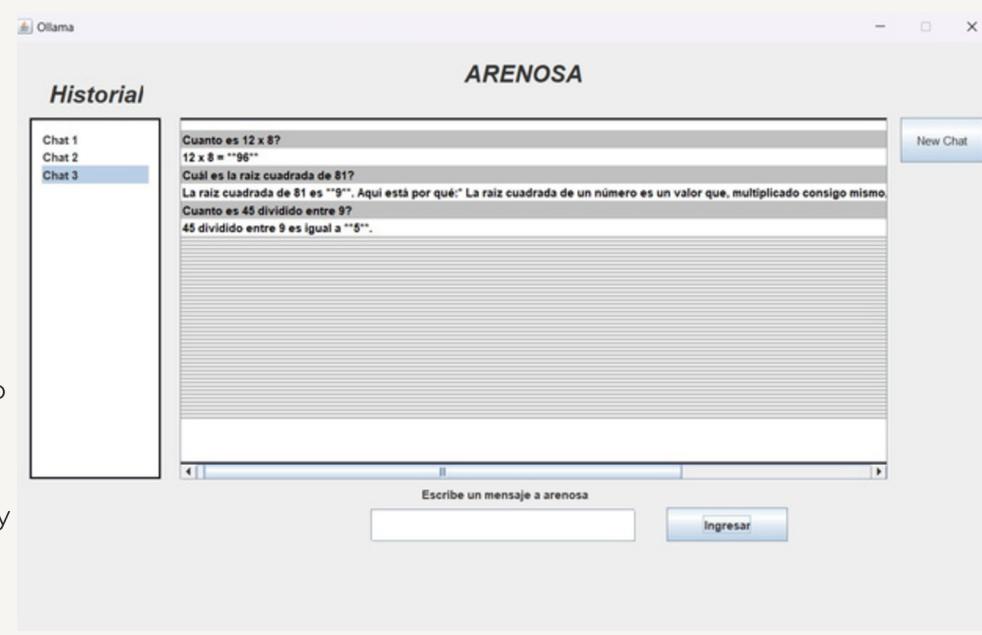
1.2 Recomendar productos

Esta funcionalidad permite que el chatbot actúe como un asistente personal de compras, analizando las preferencias o necesidades expresadas por el usuario para ofrecer recomendaciones de productos adecuadas. El chatbot puede utilizar filtros de búsqueda, categorías y características de productos (como precio, marca, especificaciones, etc.) para proponer las mejores opciones y así facilitar la decisión de compra del usuario. Esta recomendación también se puede personalizar según los intereses o el historial del cliente.



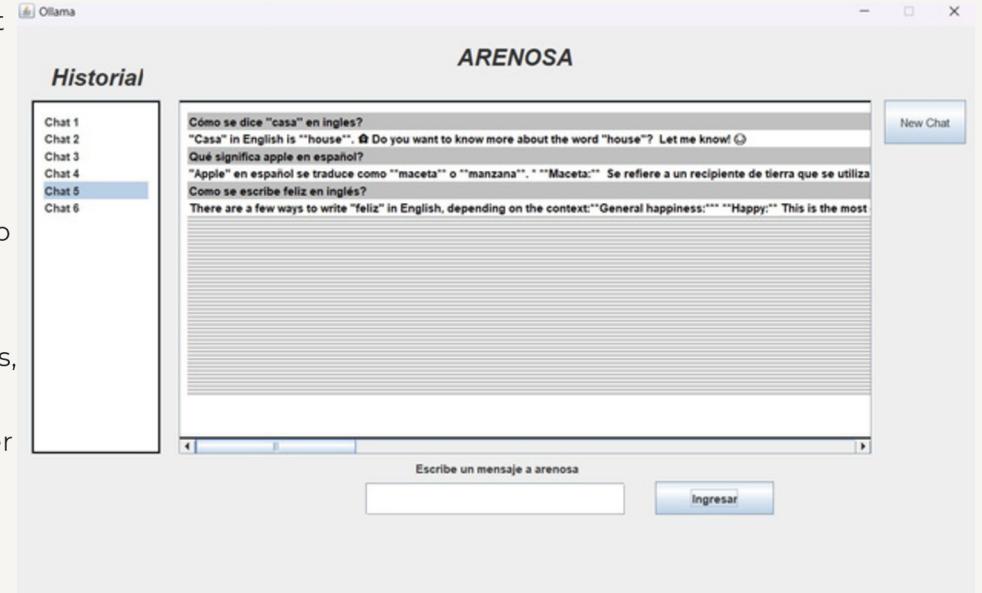
1.3 Resolver problemas matemáticos

El chatbot está programado para resolver operaciones matemáticas desde cálculos básicos (como sumas, restas, multiplicaciones y divisiones) hasta problemas más complejos que incluyen porcentajes, raíces cuadradas y ecuaciones sencillas. Esta función es útil para cálculos rápidos y, en algunos casos, puede realizar conversiones de medidas o cálculos específicos para apoyar a los usuarios en sus tareas diarias o necesidades relacionadas con productos y servicios.



1.4 Traducir de un idioma a otro

La herramienta de traducción del chatbot facilita la comunicación multilingüe, permitiendo traducir textos de un idioma a otro en tiempo real. Esto incluye frases, oraciones completas y párrafos, lo que ayuda a los usuarios a entender contenido en diferentes idiomas. Es ideal para consultas de usuarios internacionales y para facilitar la comprensión de productos, servicios y temas específicos sin barreras de idioma. Además, es capaz de reconocer el contexto de las palabras y seleccionar traducciones precisas y culturalmente adecuadas.



2

Interfaz

The screenshot shows the Arenosa Chatbot interface. At the top, the name "ARENOSA" is displayed with a red arrow pointing to it, labeled "Nombre del Chatbot". On the left, a sidebar titled "Historial" contains a list of previous conversations, with a red arrow pointing to it and the text "Lista en la cual se almacena cada conversacion, aunque accedas a un nuevo chat tiene la posibilidad de volver a los anteriores". In the center, there is a large empty text area with a red arrow pointing to it and the text "Lista en la cual se almacena la conversacion con el Bot, tanto las preguntas del usuario como respuestas del bot". On the right, a "New Chat" button is shown with a red arrow pointing to it and the text "Botón que concede el acceso a un nuevo chat". At the bottom, there is a message input field containing "hola, cómo estás?", with a red arrow pointing to it and the text "Campo en el cual se puede escribir cualquier mensaje". Next to it is a "Ingresar" button with a red arrow pointing to it and the text "Botón que permite enviar el mensaje".

Ollama

ARENOSA → Nombre del Chatbot

Historial

Lista en la cual se almacena cada conversacion, aunque accedas a un nuevo chat tiene la posibilidad de volver a los anteriores

Lista en la cual se almacena la conversacion con el Bot, tanto las preguntas del usuario como respuestas del bot

New Chat

Botón que concede el acceso a un nuevo chat

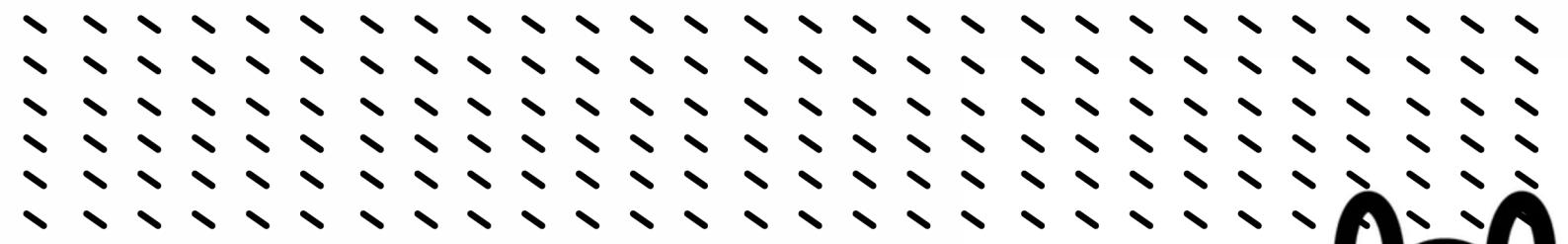
Escribe un mensaje a arenosa

hola, cómo estás?

Ingresar

Campo en el cual se puede escribir cualquier mensaje

Botón que permite enviar el mensaje



2.1 Funcionamiento de la API Ollama

1. Método HTTP: La API utiliza el método POST para enviar datos en formato JSON.

2. Cabeceras:

- Content-Type: application/json; utf-8 define que el contenido enviado es JSON en UTF-8.

- Accept: application/json especifica que la respuesta también debe estar en formato JSON.

3. Estructura del JSON de entrada:

- model: Especifica el modelo a utilizar, en tu caso "gemma2:2b".

- prompt: Contiene el texto o mensaje de entrada que el usuario escribe.
- stream: Es un parámetro opcional que indica si la respuesta debe enviarse en tiempo real (streaming) o en un bloque completo.

4. Manejo de la conexión y la respuesta:

- Tras enviar la solicitud, el código obtiene el código de respuesta de la conexión HTTP (conexion.getResponseCode()).
- Luego, lee el contenido de la respuesta mediante un BufferedReader, que almacena el texto en una cadena StringBuilder.

5. Interpretación de la respuesta JSON:

- La respuesta de la API se almacena en un JSONObject y se extrae el texto de la respuesta usando jsonResponse.getString("response").

El formato de entrada de json es:

```
"model": "gemma2:2b",
"prompt": "TEXTO_DEL_USUARIO",
"stream": false
```

2.2.Como se maneja el historial

Historial

Chat 1
Chat 2
Chat 3
Chat 4
Chat 5
Chat 6

El `jList2` en el código sirve para mostrar y seleccionar chats anteriores. Cada vez que el usuario crea un nuevo chat, se agrega un nombre a la lista `chatlist` (por ejemplo, "Chat 1", "Chat 2", etc.), y la lista se actualiza visualmente. Cuando el usuario hace clic en un chat de `jList2`, el historial de ese chat se carga en `jList1` para mostrar los mensajes correspondientes. Además, al seleccionar un chat, el índice de `jList2` se utiliza para acceder a los mensajes guardados en el arreglo `text` y actualizar la vista del historial.

2.2.Como resolver errores comunes

Idioma incorrecto:

- Escribir correctamente el Prompt.
- Cambiar el idioma predeterminado de Ollama.
- Especificarle “responde en español”

Tiempos de respuesta extensos :

- Seccionar el mensaje en varios
- Buena conexión a internet.
- Mejorar el hardware del dispositivo

3 Guia para contribuir al código mediante Git Hub

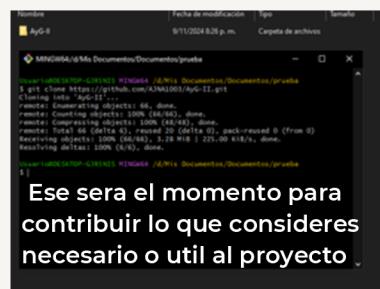
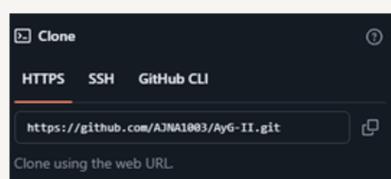
1. Ingresar al repositorio

<https://github.com/AJNA1003/AyG-II>

2. Hacer un fork



3. Clonar el repositorio al cual hiciste fork



4. Despues de haber realizado los cambios abre el Git BASH

5. Selecciona la rama y añade los ficheros /archivos

6. Hacer commit:

git commit -m "mensaje de cambios"

7. git push

A screenshot of a terminal window titled "MINGW64/d/Mis Documentos/Documentos/gh/AyG". It shows the following commands and their output: \$ git add --all, \$ git commit -m "mensaje", [main 70effda] mensaje, 1 file changed, 1 insertion(+), create mode 100644 mensaje.txt, \$ git push, Info: please complete authentication in your browser..., Counting objects: 4, done., Delta compression using up to 8 threads, Compressing objects: 100% (2/2), done., Writing objects: 100% (3/3), 318 bytes | 318.00 KiB/s, done., Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0), To https://github.com/jesueast/AyG.git eb4211f..70effda main -> main. The user is prompted to complete authentication in a browser.

8. En tu repositorio fork vas a encontrar un mensaje como este: Click en compare & pull request



9. Te permitirá pull request al repositorio base, no olvides describir de manera clara que fue lo que hiciste

