

✓ Instituto Federal de Educação Ciência e Tecnologia de Brasília

Curso Técnico em Desenvolvimento de Sistemas

Desenvolvimento de Sistemas para Web

Alunos: Eduardo Moreira Pinto, Pedro Henrique Carvalho da Silva, Lídia Alves

Desenvolvemos esse projeto como parte da análise do preço do Bitcoin nos últimos 100 dias, coletando dados de uma API pública fornecida pela corretora OKX que fornece informações financeiras. Primeiramente foi criada uma estrutura de lista, definindo as colunas. Depois foi realizado o consumo da API, retornando os dados. Com os dados alocados, foi preciso realizar o tratamento dos dados, trabalhando a tipagem e formatação da data usando o módulo **datetime** do python.

A partir dos dados da API, tomamos liberdade para criar uma nova coluna a partir dessas informações; a coluna Volume(USDT), recebendo o valor a partir da multiplicação do Preço e volume em bitcoin.

Após tratar todos os dados, foi gerado um dataframe a partir da lista populada, utilizando a biblioteca pandas. Encontramos algumas barreiras, como a formatação da data que inicialmente tinha o valor na unidade (Timestamp Unix). Também na questão 07 ao criar uma nova linha, nos deparamos com dificuldades ao tentar usar o método `.append()`, consequentemente descobrimos que o método foi descontinuado e não está mais presente nas versões atuais do pandas.

No final, geramos um gráfico usando **Matplotlib** para visualizar as variações no valor de fechamento do Bitcoin durante o período dos últimos 100 dias. Baseado no gráfico abaixo, pode-se notar uma disparada no preço do bitcoin no final de 2024, atingindo a máxima de 107.000 dólares. Desde então, o preço tem variado entre 92.000 dólares e 105.000 dólares, mostrando um período de consolidação do mercado.


```
1 # Para contexto sobre o projeto, por favor, visualizar a questão #10 ao final do documento.
2
3
4 import requests
5 import pandas as pd
6 import matplotlib.pyplot as plt
7 from datetime import datetime
8 import pytz
9
10 days = 100 # Range da quantidade de dias que irão ser retornados. Obs: essa API gratuita limita até 100 dias.
11 url = f'https://www.okx.com/api/v5/market/history-candles?instId=BTC-USDT&limit={days}&bar=1D' #retorna o volume e preço de fechamento do gráfico diário bitcoin dos ultimos 30 dias
12
13 response = requests.get(url); #realiza a requisicao HTTP para a API da OKX, retornando o status, consequentemente os dados caso a requisição for bem sucedida
14
15 bitcoin = {
16     'timestamp': [],
17     'close_price': [],
18     'btc_volume': [],
19     'usdt_volume': [],
20 }
21
22 if response.status_code == 200 and response.json()['code'] == '0' :
23     data = response.json()['data'] #recebe os dados caso a requisicao for bem sucedida
```

```

24 else:
25     print(f'Erro na requisicao: \nStatus Code: {response.status_code} \n{response.json()["msg"]}') #printa o erro da requisicao, caso mal sucedida
26
27 if data:
28     for item in data:
29         timestamp = datetime.utcfromtimestamp(int(item[0]) / 1000) #converte de milissegundos para segundos
30         formatted_timestamp = timestamp.strftime("%d-%m-%Y")
31         close_price = round(float(item[4]), 1) #converte para float, uma casa decimal
32         btc_volume = round(float(item[5]), 2) #converte para float, duas casas decimais
33         usdt_volume = btc_volume * close_price #converte o volume em bitcoin para USDT. (volume em bitcoin * preço do bitcoin)
34         usdt_volume = "{:.2f}".format(usdt_volume) #remove a notação científica que vem por padrão do python ao lidar com numeros grandes
35
36         bitcoin['timestamp'].append(formatted_timestamp)
37         bitcoin['close_price'].append(close_price)
38         bitcoin['btc_volume'].append(btc_volume)
39         bitcoin['usdt_volume'].append(usdt_volume)
40
41     bitcoin_df = pd.DataFrame(bitcoin) #usa o pandas para criar o dataframe usando os dados acima
42     bitcoin_df.columns = ['Data(GMT+8)', 'Preço', 'Volume(BTC)', 'Volume(USDT)'] #renomea as colunas
43
44
45 # 01 - Apresente em tela toda a base de dados
46 print("Toda a base de dados:")
47 display(bitcoin_df)
48
49 # 02 - Apresentar o tamanho do DataFrame
50 print("\n\nTamanho do DataFrame:")
51 print(f"Linhas: {bitcoin_df.shape[0]}, Colunas: {bitcoin_df.shape[1]}")
52
53 # 03 - Acesse a linha (x) e apresente em tela todas as características do item
54 line_x = 2 # Defina o valor de x (o número da linha a ser acessada)
55 print(f"\n\nCaracterísticas da linha {line_x}:")
56 print(bitcoin_df.iloc[line_x])
57
58 # 04 - Verifique se o dataframe está vazio
59 print("\n\nVerificando se o DataFrame está vazio:")
60 if bitcoin_df.empty:
61     print("O dataframe não está preenchido")
62 else:
63     print("O dataframe está preenchido")
64
65 # 05 - Apresente em tela os 5 primeiros registros da base de dados
66 print("\n\nPrimeiros 5 registros:")
67 display(bitcoin_df.head())
68
69 # 06 - Exclua um item (linha) de sua base de dados.
70 line_to_exclude = 1 # Defina a linha a ser excluída
71 bitcoin_df = bitcoin_df.drop(index = line_to_exclude)
72 print(f"\n\nDataFrame após exclusão da linha { line_to_exclude }:")
73 display(bitcoin_df)
74
75 # 07 - Adicionar um item (linha) ao DataFrame
76 new_line = ['31-10-2024', 69225.4, 4990.44, 312463174.13]
77 bitcoin_df.loc[ len(bitcoin_df) + 1 ] = new_line #cria uma linha após a última, e recebe os dados de nova_linha
78 print("\n\nDataFrame após adicionar uma nova linha:")
79 display(bitcoin_df)
80

```

```
81 # 08 - Transpor uma coluna para uma linha
82 print("\n\nDataFrame transposto:")
83 display(bitcoin_df.T)
84
85 # 09 - Apresentar somente a 1ª e 2ª colunas do DataFrame
86 print("\n\nSomente as colunas 1 e 2:")
87 display(bitcoin_df.iloc[:, :2])
88
89 # 10 Informações sobre o projeto.
90
91 # Desenvolvemos esse projeto como parte da análise do preço do Bitcoin nos últimos 100 dias, coletando dados de uma API pública da corretora OKX
92 # que fornece informações financeiras. Primeiramente foi criado uma estrutura de lista, definindo as colunas. Depois foi realizado o consumo da API,
93 # retornando os dados. Com os dados alocados, foi preciso realizar o tratamento de tipagem, e formatação da data usando o módulo datetime do python.
94 # A partir dos dados da API, que retornou Volume(BTC) e Preço, tomamos liberdade para criar uma nova coluna a partir dessas informações; a coluna Volume(USDT),
95 # recebendo o valor a partir da multiplicação do preço e volume em bitcoin.
96 # Após tratar todos os dados, foi gerado um dataframe a partir da lista populada, utilizando a biblioteca pandas.
97 # Encontramos algumas barreiras, como a formatação da data que inicialmente tinha o valor na unidade (Timestamp Unix), e também na questão #7 ao criar uma nova
98 # linha, nos deparamos com dificuldades ao tentar usar o método .append(), consequentemente descobrimos que o método foi descontinuado e não está mais presente
99 # nas versões atuais do pandas.
100
101 # No final, geramos um gráfico usando Matplotlib para visualizar as variações no valor de fechamento do Bitcoin durante o período dos últimos 100 dias.
102 # Baseado no gráfico abaixo, pode-se notar uma disparada no preço do bitcoin no final de 2024, atingindo a máxima de 107.000$.
103 # Desde então, o preço tem variado entre 92.000$ e 105.000$, mostrando um período de consolidação do mercado.
104
105
106 # Aqui está o gráfico que fiz sobre a variação dos preços de fechamento do Bitcoin:
107 from matplotlib import pyplot as plt
108
109 # Convertendo a coluna 'Data(GMT+8)' para datetime
110 bitcoin_df['Data(GMT+8)'] = pd.to_datetime(bitcoin_df['Data(GMT+8)'], format='%d-%m-%Y')
111
112 # Ordenando os dados por data (caso não estejam)
113 bitcoin_df = bitcoin_df.sort_values(by='Data(GMT+8)')
114
115 # Criando o gráfico de preço ao longo do tempo
116 plt.figure(figsize=(8, 4.5)) #dimensao do tamanho do grafico
117 plt.plot(bitcoin_df['Data(GMT+8)'], bitcoin_df['Preço'], marker='o', linestyle='-', color='green', label='Preço do Bitcoin')
118
119 # Formatando o gráfico
120 plt.xlabel('Data (GMT+8)') # Define o nome do rótulo das datas.
121 plt.ylabel('Preço (USDT)') # Define o nome do rótulo dos preços.
122 plt.title('Variação do Preço do Bitcoin ao longo dos últimos 100 dias') # Título do gráfico
123 plt.xticks(rotation=45) # Rotacionar datas para melhor visualização
124 plt.legend() # Exibe as legendas, nesse caso: "Preço do Bitcoin", pois é a única linha existente no gráfico
125 plt.grid() # Cria as linhas de fundo do gráfico, para melhorar a visibilidade
126
127 # Exibir gráfico
128 print("\n\nExibição do gráfico")
129 plt.show()
130
131
132
```



Toda a base de dados:

	Data(GMT+8)	Preço	Volume(BTC)	Volume(USDT)
0	10-02-2025	97020.2	4772.86	463063831.77
1	09-02-2025	97224.0	7288.32	708599623.68
2	08-02-2025	96488.7	2937.97	283480905.94
3	07-02-2025	95855.6	5638.85	540515350.06
4	06-02-2025	98302.1	9588.19	942539212.20
...
95	07-11-2024	76418.8	10343.69	790452377.37
96	06-11-2024	76056.3	12139.15	923258834.14
97	05-11-2024	74296.4	30369.04	2256310343.46
98	04-11-2024	69632.1	10628.57	740089649.10
99	03-11-2024	68645.5	6845.03	469880506.87

100 rows × 4 columns

Tamanho do DataFrame:
Linhas: 100, Colunas: 4

Características da linha 2:
Data(GMT+8) 08-02-2025
Preço 96488.7
Volume(BTC) 2937.97
Volume(USDT) 283480905.94
Name: 2, dtype: object

Verificando se o DataFrame está vazio:
O dataframe está preenchido

Primeiros 5 registros:

	Data(GMT+8)	Preço	Volume(BTC)	Volume(USDT)
0	10-02-2025	97020.2	4772.86	463063831.77
1	09-02-2025	97224.0	7288.32	708599623.68
2	08-02-2025	96488.7	2937.97	283480905.94
3	07-02-2025	95855.6	5638.85	540515350.06
4	06-02-2025	98302.1	9588.19	942539212.20

DataFrame após exclusão da linha 1:

	Data(GMT+8)	Preço	Volume(BTC)	Volume(USDT)
0	10-02-2025	97020.2	4772.86	463063831.77
1	09-02-2025	97224.0	7288.32	708599623.68
2	08-02-2025	96488.7	2937.97	283480905.94
3	07-02-2025	95855.6	5638.85	540515350.06
4	06-02-2025	98302.1	9588.19	942539212.20

2	08-02-2025	96488.7	2937.97	283480905.94
3	07-02-2025	95855.6	5638.85	540515350.06
4	06-02-2025	98302.1	9588.19	942539212.20
5	05-02-2025	97338.6	6877.12	669409232.83
...
95	07-11-2024	76418.8	10343.69	790452377.37
96	06-11-2024	76056.3	12139.15	923258834.14
97	05-11-2024	74296.4	30369.04	2256310343.46
98	04-11-2024	69632.1	10628.57	740089649.10
99	03-11-2024	68645.5	6845.03	469880506.87

99 rows × 4 columns

DataFrame após adicionar uma nova linha:

	Data(GMT+8)	Preço	Volume(BTC)	Volume(USDT)
0	10-02-2025	97020.2	4772.86	463063831.77
2	08-02-2025	96488.7	2937.97	283480905.94
3	07-02-2025	95855.6	5638.85	540515350.06
4	06-02-2025	98302.1	9588.19	942539212.20
5	05-02-2025	97338.6	6877.12	669409232.83
...
96	06-11-2024	76056.3	12139.15	923258834.14
97	05-11-2024	74296.4	30369.04	2256310343.46
98	04-11-2024	69632.1	10628.57	740089649.10
99	03-11-2024	68645.5	6845.03	469880506.87
100	31-10-2024	69225.4	4990.44	312463174.13

100 rows × 4 columns

DataFrame transposto:

	0	2	3	4	5	6	7	8	9	10	...	91	92	
Data(GMT+8)	10-02-2025	08-02-2025	07-02-2025	06-02-2025	05-02-2025	04-02-2025	03-02-2025	02-02-2025	01-02-2025	31-01-2025	...	11-11-2024	10-11-2024	09-11-2024
Preço	97020.2	96488.7	95855.6	98302.1	97338.6	98114.6	99355.3	98831.9	99348.2	102074.3	...	86611.9	84068.2	79611.9
Volume(BTC)	4772.86	2937.97	5638.85	9588.19	6877.12	9829.71	13214.69	28649.1	6364.26	7410.79	...	33585.88	16175.66	13085.88
Volume(USDT)	463063831.77	283480905.94	540515350.06	942539212.20	669409232.83	964438064.77	1312949489.36	2831444986.29	632277775.33	756451201.70	...	2908936879.97	1359858620.01	1042516279.97

4 rows × 100 columns

Somente as colunas 1 e 2:

	Data(GMT+8)	Preço
0	10-02-2025	97020.2
2	08-02-2025	96488.7
3	07-02-2025	95855.6
4	06-02-2025	98302.1
5	05-02-2025	97338.6
...
96	06-11-2024	76056.3
97	05-11-2024	74296.4
98	04-11-2024	69632.1
99	03-11-2024	68645.5
100	31-10-2024	69225.4

100 rows × 2 columns

Exibição do gráfico

