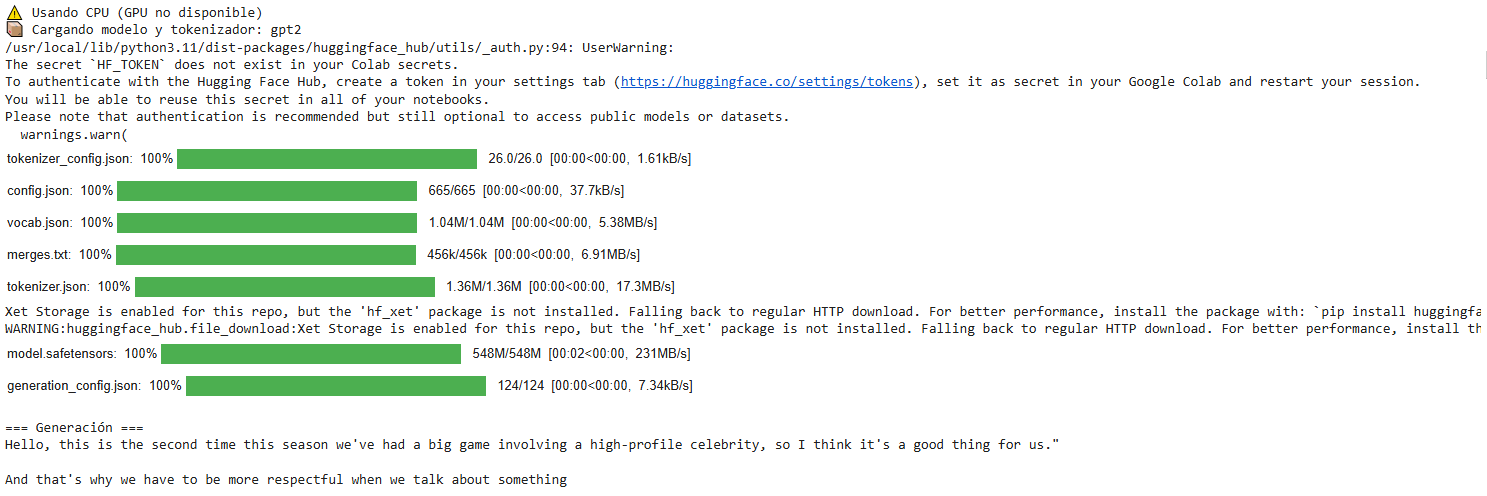
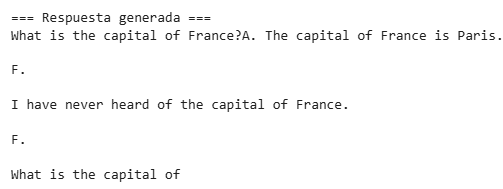
Realizado por Jhan Carlos Mosquera Valencia  
Informe

Primero tener en cuenta que este parcial se realizó en el entorno de desarrollo de Google colab en el cual se encontraran los codigos bien ex0plicados y sus respectivos resultados.

Ejercicio 1

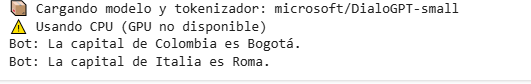
Se llevó a cabo lo solicitado por el profesor ya que el código presenta un establece un entorno de desarrollo eficiente y flexible para trabajar con modelos de lenguaje de gran tamaño (LLM), como GPT-2, utilizando las bibliotecas **Transformers** y **PyTorch**. A través de un conjunto de funciones bien definidas, se configura el entorno para cargar el modelo pre entrenado, verificar la disponibilidad de la GPU o CPU aunque hay que tener en cuenta que al realizaren en Google Colab este te proporciona ejecutar en cualquiera de las dos opciones y me agrada que se muestre en cuan se ejecuta, y generar respuestas textuales basadas en un prompt de entrada.  
Resultados:  


Ejercicio 2

Bueno esta implementación básica pero funcional de un sistema que procesa entradas de texto, genera respuestas coherentes utilizando un modelo pre entrenado y ajusta los parámetros de generación para obtener salidas más diversas.  
Resultados:  


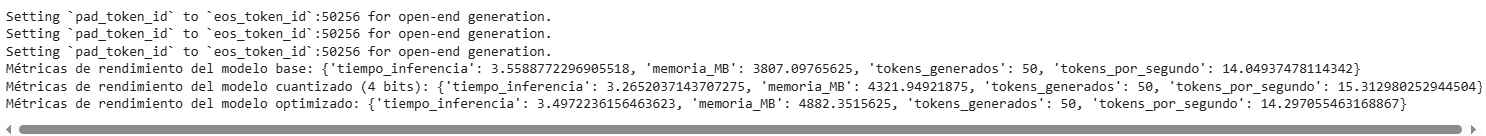
Ejercicio 3

En este caso implemento una solución efectiva para gestionar el contexto en conversaciones prolongadas haciendo que para extraer información relevante (como el nombre de un país) es una estrategia útil que permite al chatbot responder de manera flexible a una variedad de preguntas sobre países y capitales.

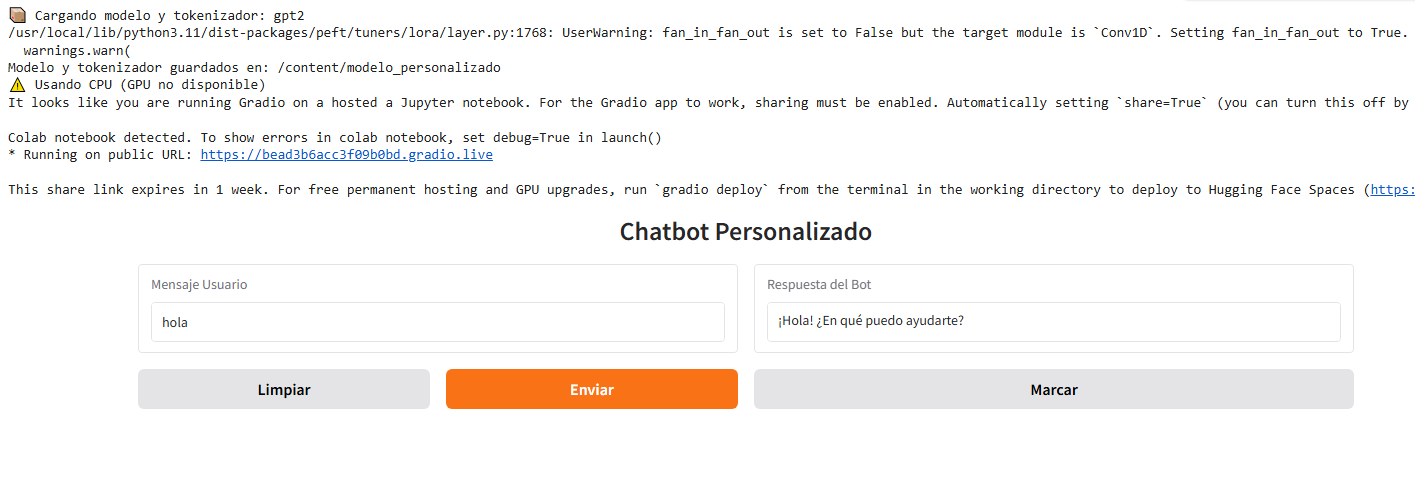
Resultado:  


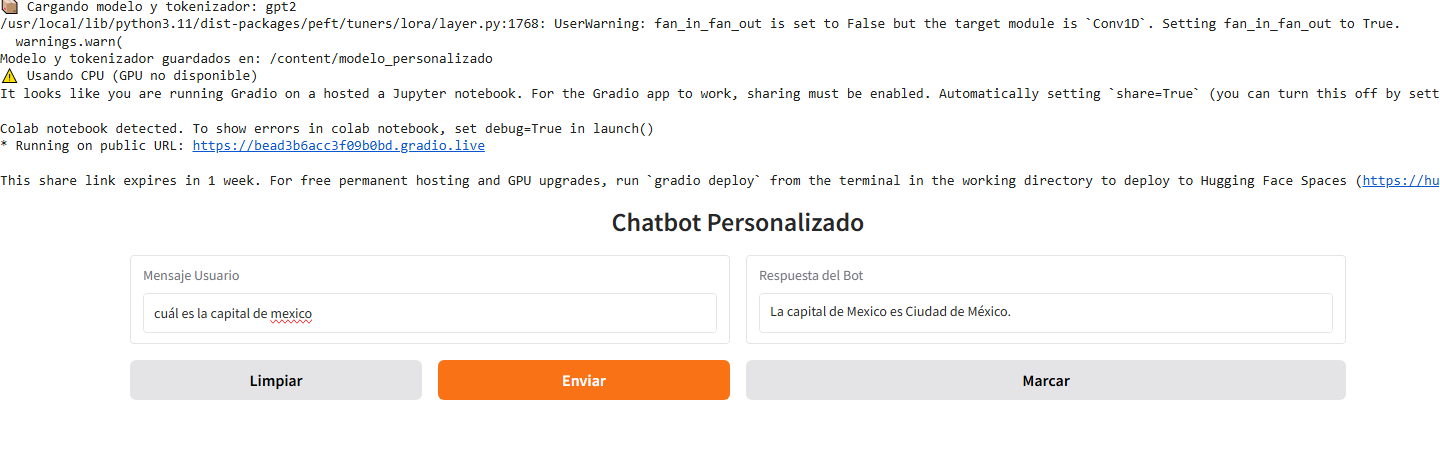
Ejercicio 4

En este caso se tuvo que instalar unas librerías y organizar el entorno para llevar a cabo con éxito el ejercicio la cual nos muestra la memoria utilizada, los tokens generados y tokens por segundo.

Resultado:

Ejercicio 5

En este punto se llevó a cabo con éxito gracias a que Google Colab tiene el uso de **Gradio** como interfaz web la cual ofrece un enfoque práctico y escalable para construir chatbots que puedan ser desplegados rápidamente y personalizados para tareas específicas.  
Resultado:  




Preguntas Teóricas

1. ¿Cuáles son las diferencias fundamentales entre los modelos encoder-only, decoder-only y encoder-decoder en el contexto de los chatbots conversacionales? Explique qué tipo de modelo sería más adecuado para cada caso de uso y por qué.

R// para **Encoder-only** los modelos se enfocan solo en leer y entender el texto de entrada. Toman una secuencia de palabras (como una pregunta del usuario) y las convierten en una representación interna, pero no generan texto. En caso de **Decoder-only** los modelos generan texto a partir de un contexto dado, es decir, toman un mensaje de inicio (como una pregunta) y generan una respuesta y por ultimo **Encoder-decoder**: Combina ambos ya que primero el encoder procesa la entrada para después el decoder genera una respuesta.

1. Explique el concepto de "temperatura" en la generación de texto con LLMs. ¿Cómo afecta al comportamiento del chatbot y qué consideraciones debemos tener al ajustar este parámetro para diferentes aplicaciones?

R// la temperatura es la respuesta que genera el modelo ya que si muestra una temperatura baja (cerca de 0) hace que el modelo sea más predecible y coherente, mientras que una temperatura alta (cerca de 1) hace que el modelo sea más creativo y variado en sus respuestas.

1. Describa las técnicas principales para reducir el problema de "alucinaciones" en chatbots basados en LLMs. ¿Qué estrategias podemos implementar a nivel de inferencia y a nivel de prompt engineering para mejorar la precisión factual de las respuestas?

R// algunas técnicas que se podrían implementar para el problema de las “alucinaciones” es fijar límites claros es decir limitar la longitud de las respuestas y especificar el contexto puede reducir la tendencia a generar información falsa, otra técnica sería una verificación adicional que use herramientas o base de datos externas para confirmar la información generada por el modelo.