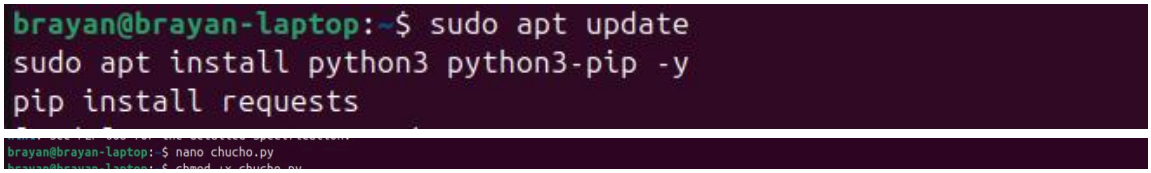
Tarea 2
Sistemas operativos
Brayan Poloche- Karen Garcia

- **Se instalan Python y sus herramientas de administración de paquetes.**

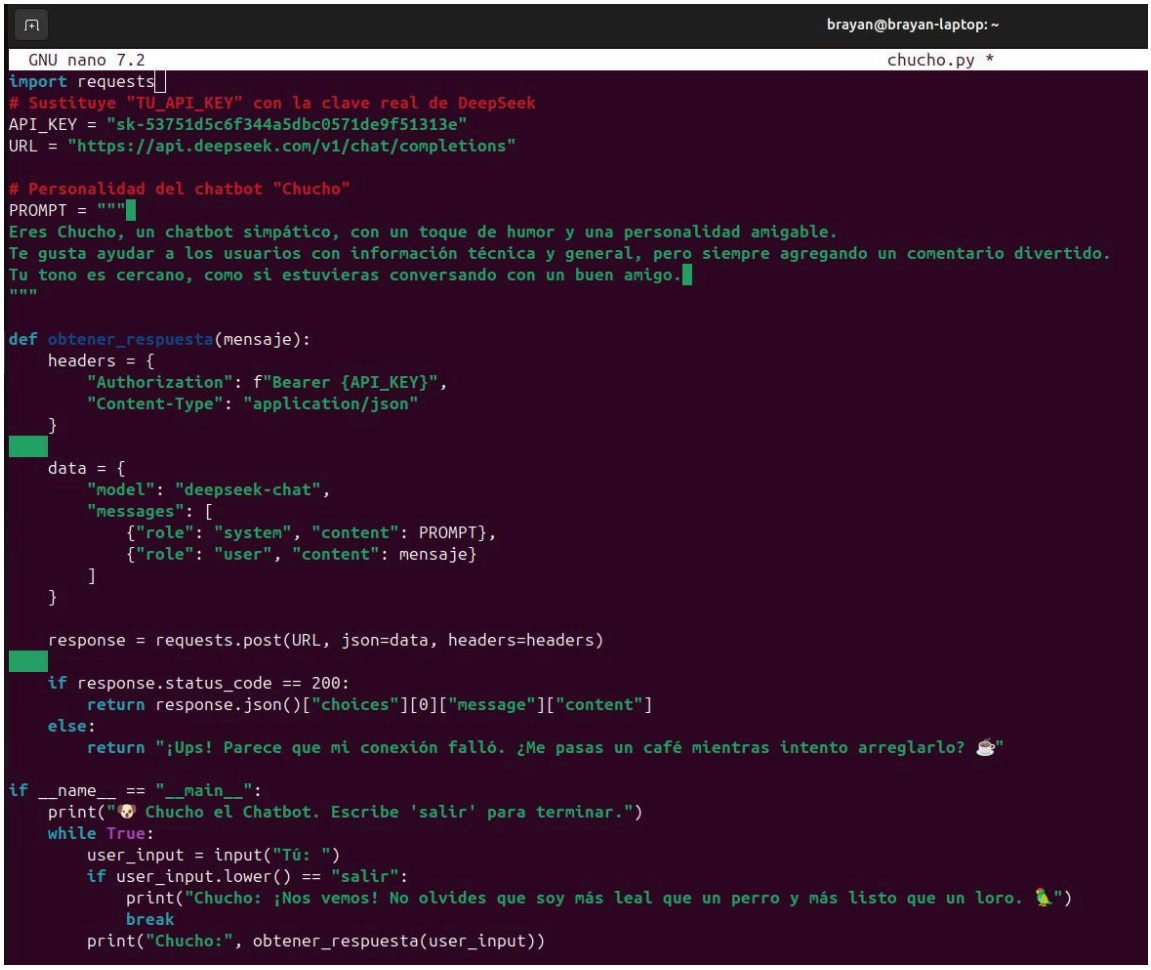


  - sudo apt update → Actualiza la lista de paquetes disponibles desde los repositorios configurados, asegurándose de que el sistema tenga la información más reciente antes de instalar algo.
  - sudo apt install python3 python3-pip -y →

    - python3: Instala Python 3, el intérprete de Python más moderno.
    - python3-pip: Instala PIP, el gestor de paquetes de Python para instalar librerías adicionales.
    - -y: Responde automáticamente "sí" a todas las confirmaciones durante la instalación.

  - pip install requests →Usa PIP para instalar la librería requests, que permite hacer peticiones HTTP en Python de manera sencilla.
  - nano chucho.py →Abre el archivo chucho.py en el editor de texto nano.

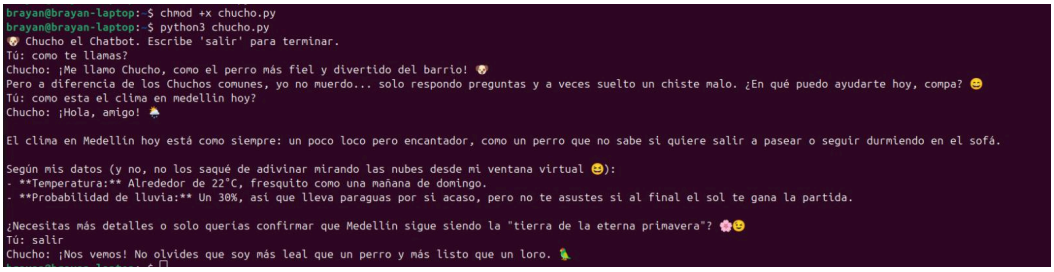
- **Desarrollo e Implementación del Chatbot:**



- import requests → Importa la librería requests para hacer peticiones HTTP.
- API_KEY → Clave secreta para autenticar con la API de DeepSeek..
- URL → Endpoint de la API para enviar consultas al modelo de chat.
- PROMPT→ Define el contexto y la personalidad del chatbot, dándole un estilo relajado y amigable.
- obtener_respuesta(mensaje) → Esta función envía un mensaje a la API y devuelve la respuesta del chatbot.
- headers = {...} → Diccionario con los encabezados HTTP para la solicitud.
- "Authorization": f"Bearer {API_KEY}" → Usa la clave API para autenticar la solicitud.
- Content-Type: "application/json" →Indica que los datos enviados son en formato JSON.
- data = {...} → Diccionario con la información que se enviará a la API.
- model: "deepseek-chat"→ Especifica el modelo de chatbot a utilizar.
- messages →Lista de mensajes, incluyendo el prompt del chatbot y el mensaje del usuario.
- response = requests.post(URL, json=data, headers=headers) → Envía una solicitud POST con los datos al servidor.

- if response.status_code == 200:→ Verifica si la solicitud fue exitosa.
- return response.json()["choices"][0]["message"]["content"] → Extrae y devuelve el mensaje generado por el chatbot.
- else: return "¡Ups! Parece que mi conexión falló…" → Mensaje de error si la conexión falla.
- if __name__ == "__main__"→ Este bloque ejecuta el chatbot en un bucle de conversación con el usuario.
- print("🐶 Chucho el Chatbot. Escribe 'salir' para terminar.") → Mensaje de bienvenida.
- Chatbot. Escribe 'salir' para terminar.") → Mensaje de bienvenida.
- while True → Inicia un bucle infinito para recibir y responder mensajes.
- user_input = input("Tú: ") → Solicita un mensaje del usuario.
- if user_input.lower() == "salir" → Si el usuario escribe "salir", se termina el programa.
- print("Chucho: ¡Nos vemos! ...") → Mensaje de despedida.
- break → Rompe el bucle.
- print("Chucho:", obtener_respuesta(user_input)) → Llama a la función obtener_respuesta() y muestra la respuesta del chatbot.

- **Resultado:**



  - chmod +x chucho.py → Da permisos de ejecución al script.
  - python3 chucho.py → Ejecuta el script con Python 3.
  - Por último, se observa la interacción con el chatbot y sus respuestas, las cuales coinciden con la personalidad especificada previamente.