



UNIVERSIDAD
SANTO TOMÁS

PRESENTACIÓN



**Ingeniero Electrónico, Magister en Ingeniería con
énfasis en electrónica y estudiante del doctorado en
ingeniería con énfasis en eléctrica y electrónica de la
UDFJC**

Diego Alejandro Barragán Vargas

**Docente de electrónica Universidad Santo Tomás de
Aquino**

Enlace de Interés:

<https://scholar.google.com/citations?hl=es&user=Bp3QMQMAAAAJ>



UNIVERSIDAD
SANTO TOMÁS

Sesión 5- Tercer Práctica de Laboratorio

11 de Julio, Bogotá D.C.

Primer Punto:

Desarrolla tu propio chatbot con DeepSeek que sea personalizado y con tu propio prompt.

EJEMPLO BÁSICO DeepSeek

```
import requests

# Reemplaza 'TU_API_KEY' con tu clave API de DeepSeek
API_KEY = 'sk-53751d5c6f344a5dbc0571de9f51313e'
API_URL = 'https://api.deepseek.com/v1/chat/completions'

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

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

    try:
        response = requests.post(API_URL, headers=headers, json=data)
        response.raise_for_status() # Lanza una excepción si hay un error HTTP
        return response.json()['choices'][0]['message']['content']
    except requests.exceptions.HTTPError as err:
        return f"Error de la API: {err}"
    except Exception as e:
        return f"Error inesperado: {e}"

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

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

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

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

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

API KEY PARA EL EJERCICIO

sk-53751d5c6f344a5dbc0571de9f51313e

Material de Apoyo

<https://www.youtube.com/watch?v=VmlNd98Nzg0>

STREAMLIT

Es una biblioteca de código abierto en Python que facilita la creación y el intercambio de aplicaciones web interactivas para ciencia de datos y aprendizaje automático.

CARACTERÍSTICAS

Crear aplicaciones web rápidamente.

Compartir las aplicaciones de manera dinámica.

Diseños interactivos dinámicos y uso de python.

Construcción rápida de dashboards y paneles de control.

Facilidad del manejo de datos y construcción de modelos de aprendizaje automático.

Paso 1:

Crear Entorno Virtual

Paso 2:

**Instalar dependencias de
Streamlit y Requests**

```
pip install requests
```

```
pip install streamlit
```

Paso 3:

**Instalar dependencias de
Streamlit y Requests**

```
pip freeze > requirements.txt
```

Ahora la idea es modificarlo y desarrollar la construcción básica de la app:

```
import streamlit as st
import requests

# Mantenemos la configuración de la API
API_KEY = 'sk-53751d5c6f344a5dbc0571de9f51313e' # Considera usar st.secrets para producción
API_URL = 'https://api.deepseek.com/v1/chat/completions'

def enviar_mensaje(mensaje, modelo='deepseek-chat'):
    # (Misma implementación que tu versión original)
    # ...
```

Gestionar el estado de la conversación:

```
# Inicializar el historial de chat
if 'messages' not in st.session_state:
    st.session_state.messages = []
```

Mostrar el historial del chat:

```
# Mostrar mensajes anteriores
for message in st.session_state.messages:
    with st.chat_message(message["role"]):
        st.markdown(message["content"])
```


Manejar la entrada del usuario:

```
# Input del usuario
if prompt := st.chat_input("Escribe tu mensaje..."):
    # Agregar mensaje de usuario al historial
    st.session_state.messages.append({"role": "user", "content": prompt})

    # Mostrar mensaje de usuario
    with st.chat_message("user"):
        st.markdown(prompt)

    # Generar respuesta
    if prompt.lower() == "salir":
        respuesta = "¡Hasta luego! 🤖"
    else:
        respuesta = enviar_mensaje(prompt)

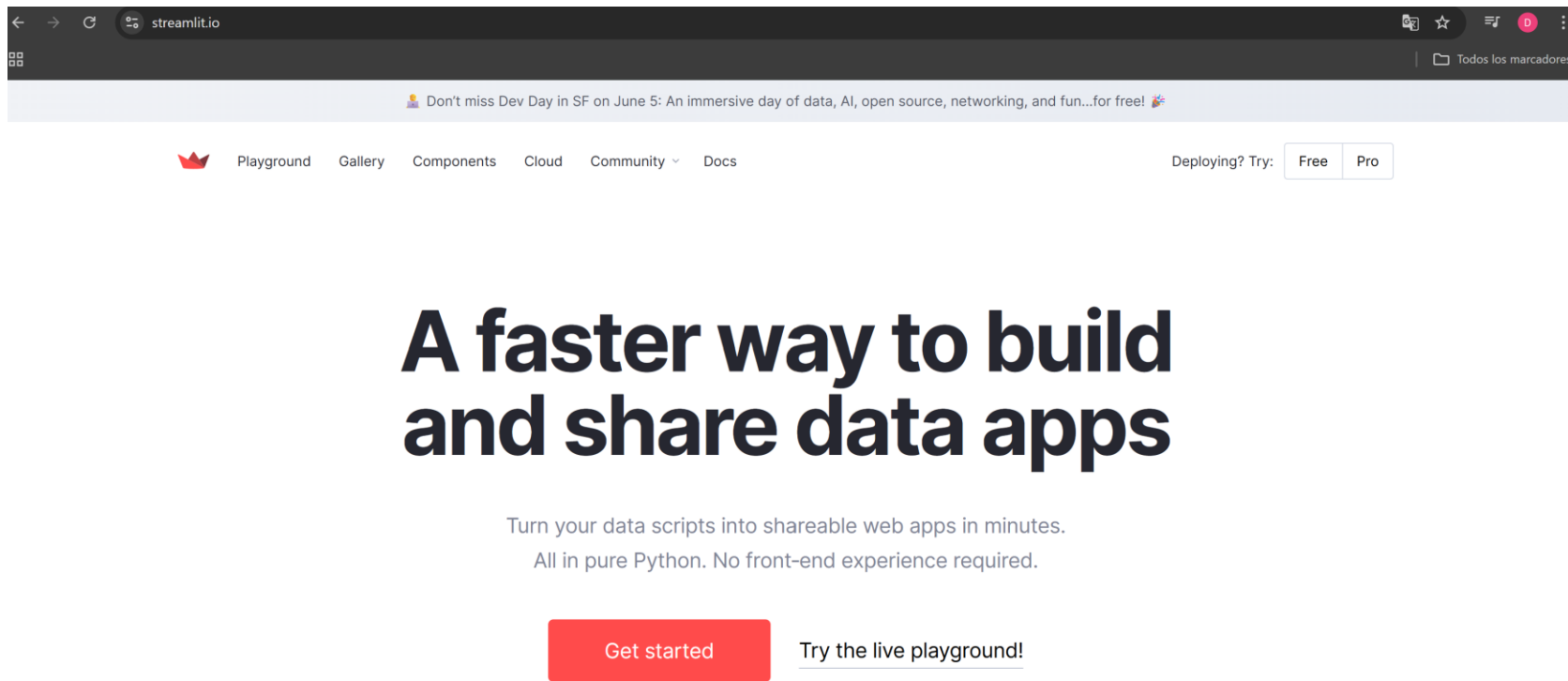
    # Mostrar y guardar respuesta
    with st.chat_message("assistant"):
        st.markdown(respuesta)
    st.session_state.messages.append({"role": "assistant", "content": respuesta})
```

Ejecución de la app:

```
streamlit run nombre_del_archivo.py
```

Posteriormente se sube a github y se deja como repositorio público

Luego se ingresa a Streamlit y se genera el importe del repositorio de github.



Segundo Punto

Desplegar en streamlit un chatbot sobre qué son los sistemas digitales y compartir para visualizar.

VOZ EN UN CHATBOT

Tutor IA Voz

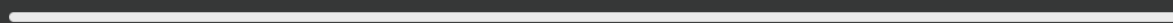
Chatbot Habilitado por Voz - DeepSeek

Start recording

Tu: comprendo me puedes por favor explicar Qué es la Inteligencia artificial generativa



3:31 / 3:31





UNIVERSIDAD
SANTO TOMÁS

GRACIAS