

Práctica de Laboratorio 3: Chatbot con DeepSeek y Streamlit

Tu Nombre

5 de septiembre de 2025

1. Objetivo

Desarrollar un chatbot interactivo sobre sistemas digitales usando la API de DeepSeek y desplegarlo en Streamlit.

2. Requisitos Previos

- Python 3.7+ instalado
- Cuenta en GitHub
- Cuenta en Streamlit

3. Desarrolla tu propio chatbot con DeepSeek

1. Configuración inicial de la API

```
import requests

# Configuración de la API
APIKEY = 'sk-53751d5c6f344a5dbc0571de9f51313e'
APIURL = 'https://api.deepseek.com/v1/chat/completions'
```

2. Función para enviar mensajes a la API

```
def enviar_mensaje(mensaje, modelo='deepseek-chat'):
    headers = {
        'Authorization': f'Bearer-{APIKEY}',
        'Content-Type': 'application/json'
    }
```

2. Función para enviar mensajes a la API

```
data = {
    'model': modelo,
    'messages':
        [{ 'role': 'user', 'content': mensaje }]
}

try:
    response = requests.post
        (API_URL, headers=headers, json=data)
    response.raise_for_status()
    return response.json
        ()['choices'][0]['message']['content']
except Exception as e:
    return f'Error: {str(e)}'
```

3. Lógica principal del chatbot

```
def main():
    print
        ("Bienvenido al chatbot de DeepSeek
        ---- Escribe 'salir' para terminar.")

    while True:
        mensaje_usuario = input("Tu: ")

        if mensaje_usuario.lower() == 'salir':
            print("Chatbot: ~ Hasta ~ luego!")
            break

        respuesta = enviar_mensaje(mensaje_usuario)
        print(f"Chatbot: ~ {respuesta}")

if __name__ == '__main__':
    main()
```

4. Ejecución del chatbot básico

```
# Ejecutar el script  
python chatbot_basico.py  
  
# Ejemplo de interacción:  
# Tu: Hola, ¿qué es un sistema digital?  
# Chatbot: Un sistema digital es...
```

5. Comandos para instalar dependencias

```
# Instalar solo requests para el chatbot básico  
pip install requests  
  
# Para congelar dependencias  
pip freeze > requirements.txt
```

6. Desplegar en Streamlit

1. Ingresar a <https://streamlit.io/>
2. Conectar cuenta de GitHub
3. Seleccionar repositorio y archivo `app.py`
4. Click en **Deploy**

4. Resultados Esperados

- Chatbot funcional sobre sistemas digitales
- Código disponible en repositorio público GitHub
- Aplicación web desplegada en Streamlit