

```
In [2]: from optparse import Values
from tkinter.tix import Tree
import requests as r
import datetime as dt
import pandas as p
import matplotlib.pyplot as plt
import sqlite3 as sql
import pandas as pd
import json
import plotly.graph_objs as go

#Selección de opción
opcion = 0
while True:
    opcion = input('Seleccione una opción: \n1 Actualización de datos\n2 Visualización de datos\n')
    #print(response.text)
    if opcion == '1':
        com = input("Ingrese ticker: ").upper()
        startTime= input("Ingrese una fecha de inicio con el siguiente formato (YYYY-MM-DD): ")
        año, mes, dia = map(int, startTime.split('-'))
        endTime= input("Ingrese una fecha de cierre con el siguiente formato (YYYY-MM-DD): ")
        año, mes, dia = map(int, endTime.split('-'))
        fecha = startTime + "/" + endTime
        url = "https://api.polygon.io/v2/aggs/ticker/"+ str(com) +"/range/1/day/"+ str(fecha)+ "?adjust=1"
        payload={}
        files={}
        headers = {}
        response = r.request("GET", url, headers=headers, data=payload, files=files)
        res = []
        data = response.json()
        for s in data['results']:
            res.append(s)
            df = pd.DataFrame(res)
            df ['Date'] = pd.to_datetime (df['t'], unit ='ms')
            df = df.rename(columns={
                'o': 'open',
                'c': 'close',
                'h': 'high',
                'l': 'low',
                'v': 'volume',
                'n': 'transactions'})
            conn = sql.connect ('df.db')
            df.to_sql ('df', conn, if_exists='replace')

            conn = sql.connect ('df.db')
            df = pd.read_sql ('SELECT * FROM df', conn)

        # Pedido al servidor
        def pedido_servidor(ticket,date_from,date_to):
            results = []
            url = "https://api.polygon.io/v2/aggs/ticker/"
            stocksT = ticket
            mult = '1'
            timestamp = 'day'
            adjusted = 'adjusted=true'
            sort = 'sort=asc'
            limit = '1000'
            apiKey = 'apiKey=G_Yb5HG6ld1S0k0SHQdr8w6Q4FKYJEVE'
            x = r.get(url+stocksT+'/range/'+mult+ '/' +timestamp+'/' + date_from+'/' + date_to +'?adjusted=1')
            y =x.json()
            date = 0
            if str(x.status_code)[0] == '4':
                print('Hubo un Error')
            else:
                if y['queryCount'] == 0: print('Sin Valor')
                else: results = y ['results']
            def check_ticker(name):
                pars = r.get("https://api.polygon.io/v2/aggs/ticker/"+name+"/range/1/day/2021-07-22/2021-07-22")
                complete_name = pars.json()['results']
                return complete_name
            #print(response.text)
        elif opcion == '2':
            fig = go.Figure()
            fig.add_trace(go.Candlestick(x=df.index,
                open=df['open'],
                high=df['high'],
                low=df['low'],
                close=df['close'], name = 'market data'))
            fig.update_layout(
                title='Evolución Precio',
                yaxis_title='Precio (USD)')
            fig.update_xaxes(
                rangeslider_visible=True,
                rangeselector=dict(
                    buttons=list([
                        dict(count=15, label="15m", step="minute", stepmode="backward"),
                        dict(count=45, label="45m", step="minute", stepmode="backward"),
                        dict(count=1, label="HTD", step="hour", stepmode="todate"),
                        dict(count=3, label="3h", step="hour", stepmode="backward"),
                        dict(step="all")
                    ])
                )
            )
            fig.show()
            break
        else: print('Opción invalida')
```

Seleccione una opción:
1 Actualización de datos
2 Visualización de datos
tsla
Opción invalida
Seleccione una opción:
1 Actualización de datos
2 Visualización de datos
1
Ingrese ticker: tsla
Ingrese una fecha de inicio con el siguiente formato (YYYY-MM-DD): 2021-01-01
Ingrese una fecha de cierre con el siguiente formato (YYYY-MM-DD): 2021-04-01
Seleccione una opción:
1 Actualización de datos
2 Visualización de datos
2

Evolución Precio

