

```

In [1]: # Importar Las bibliotecas necesarias
import requests
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from datetime import datetime

# URL de La API de CoinGecko para obtener datos de criptomonedas
url = 'https://api.coingecko.com/api/v3/coins/bitcoin/market_chart?vs_currency=usd&days=30'

# Realizar una solicitud GET a La API para obtener datos de precio de Bitcoin
response = requests.get(url)

# Verificar si La solicitud fue exitosa (código de estado 200)
if response.status_code == 200:

    # Convertir La respuesta a formato JSON
    data = response.json()

    # Extraer Las fechas y Los precios del JSON
    timestamps = [entry[0] for entry in data['prices']]
    prices = [entry[1] for entry in data['prices']]

    # Corregir La conversión de marcas de tiempo UNIX a fechas legibles
    dates = [datetime.utcfromtimestamp(ts / 1000) for ts in timestamps]

    # Crear un gráfico de líneas para mostrar La variación del precio de Bitcoin en Los
    plt.figure(figsize=(10, 6))

    plt.plot(dates, prices, marker='o', color='g', label='Precio de Bitcoin (USD)')

    plt.xlabel('Fecha')
    plt.ylabel('Precio (USD)')

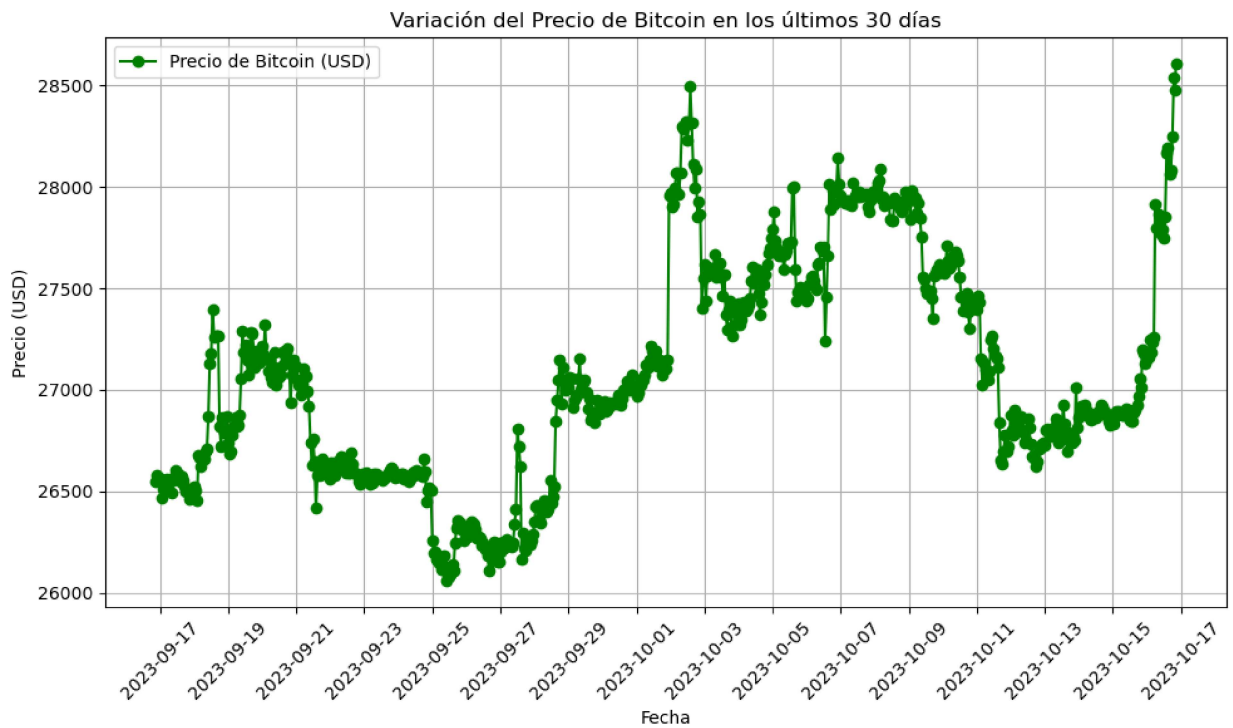
    plt.title('Variación del Precio de Bitcoin en los últimos 30 días')

    plt.legend()
    plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d')) # Formato de
    plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=2)) # Espaciamiento de
    plt.xticks(rotation=45)
    plt.grid(True)

    plt.tight_layout()
    plt.show()

else:
    print("Error al obtener datos de la API. Código de estado: ", response.status_code)

```



```
In [3]: import requests
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from datetime import datetime

# Lista de criptomonedas que deseas analizar
criptomonedas = ['bitcoin', 'ethereum', 'ripple']

# Función para obtener datos de la API y crear visualización
def obtener_datos_y_visualizar(criptomoneda):
    url = f'https://api.coingecko.com/api/v3/coins/{criptomoneda}/market_chart?vs_currency=us&days=30'

    response = requests.get(url)

    if response.status_code == 200:
        data = response.json()

        timestamps = [entry[0] for entry in data['prices']]
        prices = [entry[1] for entry in data['prices']]

        # Convertir las marcas de tiempo UNIX a fechas legibles
        dates = [datetime.utcfromtimestamp(ts / 1000) for ts in timestamps]

        plt.figure(figsize=(10, 6))

        plt.plot(dates, prices, marker='o', label=f'Precio de {criptomoneda.capitalize()}')

        plt.xlabel('Fecha')
        plt.ylabel('Precio (USD)')

        plt.title(f'Variación del Precio de {criptomoneda.capitalize()} en los últimos 30 días')
        plt.legend()

        plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))
        plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=2))
```

```

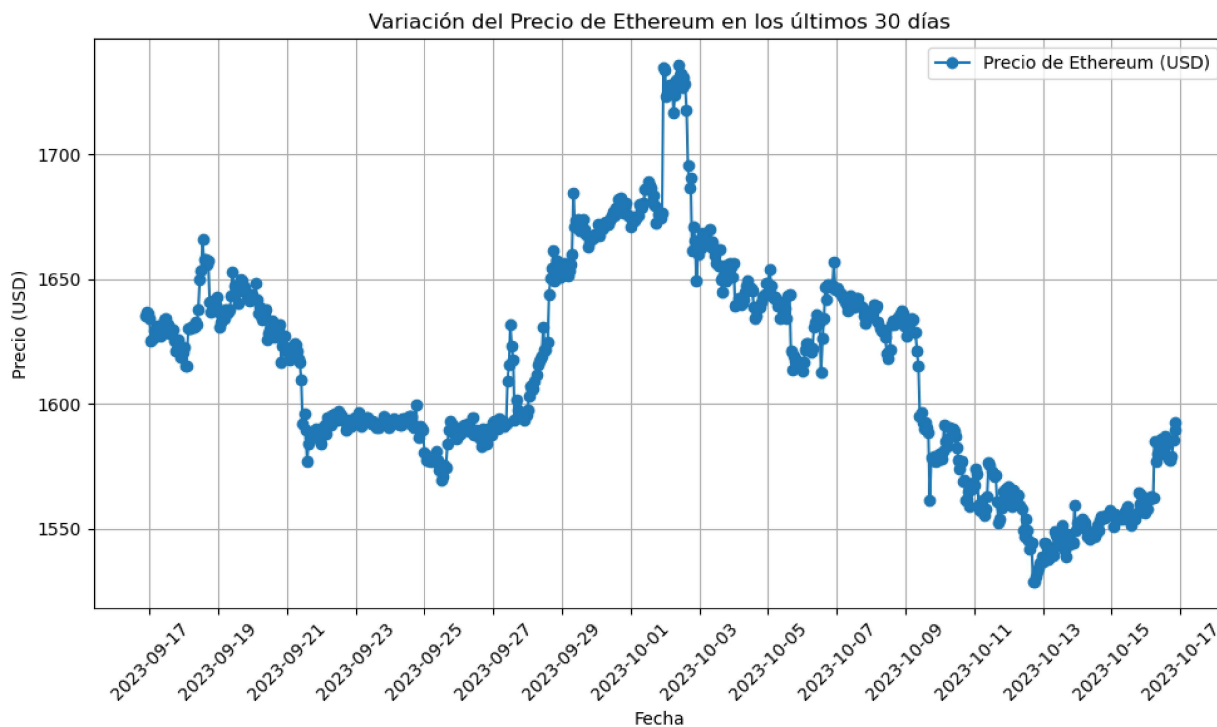
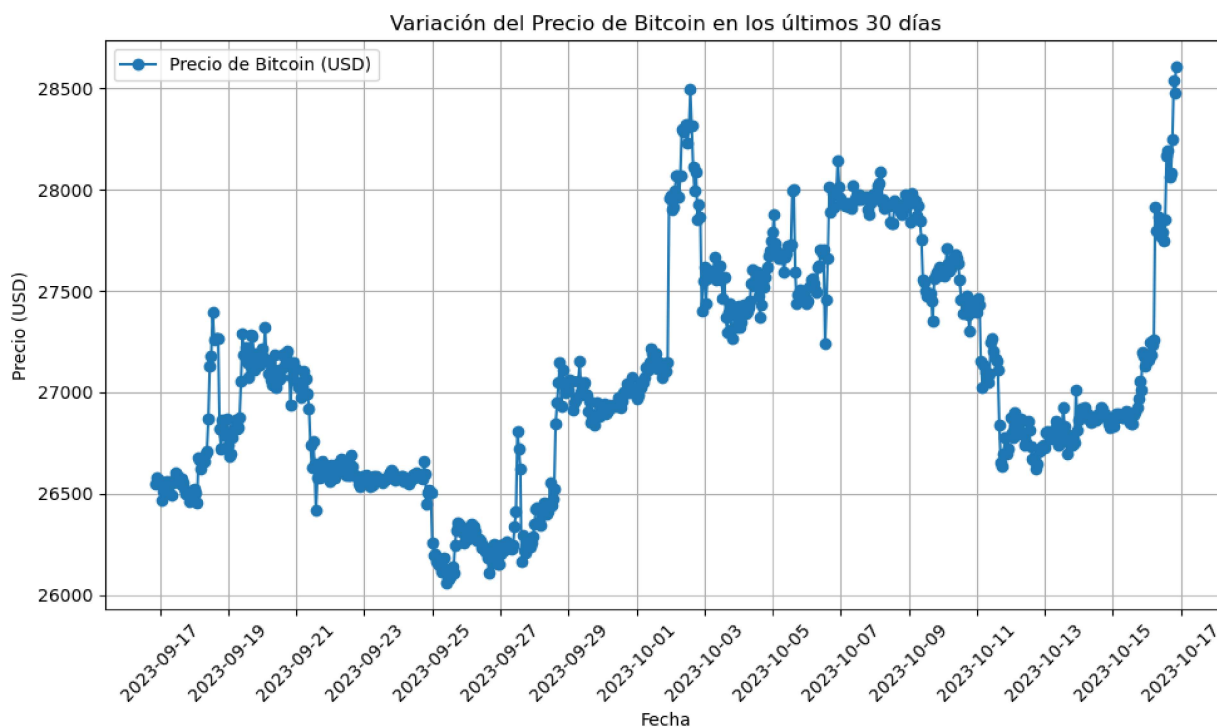
plt.xticks(rotation=45)
plt.grid(True)

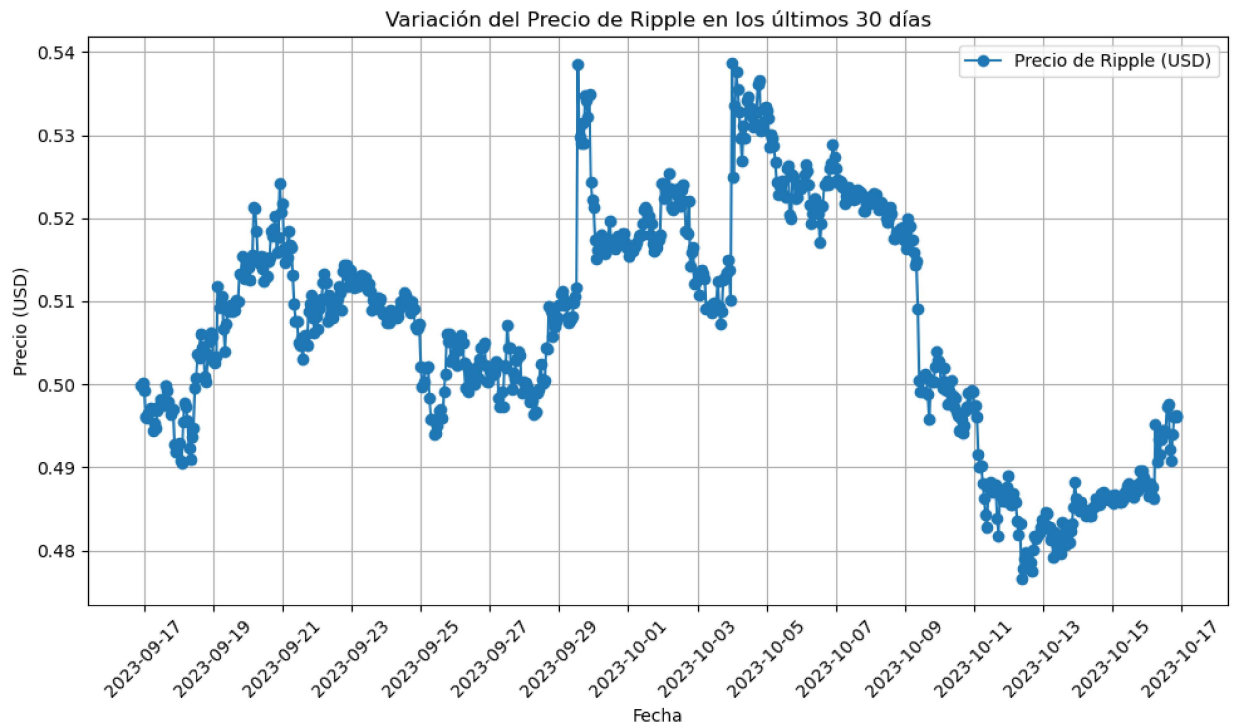
plt.tight_layout()
plt.show()

else:
    print(f"Error al obtener datos de la API para {criptomonedas}. Código de estado: ")

# Llamar a la función para cada criptomoneda en la lista
for criptomoneda in criptomonedas:
    obtener_datos_y_visualizar(criptomoneda)

```





```
In [4]: import requests
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from datetime import datetime

# Lista de criptomonedas que deseas analizar
criptomonedas = ['bitcoin', 'ethereum', 'ripple']

# Función para obtener datos de la API y crear visualización
def obtener_datos_y_visualizar(criptomoneda):
    url = f'https://api.coingecko.com/api/v3/coins/{criptomoneda}/market_chart?vs_currency=us&days=30'

    response = requests.get(url)

    if response.status_code == 200:
        data = response.json()

        timestamps = [entry[0] for entry in data['prices']]
        prices = [entry[1] for entry in data['prices']]

        # Convertir las marcas de tiempo UNIX a fechas legibles
        dates = [datetime.utcfromtimestamp(ts / 1000) for ts in timestamps]

        plt.figure(figsize=(10, 6))

        plt.plot(dates, prices, marker='o', label=f'Precio de {criptomoneda.capitalize()}')

        plt.xlabel('Fecha')
        plt.ylabel('Precio (USD)')

        plt.title(f'Variación del Precio de {criptomoneda.capitalize()} en los últimos 30 días')
        plt.legend()

        plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))
        plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=2))
```

```

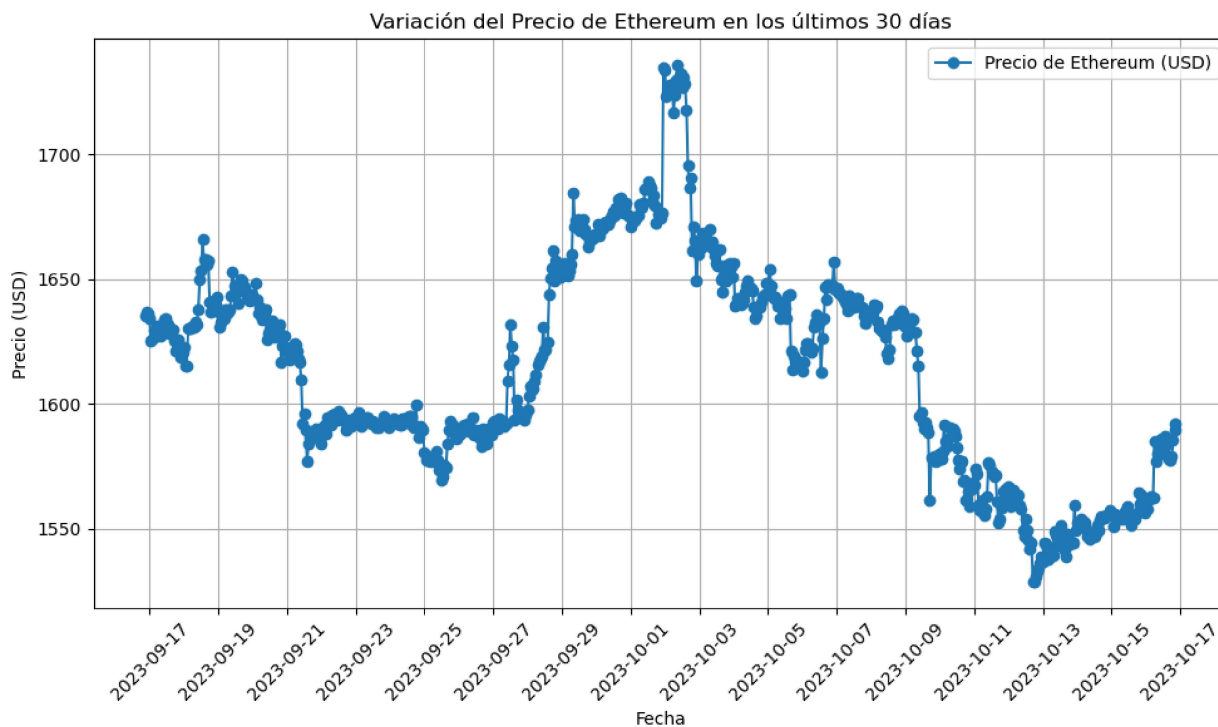
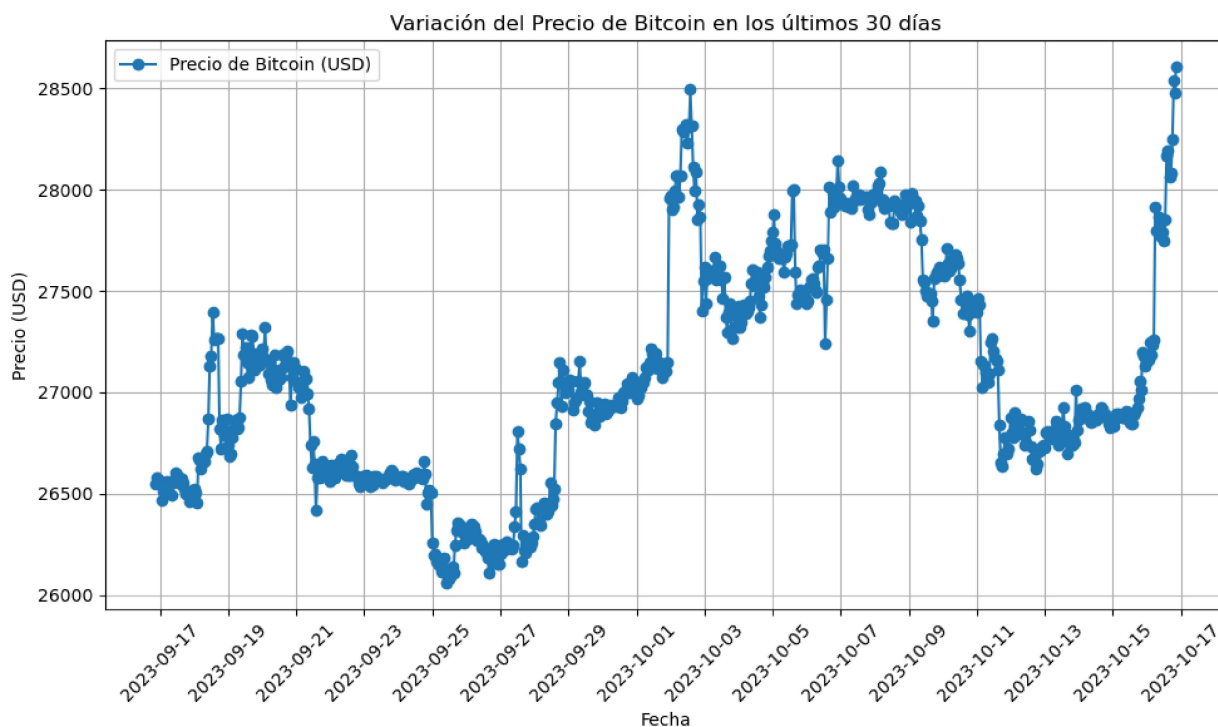
plt.xticks(rotation=45)
plt.grid(True)

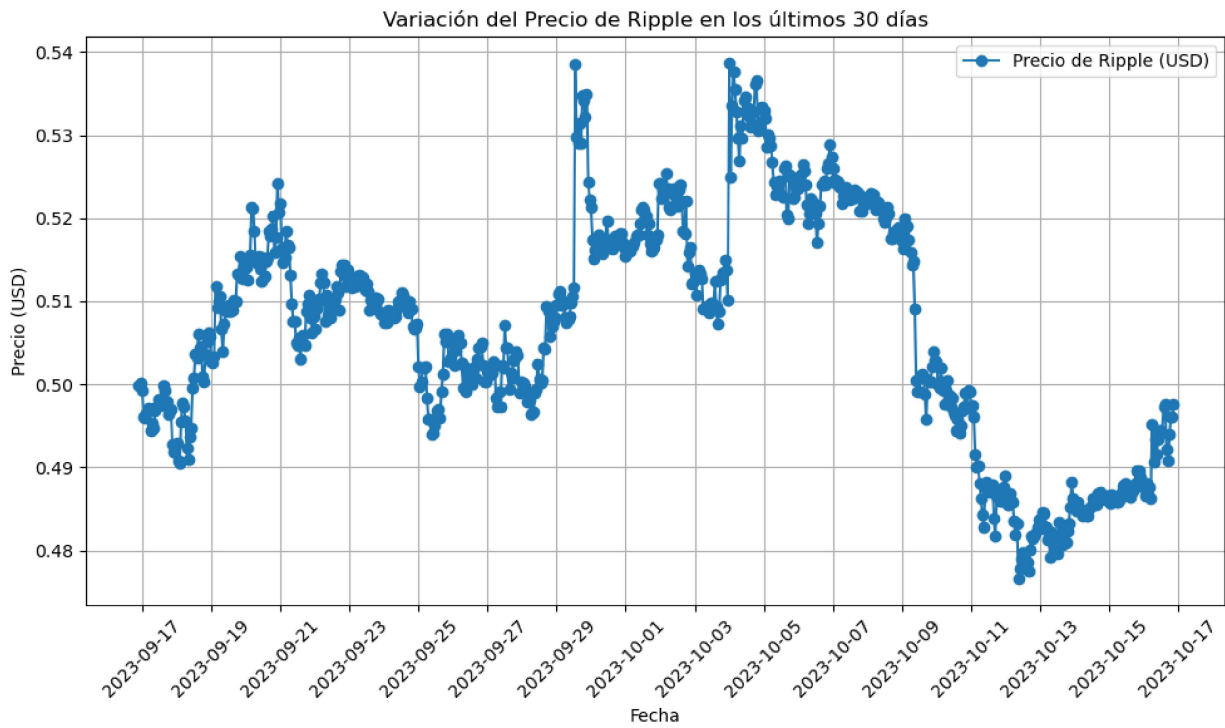
plt.tight_layout()
plt.show()

else:
    print(f"Error al obtener datos de {criptomoneda.capitalize()}. Código de estado

# Llamar a La función para cada criptomoneda en la lista
for criptomoneda in criptomonedas:
    obtener_datos_y_visualizar(criptomoneda)

```





In []: