

La API de GPT

Objetivo

El principal objetivo de este taller es el de afianzar los conocimientos adquiridos sobre API y arriesgarnos a usar una de las API (Generative Pre-trained Transformer) de OpenAI más modernas y complejas que existen hoy como lo es la API de GPT y a descubrir cómo podría ser aprovechada en soluciones IoT.

Los estudiantes entenderán cómo utilizar esta API en diversos contextos, desde generación de texto hasta comprensión del lenguaje natural, explorando su valor y potencial en aplicaciones de inteligencia artificial y procesamiento del lenguaje natural.

Conocimientos previos

¿Qué es la API de GPT de OpenAI? La API GPT (Generative Pre-trained Transformer) de OpenAI está diseñada para permitir a los desarrolladores integrar capacidades avanzadas de procesamiento del lenguaje natural (NLP) en cualquiera de sus aplicaciones y proyectos. GPT es un modelo de lenguaje de inteligencia artificial que ha sido pre-entrenado con vastas cantidades de texto y puede generar texto coherente y relevante en respuesta a una variedad de entradas y consultas.

Características Clave de la API de GPT:

1. **Generación de Texto:** La API de GPT permite generar texto auténtico y coherente en respuesta a preguntas, consultas o estímulos de entrada.
2. **Comprensión del Lenguaje Natural:** Utiliza técnicas de procesamiento del lenguaje natural para comprender y responder de manera contextualmente relevante.
3. **Flexibilidad en el Uso:** Se puede integrar en una variedad de aplicaciones y plataformas, desde chatbots hasta sistemas de recomendación y más.
4. **Modelos Pre-entrenados:** Basada en modelos de lenguaje pre-entrenados, lo que significa que ya ha aprendido patrones lingüísticos y contextuales a partir de grandes conjuntos de datos.
5. **Documentación Detallada:** La API de GPT está acompañada de una documentación exhaustiva que describe cómo interactuar con ella, qué parámetros son necesarios y qué resultados se pueden esperar.
6. **Seguridad y Privacidad:** OpenAI ha implementado medidas de seguridad y privacidad para proteger los datos y la información procesada a través de la API.

Ejemplos de Uso: La API de GPT se puede aplicar en una amplia gama de escenarios, como la generación de contenido para redes sociales, la redacción de textos creativos, la asistencia en la escritura de correos electrónicos o informes, la traducción automática de idiomas, entre otros.

Este entendimiento previo de la API de GPT sienta las bases para explorar y aprovechar sus capacidades en diversas aplicaciones y proyectos que requieran procesamiento avanzado del lenguaje natural.

Pasos a seguir para poder usar la API de GPT

La clave está en llegar a obtener una API Key. Una vez la tengas, simplemente la usarán en lo que la imaginación te dicte. De manera, que lo que aquí se describe es justamente los pasos para obtener esa API Key:

Paso 1. Debes suscribirte a la plataforma de desarrolladores de OpenAI

Esto es muy diferente a la suscripción que tú puedes tener de Chat GPT. Incluso, tú puedes ser un usuario que paga la suscripción a Chat GPT 4.0, pero para la API de GPT tu no existes, es como otra cosa, es "la plataforma de desarrolladores de OpenAI". Algo similar pasa con Whatsapp, una cosa es que tu seas usuario de whatsapp, otra cosa es que te conviertas en desarrollador. Esta es lo que debes hacer para suscribirte:

Método 1: <https://openai.com/> > Products > API Login > API

Nota: en el día a día: Es mejor que vayas directamente a la plataforma de desarrolladores: <https://platform.openai.com>

Paso 2. Pagar

Ya lo sé. Esto no te ha gustado, pero es necesario pagar para poder usar la API de GPT. Lo bueno es que los costos son realmente irrisorios, al menos para lo que quiere hacer una persona cuando desarrolla y para cumplir con esta guía quizá sea suficiente con fracciones de un dólar. Además el pago es similar a cuando tu le recargas dinero al celular pre-pago, de manera que no implica un compromiso de pagos mensuales. El proceso es este:

<https://platform.openai.com> >  > Menu lateral > Billing

Nota: No tenemos muy claras las nuestra estrategias de OpenAI para los que recién ingresan a hacer pruebas, pero al parecer son las siguientes:

- Aunque no tengas cargada la cuenta con dinero, puedes obtener una API Key
Es muy muy posible que puedas obtener una API Key
- Es posible que también puedas consumir servicios con esa API Key durante un tiempo, antes de que debas pagar.

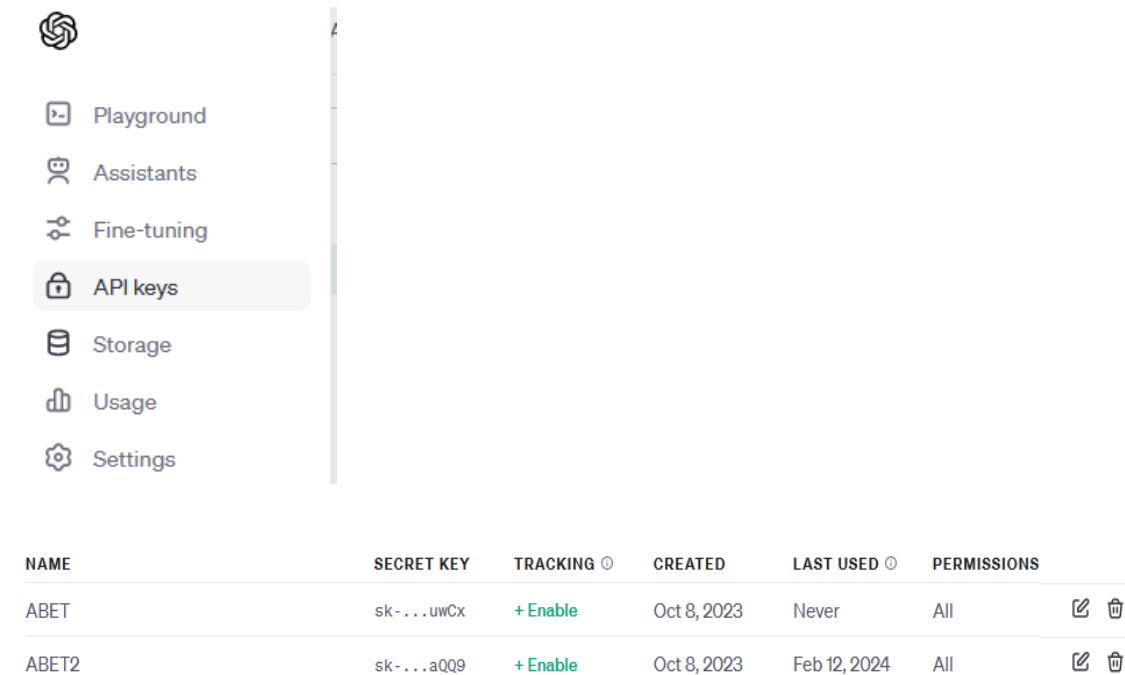
Paso 3. Obtener una API Key

<https://platform.openai.com> > Menu horizontal superior > Dashboard > API keys

Tarea 1. Registro en OpenAI y Obtención de Credenciales

Paso 1. Visita el sitio de la plataforma de desarrolladores de [OpenAI](https://openai.com/) y regístrate. El proceso detallado es: <https://openai.com/> > Products > API Login > API

Paso 2. Una vez registrado entra a la sección de API keys (el proceso detallado es: <https://platform.openai.com/> > Menu horizontal superior > Dashboard > API keys). Una vez ahí se genera la API_KEY. Asegura la llave debido a que una vez salgas de esa pestaña no podrás volver a obtenerla y tendrás que crear otra.



NAME	SECRET KEY	TRACKING ⓘ	CREATED	LAST USED ⓘ	PERMISSIONS
ABET	sk-...uWCx	+ Enable	Oct 8, 2023	Never	All
ABET2	sk-...aQQ9	+ Enable	Oct 8, 2023	Feb 12, 2024	All

Errores comunes

- Tu puedes ver una tabla de las API keys que ha creado, por ejemplo esta

NAME	SECRET KEY	TRACKING ⓘ	CREATED	LAST USED ⓘ	PERMISSIONS
MI API Key de prueba1	sk-...ogZr	Enabled	9 feb 2023	16 ene 2024	All
Mi API Key de prueba 2	sk-...eqgu	Enabled	27 feb 2024	Never	All

pero eso no significa que puedes copiar de allí tu “SECRET KEY”. Si no la anotaste en el momento de crearla, habrás perdido tu API Key. Quizá se pueda recuperar revisando el código donde la hayas usado. Para este taller, lo mejor es que crees una nueva API Key y te apresures a guardarla en algún lugar propio pero no público. Puede ser en google drive, pero en carpeta privada.

- Si tu guardas tu API Key en servidores públicos como github es algo que OpenAI detecta, de manera que procede a bloquear la API Key. Vea por ejemplo el siguiente mensaje corresponde a una API Key bloqueada por esa razón



Hi there,

We have determined that your OpenAI API key "**Chatbot**" (**sk-Z4F...Nbhh**) was leaked, and have disabled it with immediate effect.

This may be because you committed your API key to an online service such as GitHub, or your key may have been compromised in another way.

Head over to the [API Keys](#) page to create a new API key.

If your API key was stored in any locations—for instance, in code you are running—it will need to be updated before you can run this code again.

Finally, we ask that you please review our help center guides on [Best practices for API key safety](#) and [Preventing unauthorized usage](#).

Best,
The OpenAI team

- Un consejo, muy personal del profesor **Homero Ortega Boada** es que guardes tus API Key en un google sheet privado y bien protegido, con detalles como en este ejemplo:

	A	B	C	D	E
1	Fuente	Enlace de la fuente	Nombre de la API Key	API Key	Apuntes
2	OpenAI	https://platform.openai.com/api-key	Mi API Key de prueba 2	sjk-klkajkjfdkjakfklafnz mvnjksfkl	creada para hacer mi taller 3.2
3					
4					
5					
6					

Paso 3. Ahora probaremos si la llave de uso funciona.

- Se recomienda el uso de [Google Colaboratory](#) simplemente porque no todos los computadores de los estudiantes tiene instalado python, mientras que Google Colaboratory ofrece una versión en línea gratuita para programar en Python. En principio no importa el lenguaje, pero las ayudas que brinda la API de GPT son usualmente para Python. Si desea realizarlo de manera local, necesitará instalar python y la librería pip (en la mayoría de los casos pip estará instalada por defecto).

Nota sobre el uso de "pip": este comando no es parte del lenguaje python, sino una herramienta para instalar librerías a python, de manera que es un comando del sistema operativo subyacente. Los que trabajan en Ubuntu sabrán que pip es una herramienta de Ubuntu, no de python. En Google Colaboratory (Colab) se pueden enviar comandos del sistema operativo antecediendo el comando con "!", por ejemplo, se podría escribir:

```
!pip install algo
```

- Ahora instala biblioteca OpenAI en tu entorno de Python, como se indica abajo.

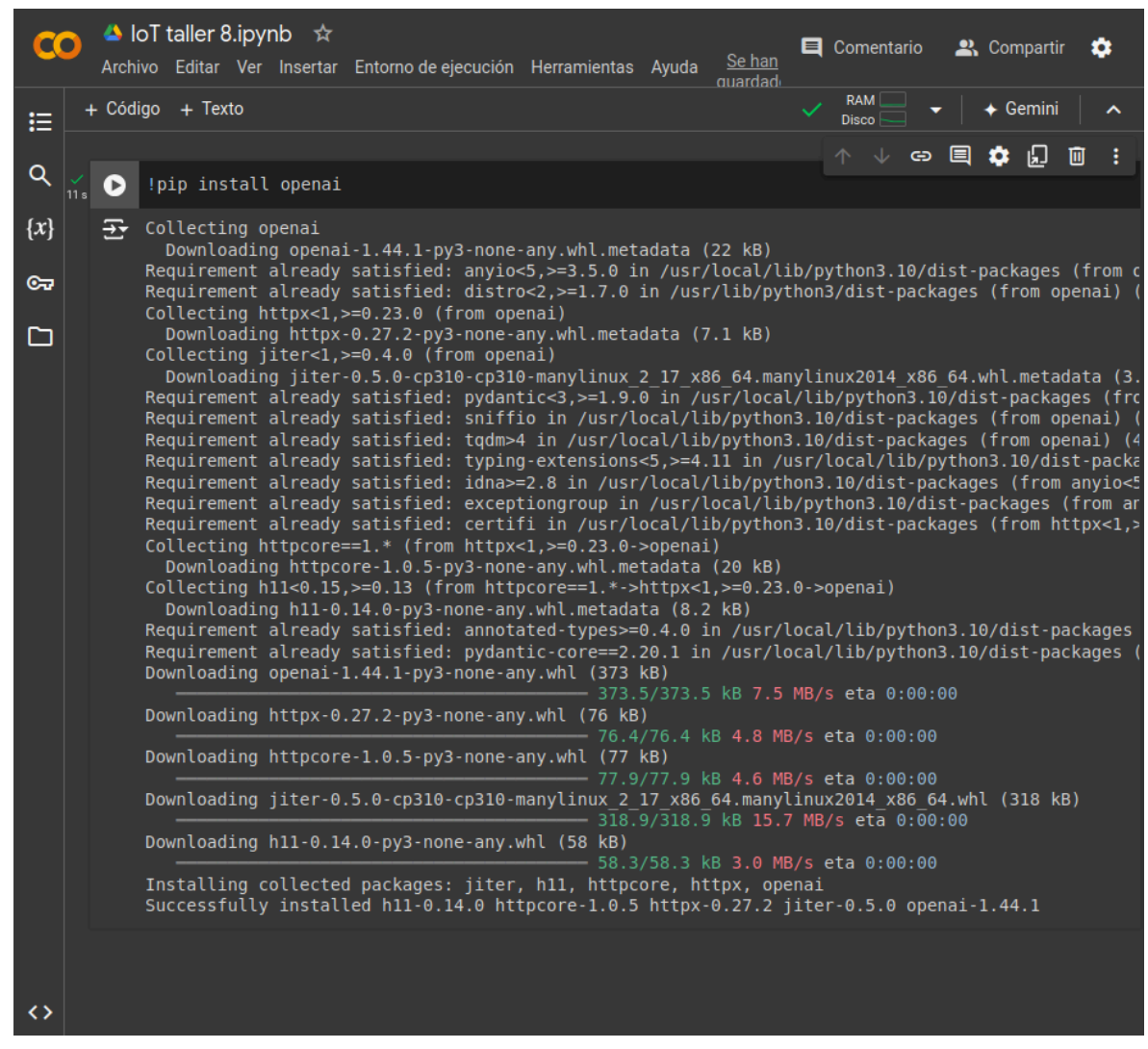
Escribe el siguiente código si usas Google Collaboratory

```
!pip install openai
```

Escribe el siguiente código si usas Python de tu computadora

```
pip install openai
```

Resultado obtenido en Colab al usar ejecutar el comando, adjuntar un pantallazo.

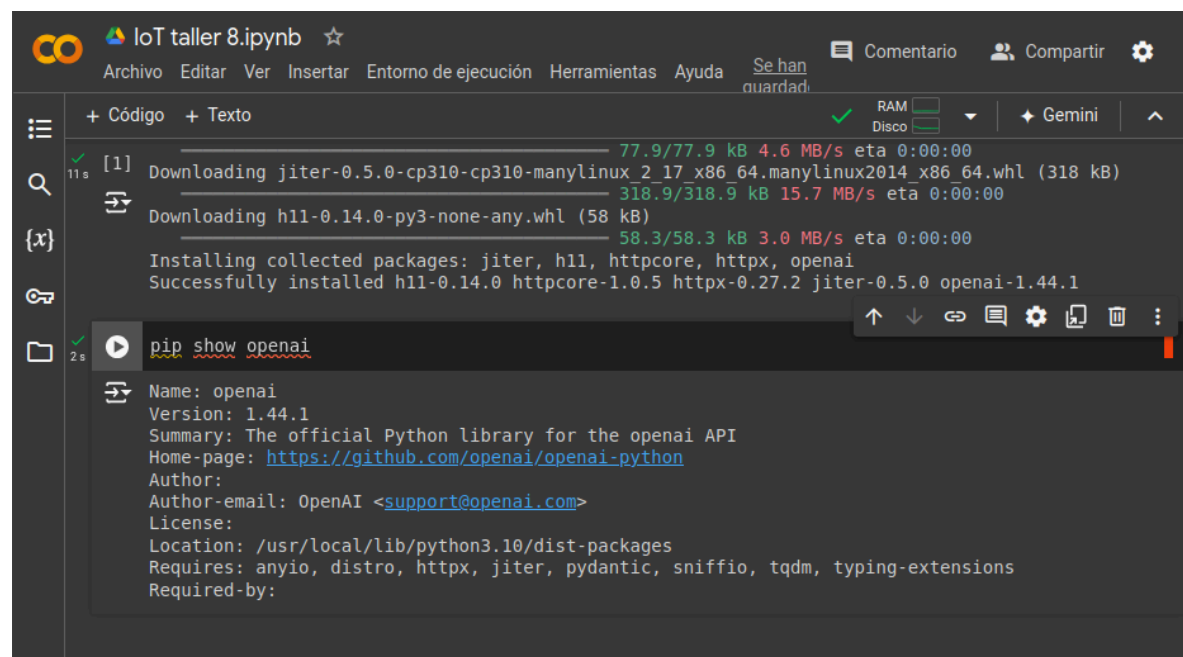
The screenshot shows the Google Colab interface. At the top, there's a header with the Colab logo, the file name 'IoT taller 8.ipynb', and various menu options like 'Archivo', 'Editar', 'Ver', 'Insertar', 'Entorno de ejecución', 'Herramientas', 'Ayuda', and 'Se han guardado'. Below the header, there's a toolbar with icons for code, text, and other functions. The main area displays the command '!pip install openai' and its output. The output shows the process of collecting and downloading various dependencies for the 'openai' package, including 'httpx', 'httpcore', 'jiter', and 'h11'. It also shows the progress of downloading the 'openai' package itself. The interface includes a left sidebar with navigation icons and a right sidebar with resource usage (RAM, Disco) and sharing options (Comentario, Compartir, Configuración).

Paso 4. Revisa la versión de OpenAI instalada

Continúa en el mismo libro de Google Colab, escribe el siguiente código y ejecútalo

```
pip show openai
```

Resultado: al correr el código, adjunto un pantallazo.



```
IoT taller 8.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se han guardado
+ Código + Texto
[1] Downloading jiter-0.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (318 kB)
318.9/318.9 kB 15.7 MB/s eta 0:00:00
Downloading h11-0.14.0-py3-none-any.whl (58 kB)
58.3/58.3 kB 3.0 MB/s eta 0:00:00
Installing collected packages: jiter, h11, httpcore, httpx, openai
Successfully installed h11-0.14.0 httpcore-1.0.5 httpx-0.27.2 jiter-0.5.0 openai-1.44.1

pip show openai
Name: openai
Version: 1.44.1
Summary: The official Python library for the openai API
Home-page: https://github.com/openai/openai-python
Author:
Author-email: OpenAI <support@openai.com>
License:
Location: /usr/local/lib/python3.10/dist-packages
Requires: anyio, distro, httpx, jiter, pydantic, sniffio, tqdm, typing-extensions
Required-by:
```

Paso 5. Obtener la respuesta de GPT

En este paso, continúa trabajando en el mismo libro de Google Colab

Hay que tener en cuenta que los creadores de la biblioteca OpenAI, se enfocaron en la programación orientada a objetos. Por lo tanto, al importar openAI, lo que se ha importado es la clase del objeto openAI. Por lo tanto, el siguiente paso es instanciar el objeto que vamos a usar.

Por eso, el primer paso es crear una instancia de la clase de interés. En este caso, la clase de interés es openAI, crearemos un objeto (instancia) llamado “client” o “miCliente”. Para ello, envía el siguiente código línea por línea para crear una instancia.

Error común: Puede no funcionar con tu propia API Key por cuestiones de registro con tarjeta de crédito. Pero vale la pena que pruebes porque al parecer OpenAI da un periodo de pruebas gratis. Para obviar esto, usal la siguiente API Key del profesor (Nota puede haber caducado, el profesor quizá pueda darte otra):

```
sk-proj-_aOZyg6zoEjbkgppRJ1SAPTokKK0ejuV7v4UiQSVwB0iSVezeBY5mP-nmKT3BIbk
FJ07oi4WagtGzzDo5Xc9piPt3ZIFx8YcuS1PUD1pCdAVUQ5FdjBqouHmXggA
```

```
import os
from openai import OpenAI
os.environ["OPENAI_API_KEY"] = "aqui debe ir tu API Key"
miCliente = OpenAI(
    # This is the default and can be omitted
    api_key=os.environ.get("OPENAI_API_KEY"),
)
```

Si usas Google Colab, recuerda, que mientras escribes código, al mismo tiempo estás escribiendo un libro, imagina que es tu libreta de apuntes y pon notas de texto que pueden serte útil a futuro, cuando necesites recordar lo que has hecho, por ejemplo:

✓ Creación de un cliente de OpenAI

Eso equivale a decir que estamos creando un objeto que es una instancia de la clase OpenAI. Como sabemos en la programación orientada a objetos, lo primero que se hace es crear un objeto para poder usar las funcionalidades de la clase

```
✓ [4] import os
0s

✓ [5] from openai import OpenAI
0s

[6] os.environ["OPENAI_API_KEY"] = "sk-aadadajkjfkasjfkjafjakljfklajkfjakl"

✓ [7] client = OpenAI(
0s      # This is the default and can be omitted
      api_key=os.environ.get("OPENAI_API_KEY"),
    )
```

Prepara el mensaje que deseas enviarle a GPT. Continúa trabajando en el mismo libro de Google Colab

```
misMensajes=[
    {
        "role": "user",
        "content": "cuentame un chiste",
    }
]
```

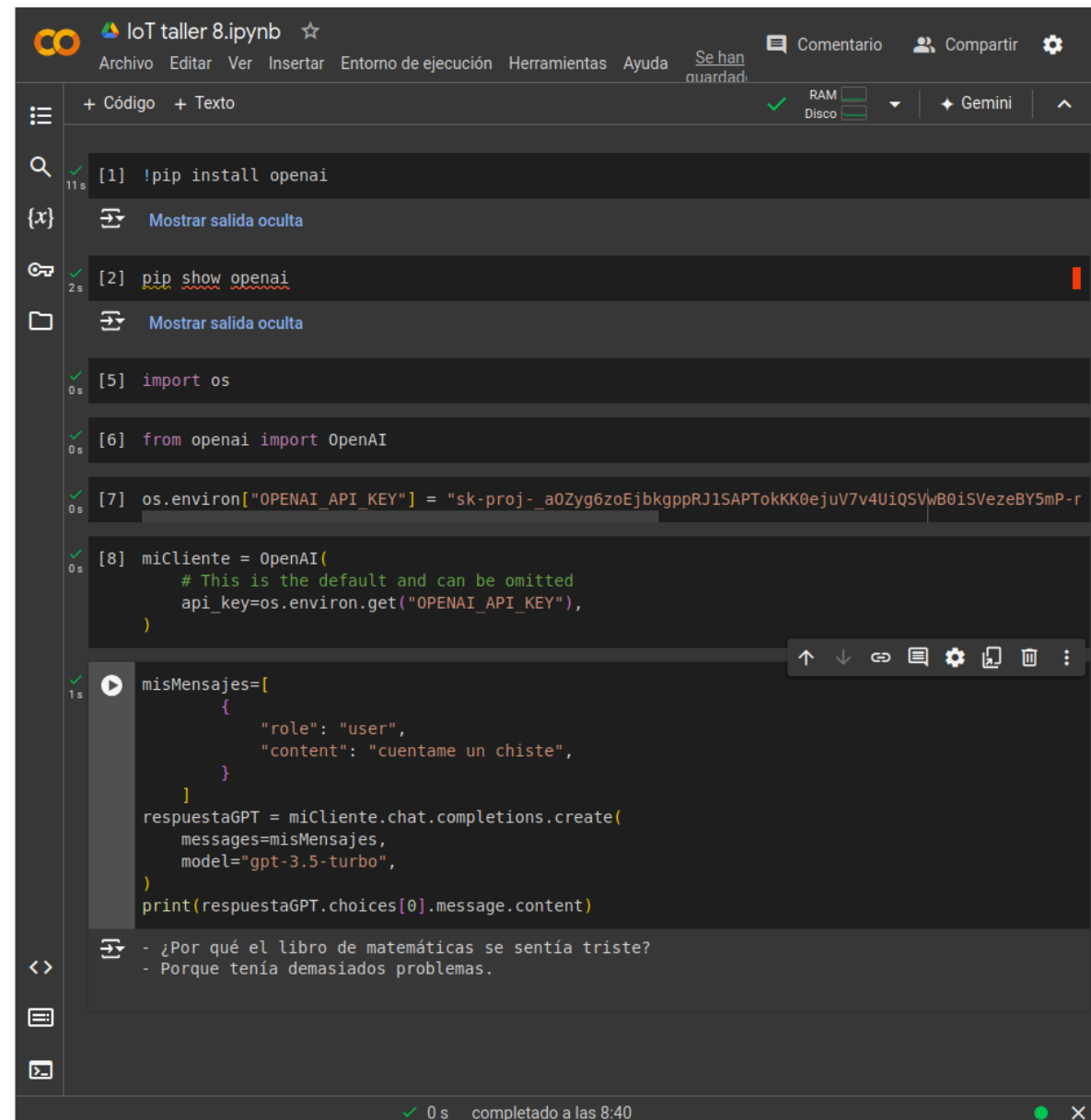
Código para enviar un mensaje a GPT y obtener respuesta. Continúa trabajando en el mismo libro de Google Colab

```
respuestaGPT = miCliente.chat.completions.create(
    messages=misMensajes,
    model="gpt-3.5-turbo",
)
```

Código para imprimir la respuesta. Continúa trabajando en el mismo libro de Google Colab

```
print(respuestaGPT.choices[0].message.content)
```

Resultado al correr el código.



The screenshot shows a Jupyter Notebook titled "IoT taller 8.ipynb". The interface includes a top bar with navigation options (Archivo, Editar, Ver, Insertar, Entorno de ejecución, Herramientas, Ayuda) and a status bar at the bottom indicating "0 s completado a las 8:40". The notebook contains several code cells:

- Cell [1]: `!pip install openai` (11 s)
- Cell [2]: `pip show openai` (2 s)
- Cell [5]: `import os` (0 s)
- Cell [6]: `from openai import OpenAI` (0 s)
- Cell [7]: `os.environ["OPENAI_API_KEY"] = "sk-proj-_a0Zyg6zoEjbkgppRJ1SAPTokKK0ejuV7v4UiQSVwB0iSVezeBY5mP-r"` (0 s)
- Cell [8]: `miCliente = OpenAI(api_key=os.environ.get("OPENAI_API_KEY"))` (0 s)
- Cell [9]: `misMensajes = [{"role": "user", "content": "cuentame un chiste"}]` (1 s)
- Cell [10]: `respuestaGPT = miCliente.chat.completions.create(messages=misMensajes, model="gpt-3.5-turbo")` (0 s)
- Cell [11]: `print(respuestaGPT.choices[0].message.content)` (0 s)

The output of the final cell shows the chatbot's response: `- ¿Por qué el libro de matemáticas se sentía triste?` and `- Porque tenía demasiados problemas.`

Tarea 2. Bot para extraer palabras clave de un bloque de texto.

Tu tarea es crear un bot capaz de extraer palabras clave de un párrafo de texto dado. Este bot ayudará a resumir y comprender mejor el contenido de un texto al identificar las

palabras más relevantes y significativas. Para conseguir esto utiliza las siguientes herramientas.

Además del modelo("model") existen otros parámetros a utilizar estos son temperature(cuyo rango es de 0 a 1), max_tokens, y top_p(rango de 0 a 1).

Se puede incluir otro rol con el cual dar instrucciones al bot este rol se conoce como system; si estos elementos se utilizarán en la tarea anterior se verían así:

```
from openai import OpenAI
client = OpenAI()

response = client.chat.completions.create(
    messages=[
        {
            "role": "system",
            "content": "Tu rol es el de certificar que la llave de uso funciona"
        },
        {
            "role": "user",
            "content": "Di Esto es una prueba"
        }
    ],
    model="gpt-3.5-turbo",
    temperature=0.5,
    max_tokens=64,
    top_p=1
)

#Se recomienda utilizar los parámetros con estos valores, pero también la experimentación.
```

Escribe aquí el nuevo código del bot

```
from openai import OpenAI
client = OpenAI()
texto_usuario = "Este es un párrafo de ejemplo que contiene palabras clave como inteligencia artificial, procesamiento de lenguaje natural y aprendizaje automático."
response = client.chat.completions.create(
    messages=[
        {
            "role": "system",
            "content": "Tu rol es el de extraer palabras clave del texto proporcionado por el usuario"
        },
        {
            "role": "user",
            "content": texto_usuario
        }
    ]
)
```

```

],
model="gpt-3.5-turbo",
temperature=0.5,
max_tokens=64,
top_p=1
)
#Se recomienda utilizar los parámetros con estos valores, pero
también la experimentación.
palabras_clave = response.choices[0].message.content.split(',')
print("Palabras clave:", palabras_clave)

```

Muestra el resultado

The screenshot shows a Jupyter Notebook cell with the following code:

```

from openai import OpenAI
client = OpenAI()
texto_usuario = "Este es un párrafo de ejemplo que contiene palabras clave como inteligencia artificial, procesamiento de lenguaje natural y aprendizaje automático."
response = client.chat.completions.create(
    messages=[
        {
            "role": "system",
            "content": "Tu rol es el de extraer palabras clave del texto proporcionado por el usuario"
        },
        {
            "role": "user",
            "content": texto_usuario
        }
    ],
    model="gpt-3.5-turbo",
    temperature=0.5,
    max_tokens=64,
    top_p=1
)
#Se recomienda utilizar los parámetros con estos valores, pero también la experimentación.
palabras_clave = response.choices[0].message.content.split(',')
print("Palabras clave:", palabras_clave)

```

The output of the cell is:

```

Palabras clave: ['Palabras clave extraídas: inteligencia artificial', ' procesamiento de lenguaje natural', ' aprendizaje automático.']

```

At the bottom of the notebook interface, it says "0 s completado a las 8:49".

Preguntas.

1. ¿Qué es temperature, max_tokens, top_p y cómo afectan al bot?
2. ¿Qué es un prompt y cómo se conecta con el rol de system?

¿Qué es 'temperature', 'max_tokens', 'top_p' y cómo afectan al bot?

- **Temperature:** Es un parámetro que controla la aleatoriedad de las respuestas generadas por el modelo. Su rango va de 0 a 1. Valores más bajos (cerca de 0) hacen que las respuestas sean más predecibles y coherentes, mientras que valores más altos (cerca de 1) introducen mayor aleatoriedad, lo que puede llevar a respuestas más creativas pero menos coherentes.
- **Max_tokens:** Es el límite máximo de palabras o tokens que el modelo puede generar en una respuesta. Si se establece un valor bajo, la respuesta será corta; si se permite un valor alto, el modelo podrá producir una respuesta más extensa.
- **Top_p:** Este parámetro controla la "nucleación" de las respuestas. Filtra las opciones menos probables cuando el modelo genera texto, manteniendo sólo las opciones con una probabilidad acumulada de p. Por ejemplo, si top_p es 0.9, el modelo elegirá entre las respuestas que, juntas, suman el 90% de probabilidad acumulada, lo que permite obtener respuestas más centradas en las opciones más relevantes sin eliminar completamente la aleatoriedad.

¿Qué es un prompt y cómo se conecta con el rol de system?

- **Prompt:** Un prompt es el mensaje o conjunto de instrucciones que se le proporciona al modelo para que genere una respuesta. Es la entrada que guía al modelo a responder de manera adecuada, y su diseño es crucial para obtener el resultado deseado. En esencia, es el contenido que el usuario (o desarrollador) envía para interactuar con el modelo.

- **Rol de system:** Dentro de una conversación en la API de GPT, el rol de system define el comportamiento o el rol que el bot debe asumir durante la interacción. El mensaje que se envía bajo este rol establece las directrices para el bot, lo que puede incluir instrucciones específicas, como "certificar que la llave de uso funciona" o "extraer palabras clave". El system actúa como un marco de referencia que moldea la manera en que el bot responde a los prompts del usuario.

Conclusiones: Cómo podrías utilizar la API de GPT para mejorar el trabajo hecho en clases anteriores.

Se podrían mejorar los trabajos realizados anteriormente haciendo uso de la API de GPT para generar texto que se puede basar en diferentes parámetros , mejorando la calidad y rapidez de las tareas que se piden generar.

También se podrían tener respuestas más personalizadas en las simulaciones, por ejemplo para el caso de las actividades donde se hace uso de Google Sheets digamos para cargar en una tabla datos simulados de diferentes sensores, esto con el fin de lograr una mejor interpretación de los datos.

Al integrarse en proyectos IoT, la API de GPT puede usarse para que responda preguntas o diferentes consultas relacionadas a la actividad en específico, digamos que se tiene un sistema x, se le podrían hacer preguntas o consultas sobre los sistemas que monitorizan el estado de ese sistema, además de poder dar predicciones sobre el estado futuro de ese sistema.