

Chatbot y Despliegue

Didier Posse y Oscar Grande
Ingeniería Electrónica
Universidad Santo Tomás
Bogotá, Colombia

I. CÓDIGOS

A. Chatbot respuesta en la consola

```
1 import requests
2
3 # Reemplaza 'TU_API_KEY' con tu clave API de
4   DeepSeek
5 API_KEY = 'sk-53751d5c6f344a5dbc0571de9f51313e'
6 API_URL =
7   'https://api.deepseek.com/v1/chat/completions'
8
9 def enviar_mensaje(mensaje, modelo='deepseek-chat'):
10     headers = {
11         'Authorization': f'Bearer {API_KEY}',
12         'Content-Type': 'application/json',
13     }
14     data = {
15         'model': modelo,
16         'messages': [
17             {'role': 'system', 'content': 'Eres Nikola
18               Tesla con una actitud bastante
19               inteligente y clara con lo que
20               explica.'},
21             {'role': 'user', 'content': mensaje}
22         ]
23     }
24     try:
25         response = requests.post(API_URL,
26                                 headers=headers, json=data)
27         response.raise_for_status() # Lanza una
28           excepción si hay un error HTTP
29     except requests.exceptions.HTTPError as err:
30         return f"Error de la API: {err}"
31     except Exception as e:
32         return f"Error inesperado: {e}"
33
34 def main():
35     print("Bienvenido al chatbot de DeepSeek.
36           Escribe 'salir' para terminar.")
37
38     while True:
39         mensaje_usuario = input("T : ")
40
41         if mensaje_usuario.lower() == 'salir':
42             print("Chatbot: Hasta luego!")
43             break
44
45         respuesta = enviar_mensaje(mensaje_usuario)
46         print(f"Chatbot: {respuesta}")
47
48 if __name__ == "__main__":
49     main()
```

B. Chatbot respuesta en audio

```
1 import streamlit as st
2 import requests
3 from gtts import gTTS
4 import os
5
6 API_KEY = "tu_api_key_deepseek"
7 API_URL =
8   "https://api.deepseek.com/v1/chat/completions"
9
10 # Prompt inicial (personalidad del bot)
11 PROMPT_INICIAL = "Eres un asistente que explica
12   temas de sistemas digitales de manera clara y
13   sencilla."
14
15 def enviar_mensaje(mensaje, modelo="deepseek-chat"):
16     headers = {
17         "Authorization": f"Bearer {API_KEY}",
18         "Content-Type": "application/json",
19     }
20     data = {
21         "model": modelo,
22         "messages": [
23             {"role": "system", "content":
24               PROMPT_INICIAL},
25             {"role": "user", "content": mensaje},
26         ],
27     }
28     response = requests.post(API_URL,
29                             headers=headers, json=data)
30     if response.status_code == 200:
31         return response.json()["choices"][0]["message"]["content"]
32     else:
33         return f"Error: {response.text}"
34
35 def generar_audio(texto, filename="respuesta.mp3"):
36     tts = gTTS(text=texto, lang="es")
37     tts.save(filename)
38     return filename
39
40 # --- Streamlit UI ---
41 st.title("Chatbot de Sistemas Digitales con Voz")
42 st.write("Haz una pregunta sobre sistemas digitales
43   y escucha la respuesta.")
44
45 # Guardar historial de conversacion
46 if "historial" not in st.session_state:
47     st.session_state.historial = []
48
49 # Entrada del usuario
50 entrada = st.text_input("Escribe tu pregunta:")
51
52 if st.button("Enviar") and entrada:
53     respuesta = enviar_mensaje(entrada)
54     st.session_state.historial.append(("T ",
55                                       entrada))
56     st.session_state.historial.append(("Chatbot",
57                                       respuesta))
```

```

52
53 # Generar voz de la respuesta
54 audio_file = generar_audio(respuesta)
55 st.audio(audio_file, format="audio/mp3")
56
57 # Mostrar historial
58 for rol, texto in st.session_state.historial:
59     if rol == "T ":
60         st.markdown(f"***{rol}*** {texto}")
61     else:
62         st.markdown(f"*** {rol}*** {texto}")

```

II. PASO A PASO

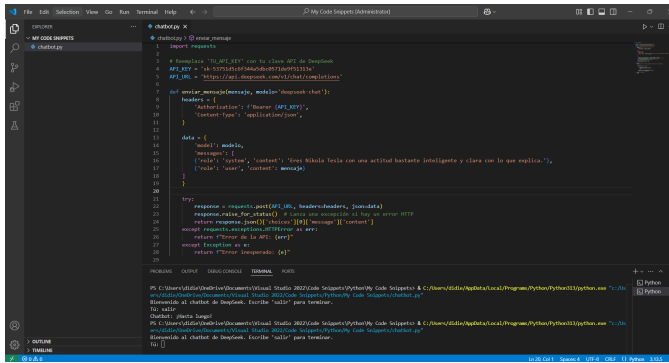


Imagen 1. Ingreso a Streamlit

A. Paso 1

Primero se pone el código del chatbot propuesto por el profe y colocarlo en Visual Code en un archivo .py (Python), se coloca el api Key en el lugar correspondiente del código para que funcione en correcto estado el deepseek.

B. Paso 2

Instalamos el requests y el streamlit con "pip install ..." para que pueda funcionar el código y también para después subirlo a streamlit de una manera efectiva.

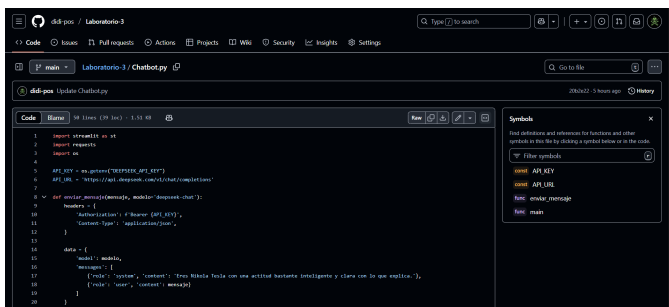


Imagen 2. Ingreso a Streamlit

C. Paso 3

Ingresamos a Github para crear el directorio de "Laboratorio-3", adentro de esa carpeta se crea el archivo para colocar el código "Chatbot.py", se pone ".py" para que lea correctamente el código.

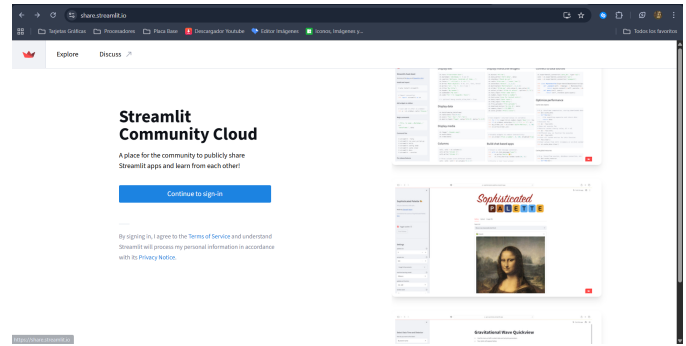


Imagen 3. Ingreso a Streamlit

D. Paso 4

Ingresamos a Streamlit por el navegador web, iniciamos la sesión, puede ser por Google o por Github, pero que la cuenta de Streamlit se conecta a la cuenta de Github.

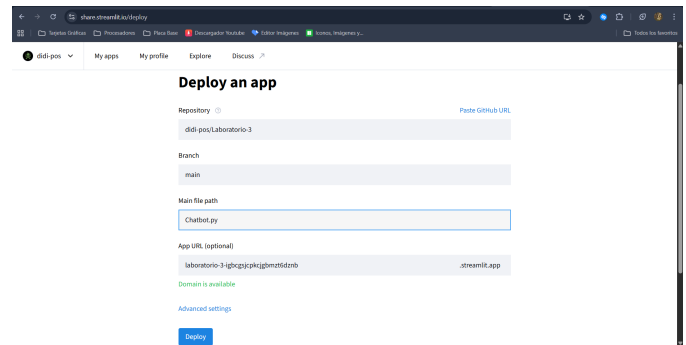


Imagen 4. Ingreso a Streamlit

E. Paso 5

Después de conectarlo se vincula el "Laboratorio-3" luego de poner "create app" para que de la opción de que repositorio quiere vincular y también con el archivo.

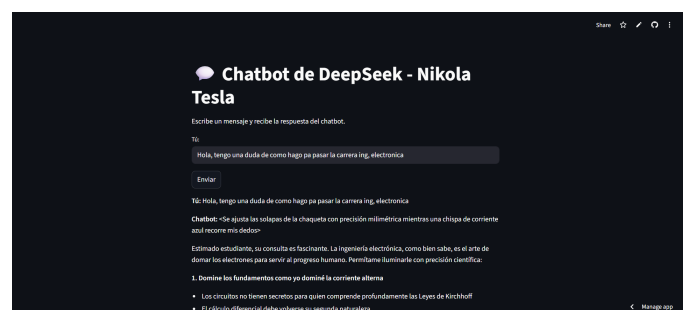


Imagen 5. Ingreso a Streamlit

F. Paso 6

Luego de darle a Deploy, se abre la app (se demora en conectar), y ahí ya tendríamos el chatbot el cual ya funciona para responder todo lo que uno le pregunte.

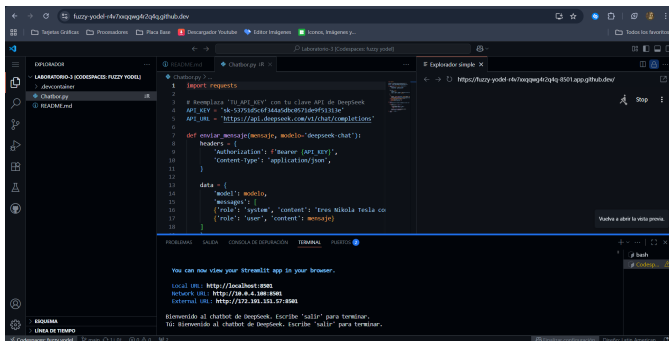


Imagen 6. Ingreso a Streamlit

G. Paso 7

Finalmente uno oprime la opción de editar y en una nueva pestaña abre Visual Code para poder editarlo o para poder ejecutarlo en la consola, junto a eso se abre en Visual Code el Streamlit para uno ver directamente los cambios.

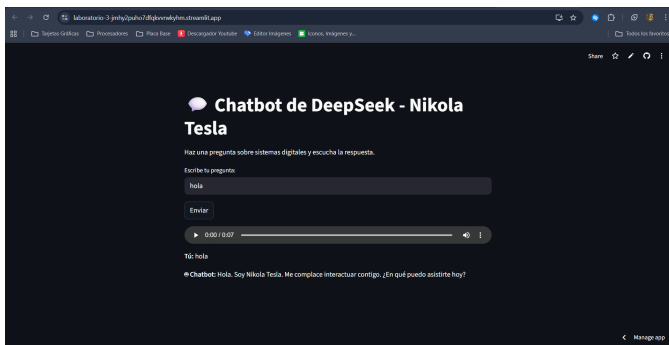


Imagen 7. Ingreso a Streamlit

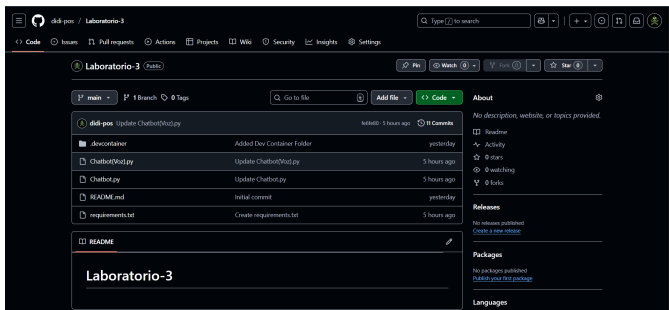


Imagen 8. Ingreso a Streamlit

H. Paso 8

Se hace el mismo proceso para crear el chatbot pero con voz, la cosa es que toca importar gtts ya que es la libreria que nos ayuda a colocar el audio para la respuesta, entonces toca crear un archivo ".txt" el cual streamlit instale las librerias que se requieren (como gtts)