PECIER

Comité Nacional Peruano de la CIER

Curso Virtual

Python para el Análisis de Datos y la Automatización en el Sector Eléctrico

27, 29 y 31 de Octubre, 03, 05, 07, 10 y 12 de Noviembre 2025.

MARVIN COTO – FACILITADOR
E-mail: mcotoj@gmail.com

PECIER
Comité Nacional Peruano de la CIER

SESIÓN 7
Parte 1

PECIER
Comité Nacional Peruano de la CIER

Introducción a APIs con Python

Objetivos de la Sesión

1. Comprender qué es una API y por qué es útil
2. Conocer los distintos tipos de APIs (REST, SOAP, GraphQL)
3. Aprender a usar Python para interactuar con APIs públicas reales
4. Entender los conceptos básicos de autenticación y seguridad
5. Aplicar buenas prácticas al usar APIs

PECIER
Comité Nacional Peruano de la CIER

¿Qué es una API?

Una **API (Application Programming Interface)** es un intermediario que permite que dos aplicaciones se comuniquen entre sí. Especialmente para obtener información.
Con una API se pueden automatizar tareas de extracción de información de un servicio web.

PECIER
Comité Nacional Peruano de la CIER

¿Qué es una API?

Analogía:
Piensa en una API como el **mesero** de un restaurante. Tú (el cliente) haces un pedido (solicitud), y el mesero (la API) lleva esa orden a la cocina (el sistema) y luego trae la comida (respuesta).

PECIER
Comité Nacional Peruano de la CIER

Tipos de APIs

1. REST (la más común)

Usa HTTP y trabaja con recursos como URLs. Es simple, flexible y el tipo más común hoy en día.

```
import requests

# API de CoinGecko para obtener el precio de Bitcoin
url = "https://api.coingecko.com/api/v3/simple/price"
params = {"ids": "bitcoin", "vs_currencies": "usd"}

response = requests.get(url, params=params)
data = response.json()
print(f"Precio actual de Bitcoin: ${data['bitcoin']['usd']}")
```

Precio actual de Bitcoin: $115958

PECIER
Comité Nacional Peruano de la CIER

Tipos de APIs

En el código anterior:

- requests.get(...): Hacemos una solicitud GET (obtener datos) a la API.
- params=...: Pasamos los parámetros de consulta (como si fuera "?ids=bitcoin&vs_currencies=usd").
- response.json(): Convertimos la respuesta JSON en un diccionario de Python.

PECIER
Comité Nacional Peruano de la CIER

Tipos de APIs

SOAP (más usado en sistemas antiguos)
Trabaja con XML, más estructurado y rígido. Requiere librerías especiales como zeep.

GraphQL
Permite solicitar exactamente los datos que se necesitan. Más moderno, pero más complejo para empezar.

PECIER
Comité Nacional Peruano de la CIER

Ejemplo con API Real: OpenWeatherMap

Descripción:
Vamos a conectarnos con la API de clima de OpenWeatherMap para consultar el clima de una ciudad.
Requisitos:
Necesitas una cuenta gratuita en https://openweathermap.org/api y obtener una **API Key**.

Ejemplo con API Real: OpenWeatherMap

PECIER
Comité Nacional Peruano de la CIER

```
import requests

API_KEY = " "  # Reemplázala con tu clave real
ciudad = "Paris"
url = f"http://api.openweathermap.org/data/2.5/weather?
q={ciudad}&appid={API_KEY}&units=metric&lang=es"

response = requests.get(url)

if response.status_code == 200:
    datos = response.json()
    temp = datos['main']['temp']
    clima = datos['weather'][0]['description']
    print(f"El clima en {ciudad} es {clima} con {temp}°C")
else:
    print(f"Error {response.status_code}: {response.text}")
```

Precios de Criptomonedas con CoinGecko

PECIER
Comité Nacional Peruano de la CIER

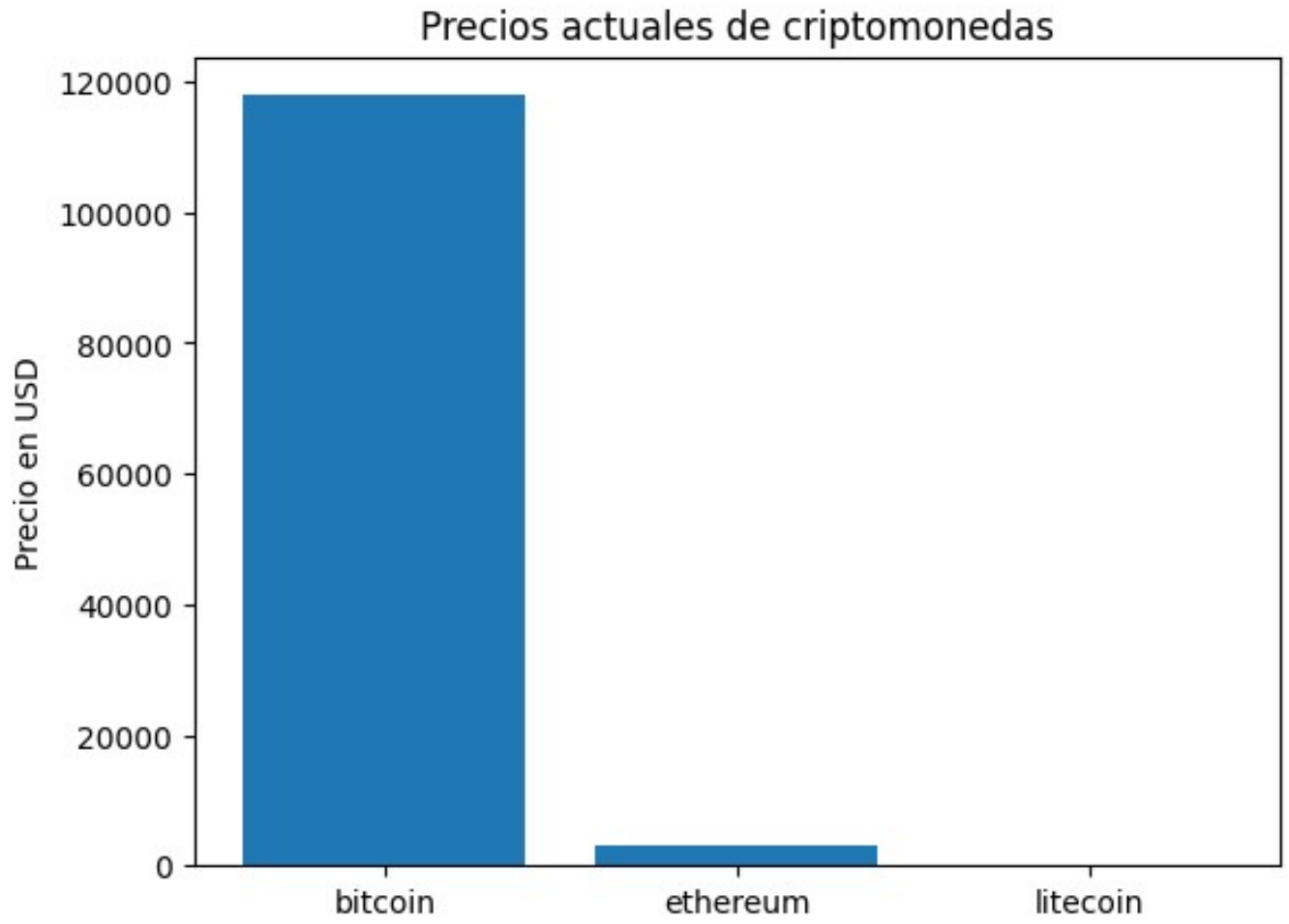
CoinGecko ofrece datos en tiempo real sin necesidad de autenticación.

```
import requests
import matplotlib.pyplot as plt

criptos = ["bitcoin", "ethereum", "litecoin"]
precios = []

for cripto in criptos:
    url = "https://api.coingecko.com/api/v3/simple/price"
    params = {"ids": cripto, "vs_currencies": "usd"}
    r = requests.get(url, params=params)
    precios.append(r.json()[cripto]['usd'])

# Gráfico de barras
plt.bar(criptos, precios)
plt.title("Precios actuales de criptomonedas")
plt.ylabel("Precio en USD")
plt.show()
```

Precios de Criptomonedas con CoinGecko

PECIER
Comité Nacional Peruano de la CIER

CoinGecko ofrece datos en tiempo real sin necesidad de autenticación.



PECIER
Comité Nacional Peruano de la CIER

APIs Relacionadas con Energía

1. **Red Eléctrica Española - REE** Documentación: https://www.ree.es/es/apidatos
2. **ENTSO-E Transparency Platform (Europa)** https://transparency.entsoe.eu/
3. **EIA U.S. Energy Information Administration** https://www.eia.gov/opendata/
4. **CO2 Signal** (intensidad de carbono por país) https://www.co2signal.com/

PECIER
Comité Nacional Peruano de la CIER

Ejemplos y ejercicios

PECIER_dia7_parte1

colab.research.google.com

PECIER

Comité Nacional Peruano de la CIER

SESIÓN 7
Parte 2

PECIER
Comité Nacional Peruano de la CIER

Uso de APIs para interactuar con Modelos de Lenguaje (LLMs) como GPT

En esta sección se estudiará la arquitectura básica de los LLMs y sus APIs, para pprender a configurar y autenticarse con APIs de modelos de lenguaje. Adicionalmente se especificará el envío de prompts y procesamiento de respuestas, para implementar casos de uso avanzados en el sector eléctrico.

PECIER
Comité Nacional Peruano de la CIER

Introducción a los LLMs y sus APIs

Los Modelos de Lenguaje (LLMs) son sistemas de IA entrenados para comprender y generar texto similar al humano. GPT (Generative Pre-trained Transformer) es uno de los modelos más avanzados, el cual se produjo después de una serie de modelos desarrollados para predecir la siguiente palabra en una secuencia. Con este principio se desarrollan sistemas de chat, de traducción, de clasificación de texto.

PECIER
Comité Nacional Peruano de la CIER

Introducción a los LLMs y sus APIs

¿Por qué usar APIs?

- Además del acceso a fuentes de datos, las API sin pueden utilizar para enviar información (solicitudes, prompts) a LLMs, como ChatGPT y obtener la respuesta. Esto permite integración con otras aplicaciones y el desarrollo de programas escalables y de mayor funcionalidad.

Dado que los modelos de lenguaje son desarrollados por empresas que cobran por utilizarlos (aunque hay gratuitos),

PECIER
Comité Nacional Peruano de la CIER

Introducción a los LLMs y sus APIs

```
# Nota: Ya están cargadas previamente el openai y el api-key
# Ejemplo básico de conexión a la API de OpenAI
#import openai

#openai.api_key = "tu-api-key"  # Nunca expongas esto en código público

response = client.chat.completions.create(
    model="gpt-4",
    messages=[
        {"role": "user", "content": "Explica cómo funciona un transformador
eléctrico"}
    ]
)

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

PECIER
Comité Nacional Peruano de la CIER

Introducción a los LLMs y sus APIs

Un transformador eléctrico es un dispositivo que transfiere energía eléctrica de un circuito a otro mediante campos magnéticos fluctuantes, que se produce a través de sus bobinas. Este proceso se realiza sin ningún tipo de conexión física y es a menudo acompañado por un cambio en voltaje y corriente.

Un transformador eléctrico consta básicamente de dos o más bobinas de cable (denominadas normalmente primario y secundario) enrolladas alrededor de un núcleo de material ferroso, usualmente hierro laminado.

Aquí está cómo funciona un transformador eléctrico paso a paso:

1. La corriente alterna en la bobina primaria genera un campo magnético en el núcleo del transformador.
2. Este campo magnético fluctuante se expande y colapsa de forma constante, lo que induce una corriente en la bobina secundaria.
3. En función de la razón entre el número de vueltas en la bobina primaria y en la secundaria, el voltaje en la bobina secundaria puede ser mayor (en un transformador elevador), menor (en un transformador reductor) o igual (en un transformador de aislamiento) al voltaje en la bobina primaria.

PECIER
Comité Nacional Peruano de la CIER

Introducción a los LLMs y sus APIs

Un par de cosas a tener en cuenta es que los transformadores funcionan sólo con corriente alterna y que siempre hay una cierta cantidad de pérdida de energía en el proceso de transferencia y transformación de la energía eléctrica, que se manifiesta en forma de calor.

PECIER
Comité Nacional Peruano de la CIER

Configuración y Autenticación

Obtención de API Key

- Crear cuenta en la plataforma del proveedor (OpenAI, Anthropic, etc.)
- Generar clave API en el panel de configuración
- Almacenar de forma segura (variables de entorno)

```
# Forma segura de manejar API keys
import os
from dotenv import load_dotenv

load_dotenv()  # Carga variables desde .env
api_key = os.getenv("OPENAI_API_KEY")
```

PECIER
Comité Nacional Peruano de la CIER

Configuración y Autenticación

Patrones de Autenticación

```
# OpenAI
openai.api_key = api_key

# Anthropic
import anthropic
client = anthropic.Client(api_key=api_key)

# Azure OpenAI
openai.api_type = "azure"
openai.api_base = "https://tu-endpoint.openai.azure.com/"
openai.api_version = "2023-05-15"
```

PECIER
Comité Nacional Peruano de la CIER

Interacción Básica: Prompts y Respuestas

Estructura de un Prompt Efectivo

1. **Contexto**: Establece el rol y conocimiento necesario
2. **Instrucción**: Qué debe hacer el modelo
3. **Ejemplos** (opcional): Demostraciones del formato esperado
4. **Pregunta/Consulta**: Solicitud específica

Interacción Básica: Prompts y Respuestas

PECIER
Comité Nacional Peruano de la CIER

```
prompt_tecnico = """
Eres un ingeniero eléctrico senior con 20 años de experiencia en
distribución de energía.
Explica en términos técnicos pero accesibles el concepto de flujo de
carga en redes eléctricas,
incluyendo:

1. Definición técnica
2. Importancia en la operación del sistema
3. Ejemplo numérico simple
4. Consideraciones prácticas

Usa ecuaciones cuando sea relevante y formato Markdown para la
respuesta.
"""

response = openai.ChatCompletion.create(
  model="gpt-4",
  messages=[{"role": "user", "content": prompt_tecnico}],
  temperature=0.7  # Controla la creatividad (0-2)
)
```

Interacción Básica: Prompts y Respuestas

PECIER
Comité Nacional Peruano de la CIER

1. Definición técnica

El flujo de carga, también conocido como análisis de flujo de potencia, es un cálculo matemático que se realiza en redes eléctricas para determinar la distribución de la energía eléctrica en sus diferentes nodos o puntos de interconexión. En términos técnicos, el análisis de flujo de carga se realiza para determinar el estado operativo de una red eléctrica bajo condiciones de carga estable.
Las ecuaciones básicas que rigen este análisis son las leyes de Kirchhoff para los circuitos eléctricos, que establecen que la suma de las corrientes en un nodo es cero (Ley de Kirchhoff para las corrientes) y que la suma de las tensiones en un bucle cerrado es cero (Ley de Kirchhoff para las tensiones).

2. Importancia en la operación del sistema

El análisis de flujo de carga es esencial en la operación y planificación de las redes eléctricas. Ayuda a los operadores del sistema a entender cómo se distribuye la electricidad en la red bajo diferentes condiciones de carga, permitiendo identificar posibles sobrecargas o deficiencias en la red. Este análisis también es crucial para la planificación de la expansión de la red, ya que permite determinar dónde y cuándo se necesitan nuevas infraestructuras de generación o transmisión.

Interacción Básica: Prompts y Respuestas

PECIER
Comité Nacional Peruano de la CIER

3. Ejemplo numérico simple
Consideremos una red eléctrica sencilla con tres nodos: un generador en el nodo 1, una carga en el nodo 2 y una interconexión al resto de la red en el nodo 3. Supongamos que el generador en el nodo 1 produce 10 MW, la carga en el nodo 2 consume 8 MW y el resto de la red consume 2 MW.
En este caso, el flujo de carga sería de 10 MW desde el nodo 1 al nodo 2 y al nodo 3. De estos 10 MW, 8 MW se consumen en el nodo 2 y los 2 MW restantes se envían al resto de la red a través del nodo 3.

4. Consideraciones prácticas
El análisis de flujo de carga es una herramienta poderosa, pero también tiene sus limitaciones. En primer lugar, este análisis asume condiciones de carga estables, lo que significa que no tiene en cuenta las fluctuaciones de carga que ocurren en la realidad. Además, el análisis de flujo de carga es un cálculo iterativo que puede ser computacionalmente intensivo, especialmente en redes eléctricas grandes y complejas.

Además, el análisis de flujo de carga a menudo requiere un modelo detallado de la red eléctrica, que puede ser difícil de obtener en la práctica. Finalmente, este análisis no puede capturar los eventos dinámicos que ocurren en la red, como los cortocircuitos o las fallas de los equipos, que pueden tener un impacto significativo en el flujo de carga.

Casos de Uso

Generación de Reportes Técnicos

```
# Datos de entrada
datos_subestacion = {
    "nombre": "Subestación Norte",
    "capacidad": "150 MVA",
    "nivel_voltaje": "230/115 kV",
    "incidentes": ["Sobretensión 15/11", "Mantenimiento programado 20/11"]
}
```

PECIER
Comité Nacional Peruano de la CIER

Casos de Uso

Generación de Reportes Técnicos

```
# Prompt
prompt_reporte = f"""
Genera un reporte ejecutivo de una página sobre el estado de la
{datos_subestacion['nombre']} con:
- Capacidad: {datos_subestacion['capacidad']}
- Nivel de voltaje: {datos_subestacion['nivel_voltaje']}

Incluye:
1. Resumen operativo
2. Análisis de incidentes recientes: {',
'.join(datos_subestacion['incidentes'])}
3. Recomendaciones de mantenimiento
4. Riesgos potenciales

Formato: Encabezado, secciones claras, puntos clave destacados.
"""
```

Casos de Uso

Generación de Reportes Técnicos

PECIER
Comité Nacional Peruano de la CIER

```
# Llamada a la API
response = client.chat.completions.create(
    model="gpt-4",
    messages=[{"role": "user", "content": prompt_reporte}],
    max_tokens=1000
)

# Mostrar resultado
print(response.choices[0].message.content)
```

Casos de Uso

PECIER
Comité Nacional Peruano de la CIER

Generación de Reportes Técnicos

Reporte Ejecutivo - Estado de la Subestación Norte

Resumen Operativo

La Subestación Norte con una capacidad de 150 MVA y un nivel de voltaje de 230/115 kV ha estado operando de manera efectiva y ha logrado mantener una operación libre de importantes incidencias. Cuenta con un sistema de disparos automáticos eficiente, logrando evitar situaciones críticas.

Análisis de Incidentes Recientes

Sobretensión 15/11: La sobretensión que se presentó en la subestación es una alerta importante puesto que puede causar daños al equipamiento. Los sistemas de protección funcionaron correctamente y automáticamente normalizaron el sistema evitando daños importantes.

Mantenimiento Programado 20/11 : Se completó con éxito el mantenimiento programado el 20 de noviembre asegurando la eficiencia de los equipos y las operaciones. Los resultados del mantenimiento indican que todos los componentes críticos están en buen estado.

Casos de Uso

PECIER
Comité Nacional Peruano de la CIER

Generación de Reportes Técnicos

Recomendaciones de Mantenimiento

Es propicio aumentar la frecuencia de las auditorías y pruebas desde trimestral a mensual, para evaluar el estado de los aisladores, transformadores y sistemas de protección. También es aconsejable mejorar la capacitación del personal sobre la detección de anomalías y la toma de acciones correctivas de manera oportuna.

Riesgos Potenciales

A pesar de que los sistemas están adecuadamente protegidos, la posibilidad de sobretensiones, como la que ocurrió recientemente es un riesgo potencial. Otros factores de riesgo incluyen desastres naturales como terremotos e inundaciones, y la posibilidad de fallos técnicos en las instalaciones.

Conclusion

Pese a los riesgos potenciales, la Subestación Norte ha demostrado ser resiliente gracias a sus sistemas de protección eficientes. Con una supervisión más cercana, auditorías más frecuentes y un mejor mantenimiento, se anticipa que la operación de la subestación será aún más efectiva y segura.

Casos de Uso

PECIER
Comité Nacional Peruano de la CIER

```
# Para convertirlo a Word:
!pip install python-docx

from docx import Document

# Texto generado por el modelo
texto = response.choices[0].message.content

# Crear documento
doc = Document()
doc.add_heading("Reporte Ejecutivo", level=1)

for linea in texto.split("\n"):
    doc.add_paragraph(linea)

# Guardar como archivo Word
doc.save("reporte_subestacion.docx")
```

Análisis de Datos con Chain-of-Thought

PECIER
Comité Nacional Peruano de la CIER

```
datos_consumo = """
Enero: 1250 MWh, 1.2% pérdidas
Febrero: 1380 MWh, 1.5% pérdidas
Marzo: 1420 MWh, 1.8% pérdidas
Abril: 1560 MWh, 2.1% pérdidas
"""
prompt_analisis = f"""
Analiza los siguientes datos de consumo y pérdidas:
{datos_consumo}
Realiza:
1. Cálculo del incremento porcentual mensual en consumo
2. Correlación entre consumo y pérdidas
3. Identificación de patrones anómalos
4. Recomendaciones técnicas

Muestra tu razonamiento paso a paso antes de dar conclusiones.
Usa formato Markdown con tablas para los cálculos.
“””
```

Análisis de Datos con Chain-of-Thought

PECIER
Comité Nacional Peruano de la CIER

```
response = client.chat.completions.create(
    model="gpt-4",
    messages=[{"role": "user", "content": prompt_analisis}],
    temperature=0.3
)

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

Análisis de Datos con Chain-of-Thought

PECIER
Comité Nacional Peruano de la CIER

1. Cálculo del incremento porcentual mensual en consumo

Para calcular el incremento porcentual mensual en consumo, se necesita restar el consumo del mes anterior al consumo del mes actual, dividir el resultado por el consumo del mes anterior y multiplicar por 100 para obtener el porcentaje.

Mes	Consumo (MWh)	Incremento porcentual
Enero	1250	
Febrero	1380	(1380-1250)/1250 * 100 = 10.4%
Marzo	1420	(1420-1380)/1380 * 100 = 2.9%
Abril	1560	(1560-1420)/1420 * 100 = 9.9%

2. Correlación entre consumo y pérdidas

Para calcular la correlación entre consumo y pérdidas, se puede usar el coeficiente de correlación de Pearson. Este coeficiente varía entre -1 y 1, donde 1 indica una correlación positiva fuerte, -1 una correlación negativa fuerte y 0 ninguna correlación.

Análisis de Datos con Chain-of-Thought

PECIER
Comité Nacional Peruano de la CIER

Mes	Consumo (MWh)	Pérdidas (%)
Enero.	1250	1.2
Febrero	1380	1.5
Marzo	1420	1.8
Abril	1560	2.1

Usando una calculadora de correlación en línea, obtenemos un coeficiente de correlación de 1. Esto indica una correlación positiva fuerte, lo que significa que a medida que el consumo aumenta, también lo hacen las pérdidas.

3. Identificación de patrones anómalos
Al observar los datos, no parece haber ningún patrón anómalo. El consumo y las pérdidas aumentan de manera constante cada mes.

Análisis de Datos con Chain-of-Thought

PECIER
Comité Nacional Peruano de la CIER

4. Recomendaciones técnicas

Dado que hay una correlación positiva fuerte entre el consumo y las pérdidas, se recomienda investigar las causas de estas pérdidas. Esto podría incluir la revisión de la infraestructura para identificar cualquier ineficiencia o problema que pueda estar contribuyendo a las pérdidas. Además, se podría considerar la implementación de medidas de ahorro de energía para reducir el consumo general.

Buenas Prácticas para APIs de LLMs

PECIER
Comité Nacional Peruano de la CIER

Manejo de Errores

```
try:
    response = openai.ChatCompletion.create(
        model="gpt-4",
        messages=[{"role": "user", "content":
prompt}],
        timeout=10  # Timeout en segundos
    )
except openai.error.APIError as e:
    print(f"Error de API: {e}")
except openai.error.Timeout as e:
    print(f"Timeout: {e}")
except openai.error.RateLimitError as e:
    print(f"Límite de tasa excedido: {e}")
```

Buenas Prácticas para APIs de LLMs

PECIER
Comité Nacional Peruano de la CIER

Segmentación de Respuestas Largas

```
def get_long_response(prompt, max_tokens=4000):
    response = openai.ChatCompletion.create(
        model="gpt-4",
        messages=[{"role": "user", "content": prompt}],
        max_tokens=max_tokens,
        stream=True  # Stream para respuestas largas
    )

    full_response = []
    for chunk in response:
        content = chunk.choices[0].delta.get("content", "")
        print(content, end="", flush=True)
        full_response.append(content)

    return "".join(full_response)
```

PECIER
Comité Nacional Peruano de la CIER

Ejemplos y ejercicios

PECIER_dia7_parte2

colab.research.google.com

PECIER
Comité Nacional Peruano de la CIER

SESIÓN 7
Parte 3

Aplicaciones de LLMs en el Sector Eléctrico

PECIER
Comité Nacional Peruano de la CIER

Veremos ejemplos de:

- Análisis de datos SCADA
- Generación de reportes técnicos
- Diagnóstico de subestaciones
- Exportación de resultados a Word o PDF
- Buenas prácticas de implementación

Requisitos técnicos

PECIER
Comité Nacional Peruano de la CIER

Instale las bibliotecas necesarias:

```
pip install openai python-dotenv pandas matplotlib tabulate python-docx
```

Uso seguro de la API Key (en notebooks Jupyter)

PECIER
Comité Nacional Peruano de la CIER

Paso 1. Define la clave en una celda previa del notebook (nunca publiques esta celda):

```
python import os os.environ["OPENAI_API_KEY"] = "sk-tu-clave-aqui"
```

Paso 2. Crear el cliente OpenAI:

```
from openai import OpenAI
client = OpenAI(api_key=os.getenv("OPENAI_API_KEY"))
```

Análisis de datos SCADA

PECIER
Comité Nacional Peruano de la CIER

```
import pandas as pd

class AsistenteSCADA:
    def __init__(self, client):
        self.client = client

    def analizar(self, datos: pd.DataFrame):
        prompt = f""" Eres un ingeniero eléctrico. Analiza estos datos SCADA:
{datos.to_markdown(index=False)}
Identifica:
1. Anomalías
2. Causas probables
3. Acciones correctivas
4. Nivel de prioridad
"""
        response = self.client.chat.completions.create(
            model="gpt-4",
            messages=[{"role": "user", "content": prompt}],
            temperature=0.2
        )
        return response.choices[0].message.content
```

Generación de reportes ejecutivos

PECIER
Comité Nacional Peruano de la CIER

```
datos_subestacion = {
    "nombre": "Subestación Norte",
    "capacidad": "150 MVA",
    "nivel_voltaje": "230/115 kV",
    "incidentes": ["Sobretensión 15/11", "Mantenimiento programado 20/11"]
}

prompt = f"""
Genera un informe ejecutivo técnico de una página para la
{datos_subestacion['nombre']}.

Capacidad: {datos_subestacion['capacidad']}
Nivel de voltaje: {datos_subestacion['nivel_voltaje']}
Incidentes recientes: {', '.join(datos_subestacion['incidentes'])}

Incluir:
1. Estado operativo general
2. Análisis de incidentes
3. Recomendaciones
4. Riesgos operativos

Formato profesional con secciones claras.
"""
```

Generación de reportes ejecutivos

PECIER
Comité Nacional Peruano de la CIER

```
response = client.chat.completions.create(
    model="gpt-4",
    messages=[{"role": "user", "content": prompt}]
)

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

Exportar resultados a Word

PECIER
Comité Nacional Peruano de la CIER

```
from docx import Document

def guardar_en_word(texto, nombre_archivo="reporte.docx"):
    doc = Document()
    for linea in texto.split("\n"):
        doc.add_paragraph(linea)
    doc.save(nombre_archivo)

guardar_en_word(response.choices[0].message.content)
```

Análisis de tendencias y anomalías

PECIER
Comité Nacional Peruano de la CIER

```
from sklearn.preprocessing import StandardScaler

def detectar_anomalias(datos: dict):
    df = pd.DataFrame(datos)
    df[['valor', 'limite']] = StandardScaler().fit_transform(df[['valor',
'limite']])
    df['anomalia'] = df['valor'] > df['limite'] * 1.2

    prompt = f"""
Estos son datos de monitoreo eléctrico:
{df.to_markdown(index=False)}

Identifica si existen patrones anómalos y qué acciones deberían tomarse.
"""
    respuesta = client.chat.completions.create(
        model="gpt-4",
        messages=[{"role": "user", "content": prompt}]
    )
    return df, respuesta.choices[0].message.content
```

Sistema de asistencia técnica

PECIER
Comité Nacional Peruano de la CIER

```
class AsistenteTecnico:
    def __init__(self, base_contextual):
        self.base = base_contextual

    def responder(self, consulta):
        contexto = self.base.get(consulta, "Sin información adicional.")
        prompt = f"""
Consulta técnica: {consulta}
Información contextual: {contexto}

Responde con base técnica y recomendaciones prácticas.
"""
        respuesta = client.chat.completions.create(
            model="gpt-4",
            messages=[{"role": "user", "content": prompt}]
        )
        return respuesta.choices[0].message.content
```

PECIER
Comité Nacional Peruano de la CIER

Buenas prácticas

- Usa os.environ o .env para proteger tu API Key.
- Establece temperature=0.2 en entornos críticos (menor creatividad, más precisión).
- Agrega límites con max_tokens=1000 para evitar respuestas excesivamente largas.
- Registra interacciones relevantes para trazabilidad y auditoría técnica.

PECIER
Comité Nacional Peruano de la CIER

Ejemplos y ejercicios

PECIER_dia7_parte3

colab.research.google.com

PECIER

Comité Nacional Peruano de la CIER

E-mail: pecier@cier.org