

## ✓ Análise Histórica Financeira dos últimos 2 anos do cripto ativo Bitcoin

A respectiva análise visa identificar se o cripto ativo mais popular do mundo apresentou crescente significativa nesses anos, e se ainda apresenta oportunidades de investimento nos dias atuais considerando período histórico do ativo.

Observação: Este relatório não visa entregar uma recomendação de cripto ativo para investimento, mas sim, em analisar essa série histórica de 2 anos e comparar como o ativo se encontra nos dias atuais em relação ao seu comportamento no ano de 2022.

Para realização desta análise, foram utilizadas as seguintes bibliotecas:

### Bibliotecas para modelagem de dados

- Pandas
- Numpy

### Bibliotecas para acesso dos dados

- Yfinance

### Bibliotecas para análises gráficas

- Matplotlib
- Seaborn

### Biblioteca para análises gráficas (alto nível - Dashboards)

- Plotly

### Biblioteca para conversão para data

- Datetime

### Biblioteca para acesso dos dados

- Yfinance

### Biblioteca para Avisos/alerta

- Warning

Instalação da biblioteca Yahoo Finance.

```
!pip install Yfinance --upgrade --no-cache-dir
```

```
Requirement already satisfied: Yfinance in /usr/local/lib/python3.10/dist-packages (0.2.44)
Requirement already satisfied: pandas>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from Yfinance) (2.2.2)
Requirement already satisfied: numpy>=1.16.5 in /usr/local/lib/python3.10/dist-packages (from Yfinance) (1.26.4)
Requirement already satisfied: requests>=2.31 in /usr/local/lib/python3.10/dist-packages (from Yfinance) (2.32.3)
Requirement already satisfied: multitasking>=0.0.7 in /usr/local/lib/python3.10/dist-packages (from Yfinance) (0.0.11)
Requirement already satisfied: lxml>=4.9.1 in /usr/local/lib/python3.10/dist-packages (from Yfinance) (4.9.4)
Requirement already satisfied: platformdirs>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from Yfinance) (4.3.6)
Requirement already satisfied: pytz>=2022.5 in /usr/local/lib/python3.10/dist-packages (from Yfinance) (2024.2)
Requirement already satisfied: frozendict>=2.3.4 in /usr/local/lib/python3.10/dist-packages (from Yfinance) (2.4.4)
Requirement already satisfied: peewee>=3.16.2 in /usr/local/lib/python3.10/dist-packages (from Yfinance) (3.17.6)
Requirement already satisfied: beautifulsoup4>=4.11.1 in /usr/local/lib/python3.10/dist-packages (from Yfinance) (4.12.3)
Requirement already satisfied: html5lib>=1.1 in /usr/local/lib/python3.10/dist-packages (from Yfinance) (1.1)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4>=4.11.1->Yfinance) (2.6)
Requirement already satisfied: six>=1.9 in /usr/local/lib/python3.10/dist-packages (from html5lib>=1.1->Yfinance) (1.16.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from html5lib>=1.1->Yfinance) (0.5.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.3.0->Yfinance) (2.8)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.3.0->Yfinance) (2024.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->Yfinance) (3.10)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->Yfinance) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->Yfinance) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->Yfinance) (2024.8.30)
```

Importação de bibliotecas para manipulação e apresentação dos dados:

```
#Modelagem de dados
import pandas as pd
import numpy as np
import pandas_datareader.data as web

#Análises gráficas
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#Análises gráficas (alto nível)
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots

#Data
import datetime

#Dados
import yfinance as yf
# Args personalizados
#yf.pdr_override()

#Avisos
import warnings
warnings.filterwarnings('ignore')
```

## ✓ API Yahoo Finance

O yfinance é uma biblioteca que permite baixar dados de mercado financeiro do site Yahoo!® finanças.

```
#Ativo de interesse - Bitcoin
ticker_bitcoin = 'BTC-USD'
```

```
#Período dos dados (2021 a 2024)
inicio = '2022-01-01'
fim = '2024-10-01'
```

```
#Coletando os dados
Dados = yf.download(ticker_bitcoin, start=inicio, end=fim)
```

```
[*****100%*****] 1 of 1 completed
```

```
#Formato dos dados:
```

```
type(Dados)
```

```
pandas.core.frame.DataFrame
def __init__(data=None, index: Axes | None=None, columns: Axes | None=None, dtype: Dtype |
None=None, copy: bool | None=None) -> None

/usr/local/lib/python3.10/dist-packages/pandas/core/frame.py
Two-dimensional, size-mutable, potentially heterogeneous tabular data.

Data structure also contains labeled axes (rows and columns).
Arithmetic operations align on both row and column labels. Can be
thought of as a dict-like container for Series objects. The primary
```

```
#Verificação do formato dos dados armazenados em Dados:
```

```
Dados.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1004 entries, 2022-01-01 to 2024-09-30
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Open        1004 non-null    float64
1   High        1004 non-null    float64
2   Low         1004 non-null    float64
3   Close       1004 non-null    float64
4   Adj Close   1004 non-null    float64
5   Volume      1004 non-null    int64
dtypes: float64(5), int64(1)
memory usage: 54.9 KB
```

```
#Análise dos primeros registros:
```

```
Dados.head()
```

	Open	High	Low	Close	Adj Close	Volume
Date						
2022-01-01	46311.746094	47827.312500	46288.484375	47686.812500	47686.812500	24582667004
2022-01-02	47680.925781	47881.406250	46856.937500	47345.218750	47345.218750	27951569547
2022-01-03	47343.542969	47510.726562	45835.964844	46458.117188	46458.117188	33071628362
2022-01-04	46458.851562	47406.546875	45752.464844	45897.574219	45897.574219	42494677905
2022-01-05	45899.359375	46929.046875	42798.222656	43569.003906	43569.003906	36851084859

Próximas etapas: 

Gerar código com Dados

Ver gráficos recomendados

New interactive sheet

Para análise de série histórica de ativos financeiros, é importante obter dados no formato OHLC (Open, High, Low, Close), o que favorece a elaboração de gráficos no formato de candlestick e facilita a interpretação dos analistas.

Dados.shape

(1004, 6)

Temos um dataSet com 1004 registros, ou seja, uma alta quantidade de cotações diárias do ativo Bitcoin, o que nos permite fazer uma análise do comportamento de seu preço no intervalo de tempo dos 3 últimos anos.

# Estatísticas descritivas dos dados

Dados.describe()

	Open	High	Low	Close	Adj Close	Volume
count	1004.000000	1004.000000	1004.000000	1004.000000	1004.000000	1.004000e+03
mean	37127.411440	37831.169607	36398.269504	37144.927437	37144.927437	2.610781e+10
std	16371.964671	16723.721005	15990.631829	16390.799175	16390.799175	1.320737e+10
min	15782.300781	16253.047852	15599.046875	15787.284180	15787.284180	5.331173e+09
25%	23640.585449	24157.261719	23198.809570	23640.781738	23640.781738	1.667016e+10
50%	30313.547852	30720.304688	29885.424805	30316.925781	30316.925781	2.460724e+10
75%	45630.405273	46852.921875	44411.678711	45634.231445	45634.231445	3.235234e+10
max	73079.375000	73750.070312	71334.093750	73083.500000	73083.500000	1.189925e+11

Acima é possível notar uma diferença significativa em mínimo e máximo do ativo no intervalo de tempo em análise. Antes de avançar na análise, já é possível notar uma alta significativa no fechamento ajustado do ativo. De valor em 15787.28 dólares para o máximo de 73083.50 dólares, é possível observar que em 2 anos houve uma alta de mais de U\$57k.

Média Móvel

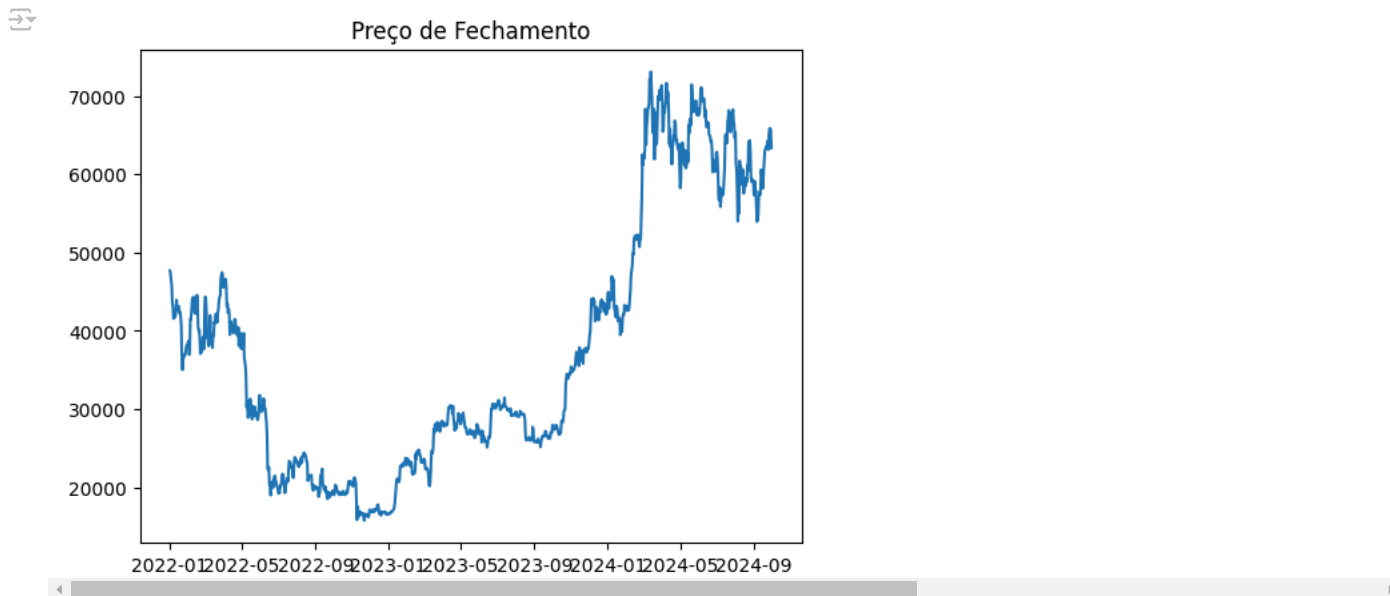
Media Móvel

A média móvel é um indicador que faz parte da Análise Técnica de ativos financeiros. Com ela, é possível identificar o equilíbrio dos preços no mercado, observando tendências de alta, neutra ou baixa. A representação gráfica das médias móveis é normalmente feita por uma linha, que se movimenta conforme os dados novos recebidos para o cálculo.

Observação: É UM INDICADOR DE TENDÊNCIA (alta, baixa ou neutra).

#Gráfico para análise do preço de fechamento da série histórica dos 3 anos

```
plt.plot(Dados['Close'])
plt.title('Preço de Fechamento');
```



A variável target para essa análise de tendência do preço está na coluna Close (fechamento diário) do cripto ativo.

Para melhora da experiência em observar o fechamento do ativo nesses 3 anos, abaixo será inseria médias móveis para análise deste gráfico, o que permite interpretar a tendência que esse preço teve conforme os intervalos de tempo.

### Função rolling

A função rolling() é uma função que permite calcular estatísticas em um range de dados.

Ela é muito útil para calcular médias móveis, desvios padrão, somatórios, mínimos e máximos, entre outras estatísticas.

Abaixo a função está fazendo o agrupamento de dados a cada 5 dias para calcular a média móvel de 5 períodos.

Período = 5

Dados['Close'].rolling(window=Período).mean() #média móvel no período de 5 dias

Date	Close
2022-01-01	NaN
2022-01-02	NaN
2022-01-03	NaN
2022-01-04	NaN
2022-01-05	46191.345313
...	...
2024-09-26	63920.928906
2024-09-27	64349.319531
2024-09-28	64860.889062
2024-09-29	65127.556250
2024-09-30	65164.827344

1004 rows × 1 columns

Agora vamos melhorar o plot, basicamente tendo como referência o matplotlib usado anteriormente:

```
# Plot do preço de fechamento com a média móvel

# Estilos
plt.style.use('seaborn-darkgrid')

# Tamanho
plt.figure(figsize=(12,6))

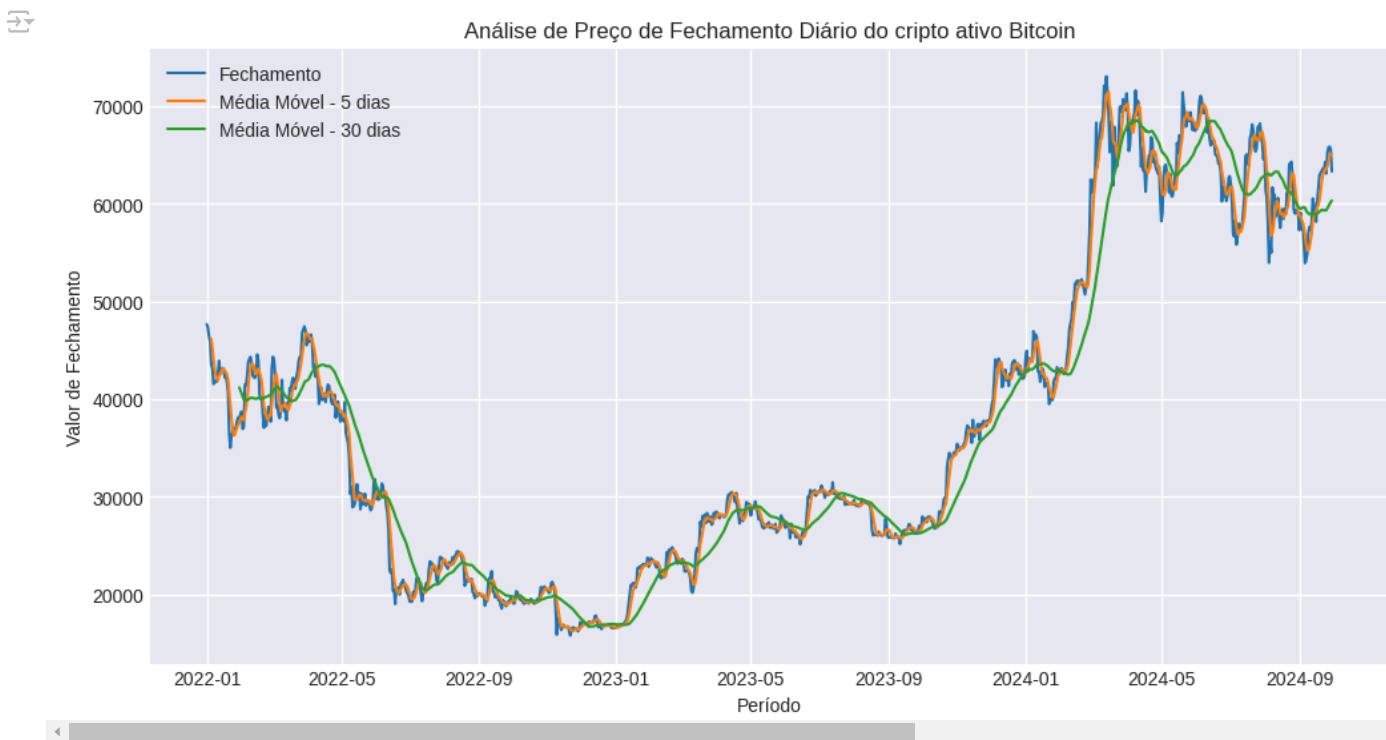
# Plots dos dados (fechamento, média móvel de 5 e 30 períodos)
plt.plot(Dados['Close'])
plt.plot(Dados['Close'].rolling(window=Período).mean())
```

```
plt.plot(Dados['Close'].rolling(window=Periodo + 25).mean())

# Legenda
plt.legend(['Fechamento', "Média Móvel - 5 dias", "Média Móvel - 30 dias"])

# Labels (x e y)
plt.ylabel("Valor de Fechamento")
plt.xlabel("Período")

plt.title('Análise de Preço de Fechamento Diário do cripto ativo Bitcoin');
```



As médias móveis suavizam a observação do preço, minimizando a oscilação do preço, facilitando a visualização da tendência.

OBSERVAÇÃO: Analistas costumam observar médias móveis de 30 dias.

Plotly - Relatórios dinâmicos

O PLOTLY é uma biblioteca Python de plotagem interativa e de código aberto que oferece suporte a mais de 40 tipos de gráficos exclusivos, abrangendo uma ampla variedade de casos de uso estatísticos, financeiros, geográficos, científicos e tridimensionais.

Abaixo será criada uma nova coluna para o dataframe, esta inserindo a média móvel de 5 períodos para cada preço do ativo em análise.

```
Period = 5
Dados['Media_Movel'] = Dados['Close'].rolling(window=Periodo).mean()

Dados.head()
```

	Open	High	Low	Close	Adj Close	Volume	Media_Movel
Date							
2022-01-01	46311.746094	47827.312500	46288.484375	47686.812500	47686.812500	24582667004	NaN
2022-01-02	47680.925781	47881.406250	46856.937500	47345.218750	47345.218750	27951569547	NaN
2022-01-03	47343.542969	47510.726562	45835.964844	46458.117188	46458.117188	33071628362	NaN
2022-01-04	46458.851562	47406.546875	45752.464844	45897.574219	45897.574219	42494677905	NaN
2022-01-05	45899.359375	46929.046875	42798.222656	43569.003906	43569.003906	36851084859	46191.345313

Próximas etapas:

Gerar código com Dados

☒ Ver gráficos recomendados

New interactive sheet

Criando um gráfico com o plotly express px:

```
px.line(Dados, y='Close', title='Análise de Fechamento do Bitcoin - Anos base: 2022 a 2024')
```



### Análise de Fechamento do Bitcoin - Anos base: 2022 a 2024



# Plotly GO - Biblioteca que permite inserir mais informações nos gráficos a serem plotados.

```
Figure_02 = go.Figure(
    data=go.Scatter(
        x=Dados.index,
        y=Dados['Close'],
        line=dict(color='firebrick', width=3)))
```

```
Figure_02.update_layout(
    title='Análise de Fechamento',
    xaxis_title='Período',
    yaxis_title="Preço de fechamento")
```

```
Figure_02.show()
```



### Análise de Fechamento



# Plot de gráficos de Area

```
Area = px.area(
    data_frame=Dados.reset_index(),
    y='Close', x='Date',
    title=f'{ticker_bitcoin}')
```

# Atualiza as configurações dos eixos x (datas) do gráfico, e inserindo botões

```

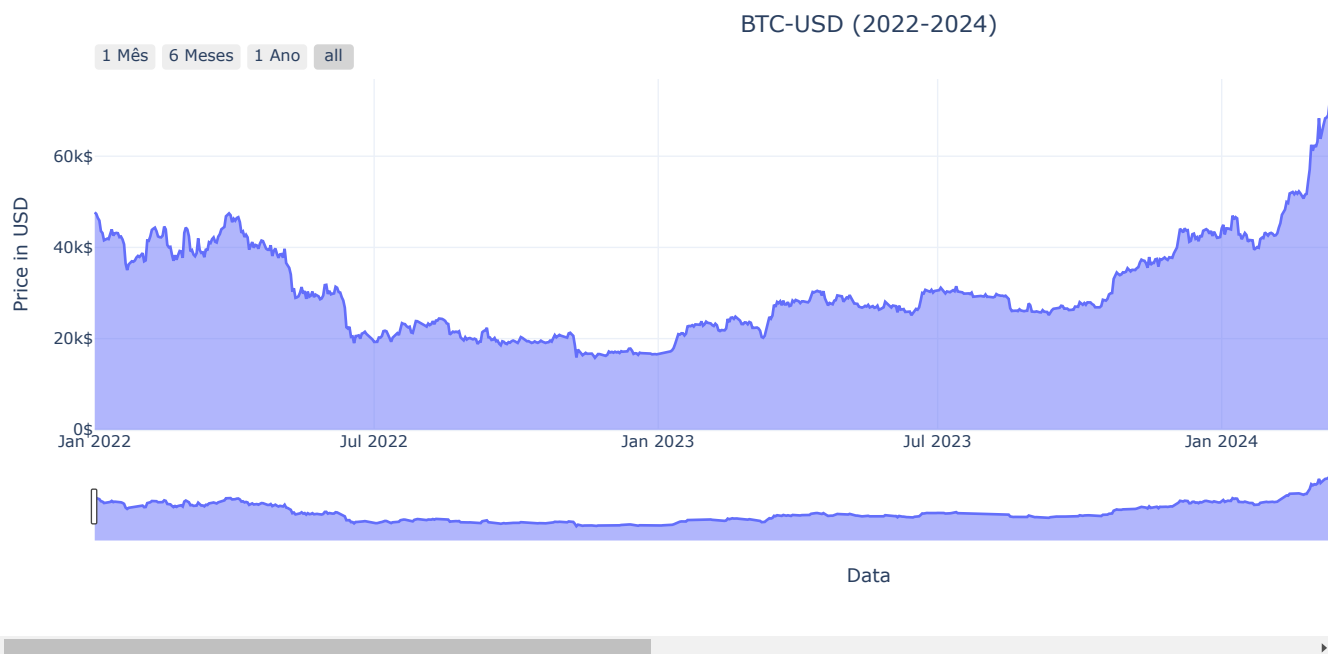
Area.update_xaxes(
    title_text='Data',
    rangelslider_visible=True,
    rangeselector=dict(
        buttons=list([
            dict(count=1, label='1 Mês', step='month', stepmode='backward'),
            dict(count=6, label='6 Meses', step='month', stepmode='backward'),
            dict(count=1, label='1 Ano', step='month', stepmode='backward'),
            dict(step='all')]))))

# Ajustando o eixo Y
Area.update_yaxes(title_text='Price in USD', ticksuffix='$')

# Ajustando o título
Area.update_layout(
    showlegend=True,
    title={
        'text': f'{ticker_bitcoin} ({Dados.index.min().year}-2024)',
        'y': 0.9,
        'x': 0.5,
        'xanchor': 'center',
        'yanchor': 'top'},
    template='plotly_white')

```

Area



## Gráficos de Candlesticks

É um estilo de gráfico financeiro que descreve abertura, alta, baixa e fechamento para uma determinada xcoordenada (tempo mais provável). As caixas representam a dispersão entre os valores open e close e as linhas representam a dispersão entre os valores low e high. Pontos de amostragem onde o valor de fechamento é maior (inferior) do que o valor de abertura são chamados de crescentes (decrecentes). Por padrão, as velas crescentes são desenhadas em verde, enquanto as decrecentes são desenhadas em vermelho.

```

Gráfico_Candlestick = go.Figure(
    data=[go.Candlestick(
        x=Dados.index,
        open=Dados['Open'],
        high=Dados['High'],
        low=Dados['Low'],
        close=Dados['Close'],
        increasing_line_color='green',
        decreasing_line_color='red')])

Gráfico_Candlestick.update_layout(
    xaxis_rangelslider_visible = False,
    title='Análise Fechamento - Bitcoin (2022-2024)',
    xaxis_title='Período',
    yaxis_title='Preço de Fechamento')

Gráfico_Candlestick.show()

```



## Análise Fechamento - Bitcoin (2022-2024)



## ▼ Criando Relatório

Gerar relatório com gráfico dinâmico para análise e estudo do comportamento do preço de fechamento do crypto ativo nos últimos 2 anos.

#Construção do Relatório

#Usando a função make\_subplots da biblioteca Plotly

```
Relatorio = make_subplots(
    rows = 2,                #2 linhas
    cols = 1,                #1 coluna
    specs = [
        [{'type':'scatter'}], #1 grafico de linha
        [{'type':'scatter'}], #+1 grafico de linha
    ],
    subplot_titles=(('Cotação','Fechamento'), #títulos dos gráficos no relatórios (2 título no relatório)
    #Compartilha o mesmo eixo X para ambos gráficos
    shared_xaxes = True,      #como ambos 2 eixos X possuem o mesmo período, aqui ele exclui a legenda do eixo X do gráfico de cima.
    # Largura Vertical
    vertical_spacing=0.075,)
```

#Ajuste do Layout

```
Relatorio.update_layout(
    #Dimensão
    width=1000, height=800, #largura e altura

    #Título do relatório
    title_text = '<b> Análise histórica de fechamento - Bitcoin </b> <br> Período de 2022 a 2024',

    #Estilo
    template='plotly_dark',

    #Ajuste de layout gráfico
    paper_bgcolor = 'rgb(0,0,0)', #Cor do fundo do documento
    plot_bgcolor = 'rgb(10,10,10)', #cor do fundo do gráfico

    #Fonte do texto
    font=dict(color='white'),)
```

#Colocando um gráfico na 1ª Linha (método .add\_trace())

```
Relatorio.add_trace(
    go.Candlestick(
        x=Dados.index,
        open=Dados['Open'],
        high=Dados['High'],
        low=Dados['Low'],
        close=Dados['Close'],
        name = 'fechamento',
        increasing_line_color='green', #quando o preço subir fica nessa cor
        decreasing_line_color='red' #quando o preço descer fica nessa outra cor.
    ), row=1, col=1 #colocando o gráfico na 1ª linha e 1ª coluna do layout
```



```

)

#Retirando a barra visível (barra abaixo da cotação)
Relatorio.update_layout(
    xaxis_rangeslider_visible = False,)

#Adicionando um gráfico na 2ª Linha
Relatorio.add_trace(
    go.Scatter(
        x=Dados.index,
        y=Dados['Close'],
        mode='lines',
        name='Fechamento',
        line=dict(color='green')
    ), row=2, col=1)

#Adicionando a Média Móvel no segundo gráfico
Relatorio.add_trace(
    go.Scatter(
        x=Dados.index,
        y=Dados['Media_Movel'],
        mode='lines',
        name='Média Móvel',
        line=dict(color='rgb(158,58,171)')
    ), row=2, col=1)

#Ajustes
Relatorio.update_layout(
    legend=dict(
        #Posição da legenda
        orientation='h',
        #horizontal

        #Eixo X e Y
        yanchor='bottom',
        #fica alinhado no máximo do y
        y=1.02,
        xanchor='right',
        x=1,
        #Fica alinhado ao máximo do x (direita)
        font=dict(size=9)
        #tamanho da fonte da legenda
    ))

#Adicionando Texto (comentário no relatório)
Relatorio.add_annotation(
    #Seta apontando para a função do gráfico (desativada)
    showarrow=False,
    #Texto
    text='Análise realizada para estudo da série histórica do cripto ativo Bitcoin',
    #Tamanho
    font=dict(size=12),

    #Aponta as direções do texto no relatório
    xref = 'x domain',
    x=1,
    yref = 'y domain',
    y=-1.4)

Relatorio.add_annotation(
    showarrow=False,
    text='Desenvolvido por: Jefferson Juarez do Nascimento',
    font=dict(size=12),
    xref = 'x domain',
    x=0,
    yref = 'y domain',
    y=-1.4)

Relatorio

```



## Exportando PDF

Para gerar o relatório em pdf, foi utilizada a API Kaleido, a qual permite exportar o respectivo relatório final no formato de pdf. Vale ressaltar que esta biblioteca já está presente dentro da biblioteca Plotly.

A API já está instalada dentro do plotly, mas, caso apresente erro, basta rodar uma nova instalação da biblioteca:

```
!pip install -U kaleido
```

## Reiniciando o servidor

" Clicar em: Ambiente de execução/Reiniciar e executar tudo "

```
#Reiniciando o Servidor  
#!pip install kaleido
```

```
#Exportando o relatório em .pdf  
Relatorio.write_image('Report_Financeiro_análise_bitcoin.pdf')
```