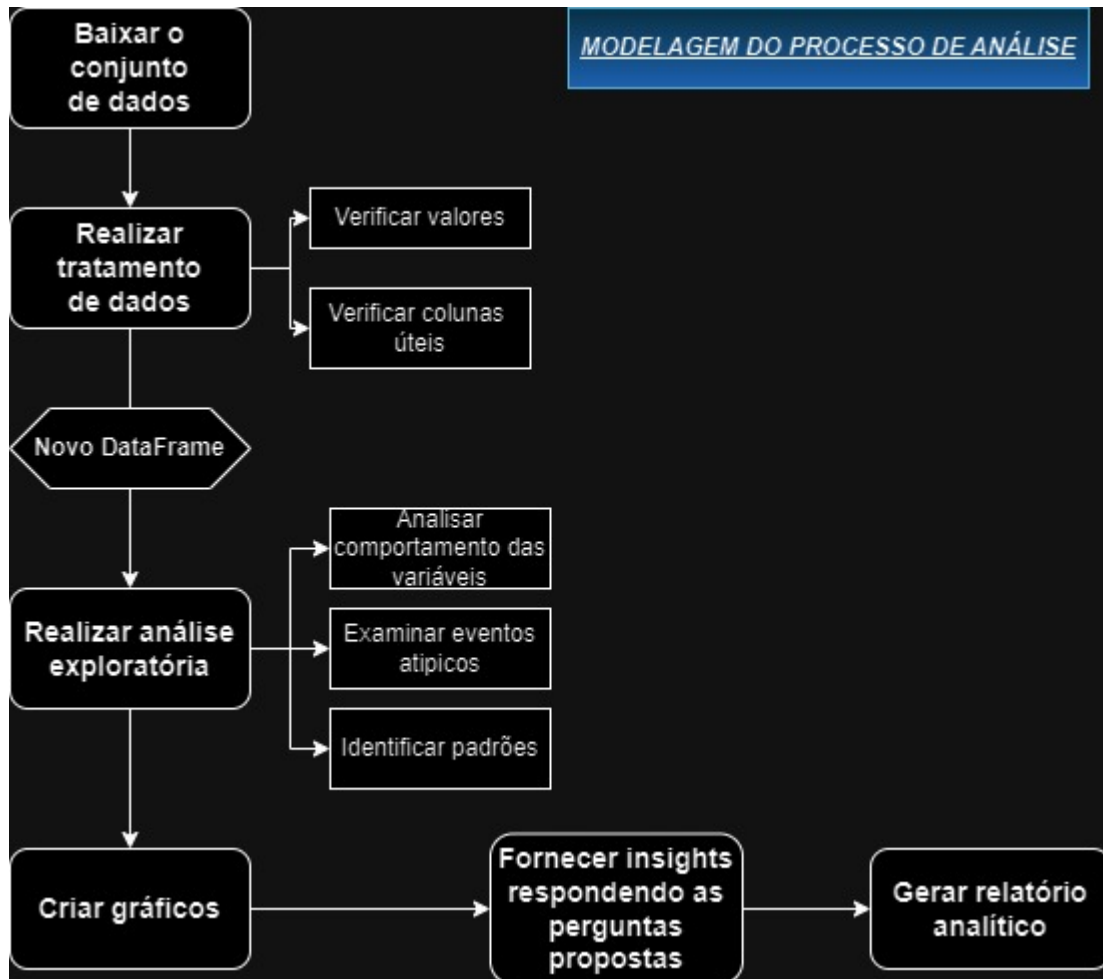


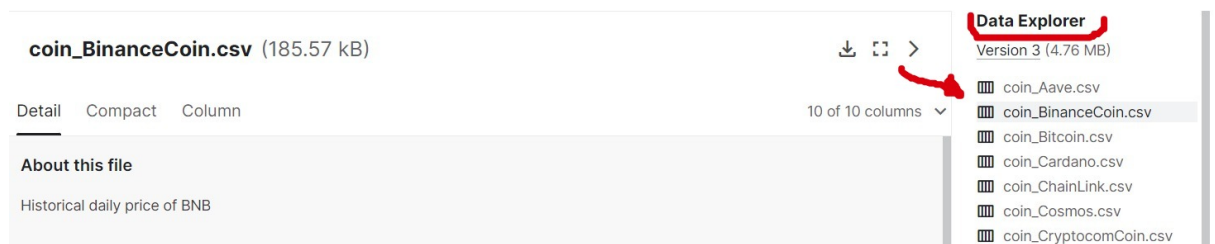
DOCUMENTAÇÃO DO PROCESSO DE USO DE FERRAMENTAS E COMPOSIÇÃO DO RELATÓRIO

Para realizar este relatório de análise, foi necessário cumprir algumas etapas que serão descritas neste documento em detalhe, para que se entenda o processo.



1. ETL = Extract, treatment, load

- Em primeiro lugar, foram extraídos 10 arquivos de criptomoedas randomicamente do site da Kaggle (<https://www.kaggle.com/datasets/sudalairajkumar/cryptocurrencypricehistory>);



#

Moedas selecionadas:

- Aave
 - BinanceCripto
 - Bitcoin
 - Chainlink
 - Cosmos
 - EOS
 - Ethereum
 - Polkadot
 - USDCoin
 - XRP
- Após a extração, foi realizado o tratamento dos dados utilizando a biblioteca *Pandas* do *Python* pela ferramenta *Google Colab* para verificar possíveis inconsistências nos valores que pudessem embarçar a análise, feito isso, os arquivos foram concatenados em um e transformados em .csv;

#

```
# Importante bibliotecas
import pandas as pd
```

```
# Extrair arquivo
aave = pd.read_csv("coin_Aave.csv")
aave
```

```
# Concatenando todas as tabelas de criptomoedas
crypto_coins = pd.concat([aave, binance, bitcoin, chainlink, cosmos, eos, ethereum, polkadot, usd, xrp]
crypto_coins
```

```
# convertendo dataframe para csv
crypto_coins.to_csv("crypto_coins.csv")
```

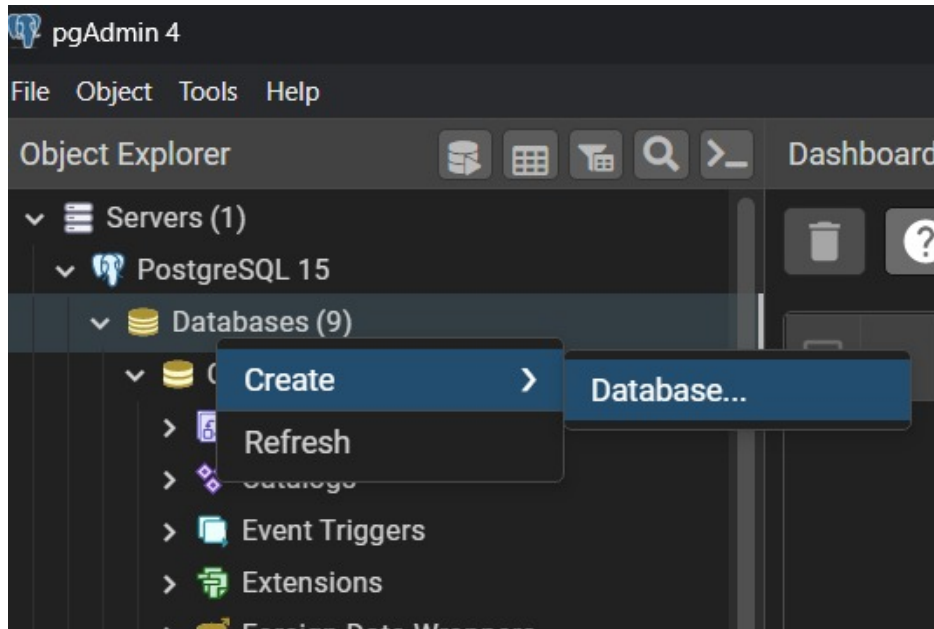
ETL - os processos de tratamento estão todos justificados no markdown:

https://github.com/mathuscm/Criptomoedas-m5/blob/main/Projeto_Criptomoedas.ipynb

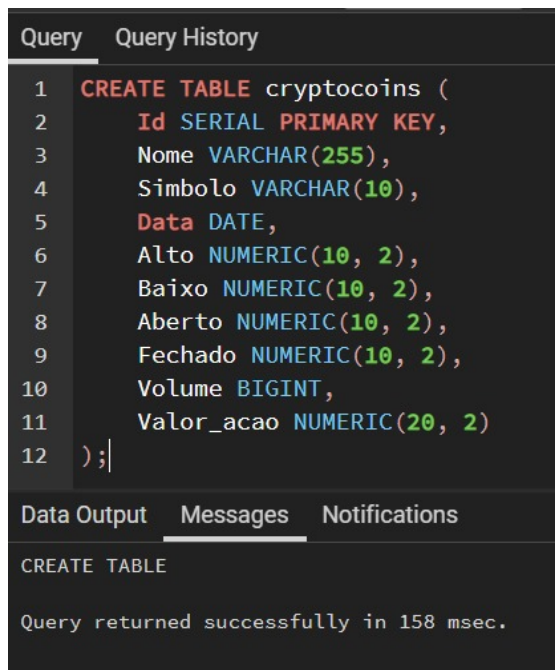
- Depois de tratar os dados e criar um novo arquivo, ele está pronto para ser carregado e manipulado no banco de dados

2. Criando Database no PostgreSQL

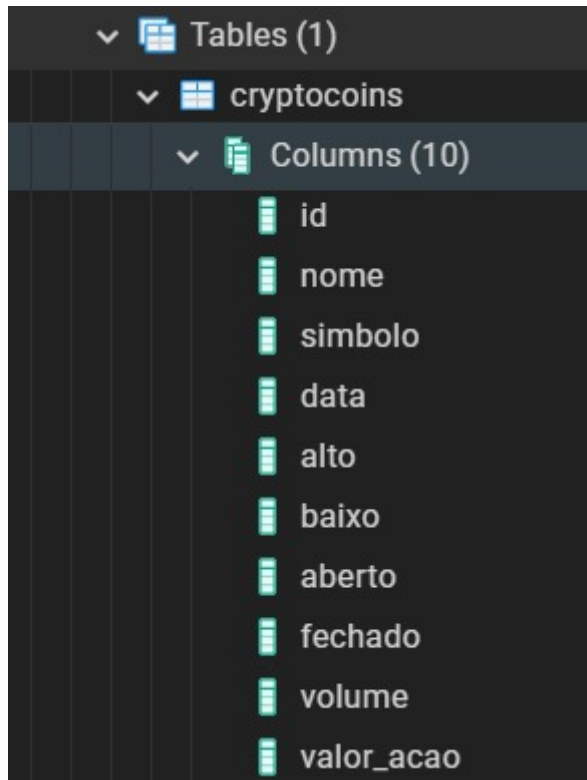
- Nesta segunda etapa e com o arquivo novo preparado, foi utilizado o *PostgreSQL* no *pgAdmin* para criar um novo *Banco de Dados* em que seria carregado o arquivo *crypto_coins.csv*



- Após criar o BD, foi criada a entidade para receber os dados do arquivo *crypto_coins.csv*



#



2.1. Inserindo os dados utilizando script com python

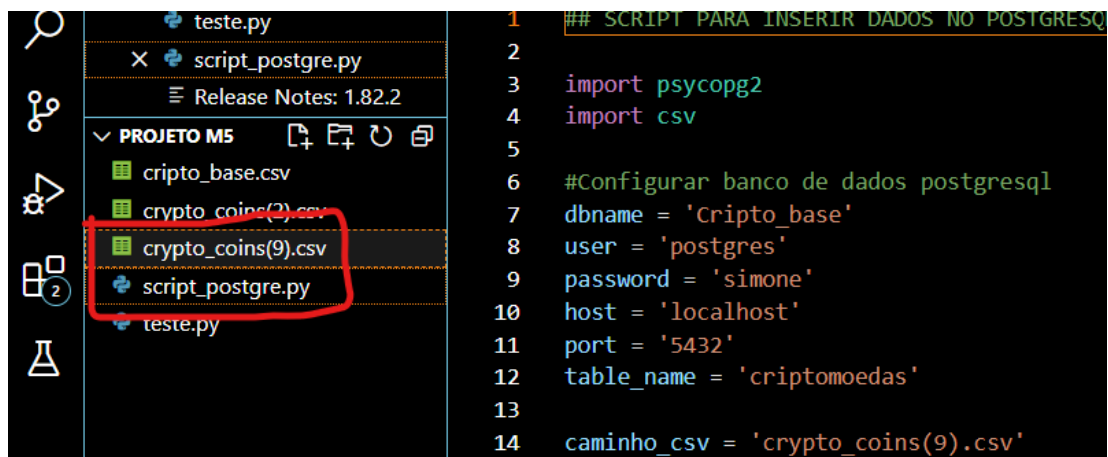
- Foi criado um *script* utilizando *python* e as bibliotecas *csv* para conseguir acessar os dados do arquivo e *psycopg2* para acessar o banco de dados no *pgAdmin* e importar os dados da *crypto_coins.csv* para a entidade *criptomoedas* dentro do BD *Cripto_base*

Nos campos de configuração do script é inserido as informações do banco de dados

q

```
deucerto.py > ...
1  ## SCRIPT PARA INSERIR DADOS NO POSTGRESQL
2
3  import psycopg2
4  import csv
5
6  #Configurar banco de dados postgresql
7  dbname = 'Cripto_base'
8  user = 'postgres'
9  password = 'simone'
10 host = 'localhost'
11 port = '5432'
12 table_name = 'criptomoedas'
13
14 caminho_csv = 'crypto_coins(9).csv'
15
16 try:
17     # acessa o banco de dados
18     conn = psycopg2.connect(dbname=dbname, user=user, password=password, host=host, port=port)
19     cursor = conn.cursor()
20
21     # Abre o arquivo CSV e insere os dados na tabela
22     with open(caminho_csv, 'r') as csv_file:
23         reader = csv.reader(csv_file)
24         next(reader)
25         for row in reader:
26             # Arredonda o valor da coluna "Valor_acao"
27             valor_acao_arredondado = round(float(row[9]), 2)
28
29             cursor.execute(
30                 f"INSERT INTO {table_name} (Id, Nome, Simbolo, Data, Valor_acao, Valor_acao_arredondado) VALUES ({row[0]}, {row[1]}, {row[2]}, {row[3]}, {row[4]}, {valor_acao_arredondado})"
```

Em *caminho_csv* é feita a busca pelo arquivo que se encontra dentro da mesma pasta do *script_postgre.py*



```
1  ## SCRIPT PARA INSERIR DADOS NO POSTGRESQL
2
3  import psycopg2
4  import csv
5
6  #Configurar banco de dados postgresql
7  dbname = 'Cripto_base'
8  user = 'postgres'
9  password = 'simone'
10 host = 'localhost'
11 port = '5432'
12 table_name = 'criptomoedas'
13
14 caminho_csv = 'crypto_coins(9).csv'
```

#

```

deucerto.py > ...
3 import psycopg2
4 import csv
5
6 #Configurar banco de dados postgresql
7 dbname = 'Cripto_base'
8 user = 'postgres'
9 password = '...'
10 host = 'localhost'
11 port = '5432'
12 table_name = 'criptomoedas'
13
14 caminho_csv = 'crypto_coins(9).csv'
15

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** Python + ▾

PS C:\Users\Nifrias\Desktop\Arquivos Python\Projeto M5> & C:/Users/Nifrias/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/Nifrias/Desktop/Arquivos Python/Projeto M5/dados inseridos com sucesso!

- Após realizar estes procedimentos, ao fazer uma consulta é possível verificar os dados

Query Query History

1 select * from criptomoedas;

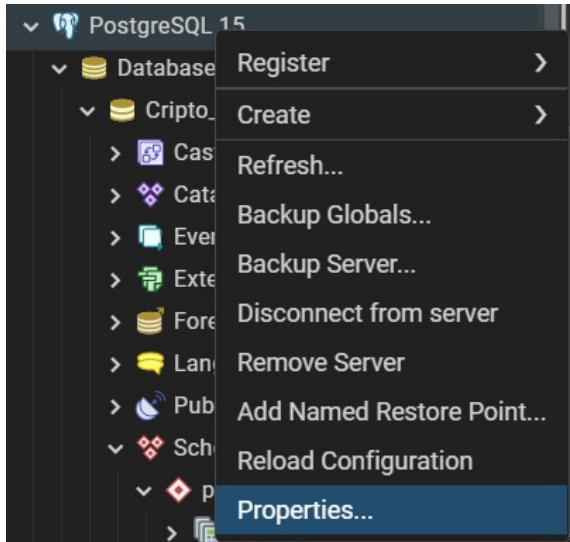
Data Output Messages Notifications

	id [PK] integer	nome character varying (255)	simbolo character varying (10)	data date	alto numeric (10,2)	baixo numeric (10,2)	aberto numeric (10,2)
1	0	Aave	AAVE	2020-10-05	55.11	49.79	52.6
2	1	Aave	AAVE	2020-10-06	53.40	40.73	53.2
3	2	Aave	AAVE	2020-10-07	42.41	35.97	42.4
4	3	Aave	AAVE	2020-10-08	44.90	36.70	39.8
5	4	Aave	AAVE	2020-10-09	47.57	43.29	43.7
6	5	Aave	AAVE	2020-10-10	51.41	46.70	46.8
7	6	Aave	AAVE	2020-10-11	51.45	48.72	49.1
8	7	Aave	AAVE	2020-10-12	54.42	48.75	49.6
9	8	Aave	AAVE	2020-10-13	57.48	49.60	52.2
10	9	Aave	AAVE	2020-10-14	57.85	49.63	51.3
11	10	Aave	AAVE	2020-10-15	52.08	44.02	51.3
12	11	Aave	AAVE	2020-10-16	43.91	39.14	42.8
13	12	Aave	AAVE	2020-10-17	43.08	38.63	39.4
14	13	Aave	AAVE	2020-10-18	41.75	39.53	41.3
15	14	Aave	AAVE	2020-10-19	41.00	35.95	40.8
16	15	Aave	AAVE	2020-10-20	36.32	32.10	36.0
17	16	Aave	AAVE	2020-10-21	37.08	32.05	32.2
18	17	Aave	AAVE	2020-10-22	39.86	33.59	35.1

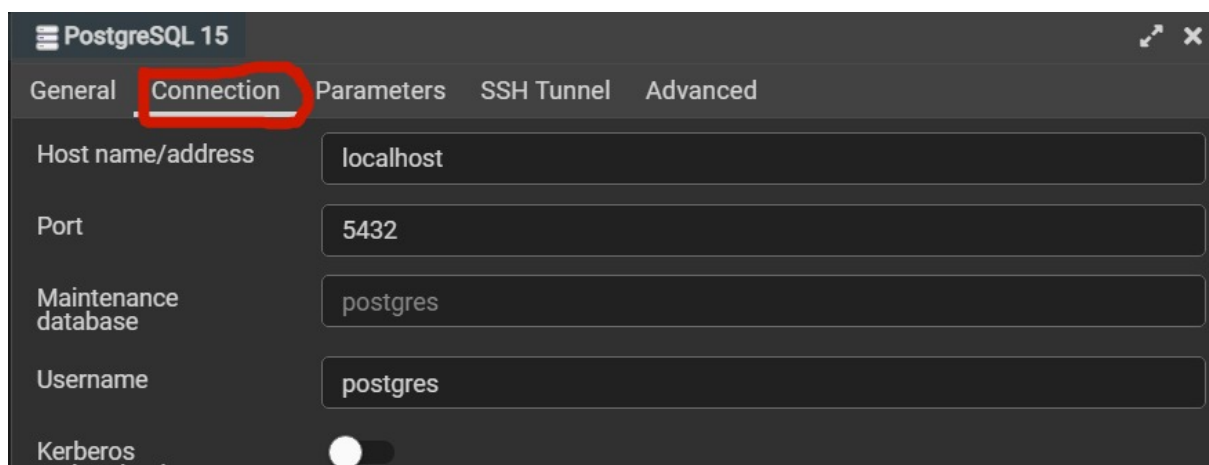
Total rows: 1000 of 14779 Query complete 00:00:00.291 Ln 1, Col 1

3. Integrando PostgreSQL com Ferramenta de visualização

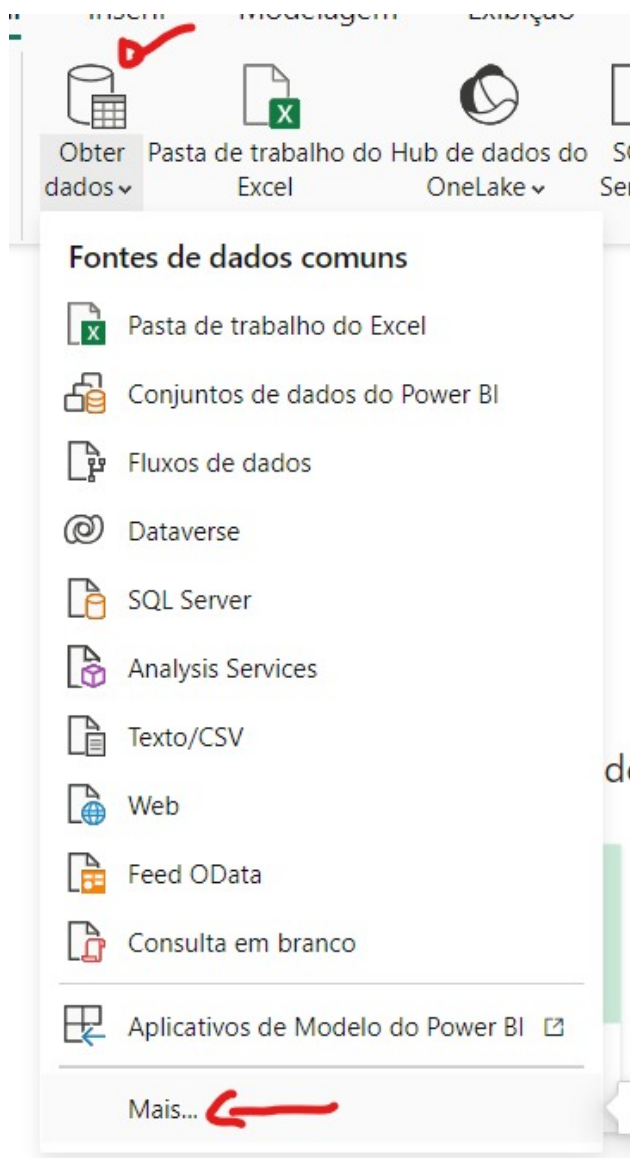
- Ao conectar a ferramenta pgAdmin com o Power BI é necessário as mesmas informações que foram preenchidas nas configurações do *script_postgre.py*, elas foram retiradas nas propriedades do PostgreSQL (*clique botão direito*)



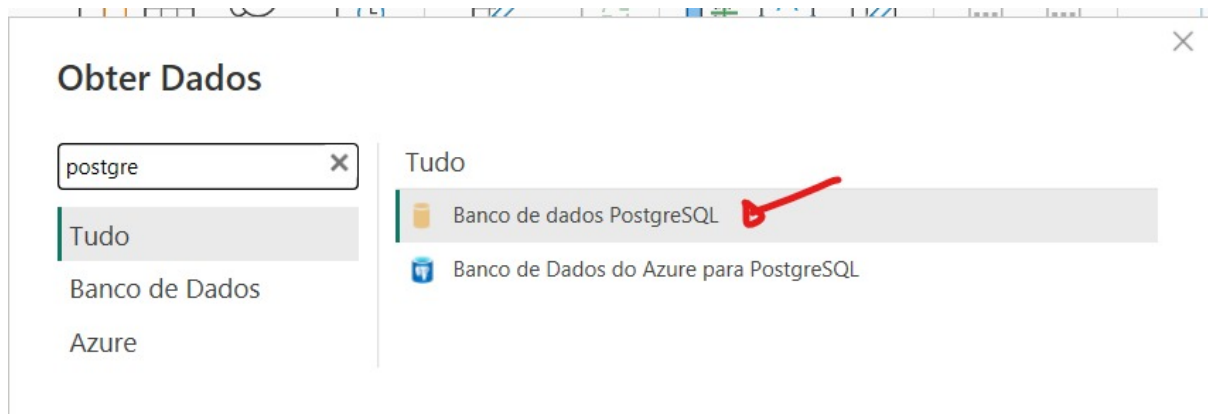
#



- Estas informações foram passadas para o **Power BI**



- Ao abrir a seção **Mais**



- Após a seleção

Banco de dados PostgreSQL

Servidor

localhost:5432

Banco de Dados

Cripto_base

Modo de Conectividade de Dados ⓘ

☒ Importar

☐ DirectQuery

▷ Opções avançadas

OK

Cancelar

Conectar

Banco de dados PostgreSQL

Banco de Dados

localhost;Cripto_base

Nome do usuário

postgres

Senha

•••••

Selecione o nível para aplicar essas configurações

localhost

Voltar

Conectar

Cancelar

#

Navegador

Opções de Exibição ▾

localhost:5432: Cripto_base [1]

☒ public.criptomoedas



public.criptomoedas

id	nome	simbolo	data	alto	baixo	aberto
0	Aave	AAVE	05/10/2020	55,11	49,79	
1	Aave	AAVE	06/10/2020	53,4	40,73	
2	Aave	AAVE	07/10/2020	42,41	35,97	
3	Aave	AAVE	08/10/2020	44,9	36,7	
4	Aave	AAVE	09/10/2020	47,57	43,29	
5	Aave	AAVE	10/10/2020	51,41	46,7	
6	Aave	AAVE	11/10/2020	51,45	48,72	
7	Aave	AAVE	12/10/2020	54,42	48,75	
8	Aave	AAVE	13/10/2020	57,48	49,6	
9	Aave	AAVE	14/10/2020	57,85	49,63	
10	Aave	AAVE	15/10/2020	52,08	44,02	
11	Aave	AAVE	16/10/2020	43,91	39,14	
12	Aave	AAVE	17/10/2020	43,08	38,63	
13	Aave	AAVE	18/10/2020	41,75	39,53	
14	Aave	AAVE	19/10/2020	41	35,95	
15	Aave	AAVE	20/10/2020	36,32	32,1	
16	Aave	AAVE	21/10/2020	37,08	32,05	
17	Aave	AAVE	22/10/2020	39,86	33,59	
18	Aave	AAVE	23/10/2020	42,55	36,92	
19	Aave	AAVE	24/10/2020	42,44	38,73	
20	Aave	AAVE	25/10/2020	40,93	37,6	
21	Aave	AAVE	26/10/2020	39,16	35,07	
22	Aave	AAVE	27/10/2020	38,87	34,47	

Selecionar Tabelas Relacionadas



Carregar

Transformar Dados

Cancelar

Ao clicar em *Transformar Dados* é possível verificar que os valores correspondem ao formato

`= Fonte([Schema="public",Item="criptomoedas"])[Data]`

simbolo	data	1.2 alto	1.2 baixo
AAVE	05/10/2020	55,11	49,79
AAVE	06/10/2020	53,4	40,73

Carregando

Carregar

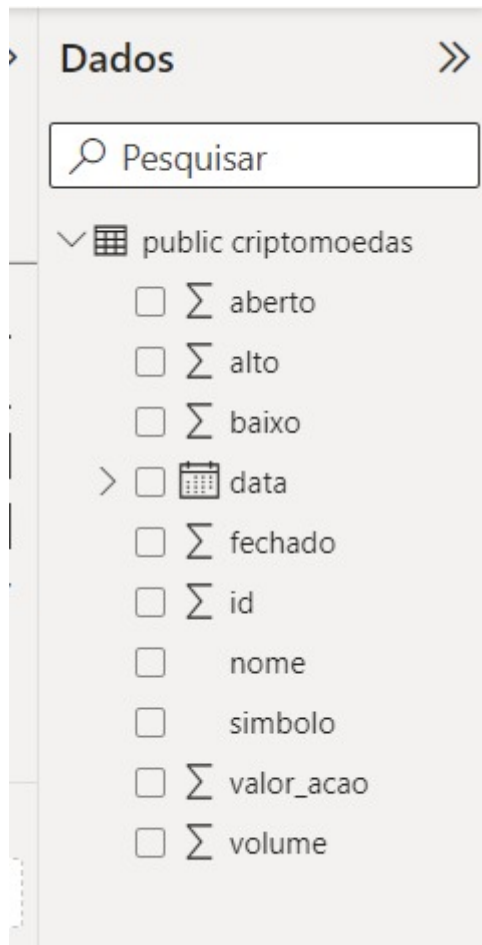


public.criptomoedas

Criando conexão no modelo...

Cancelar

Dados no Power BI



Link para o relatório: [Relatório](#)

Observações de uso:

Este documento tem cunho explicativo sobre o caminho para a execução dos arquivos para tratamento, exploração e criação do relatório dos Dados.

Para testar em sua máquina poderá se basear nos mesmos passos aqui descrito com as imagens. Segue um resumo:

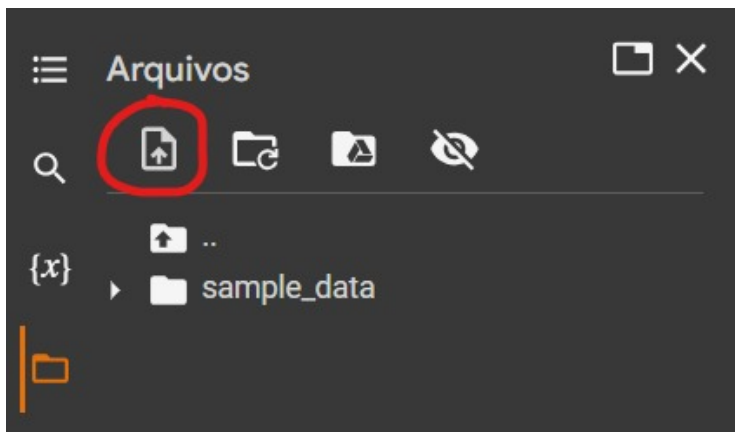
- **Acesse o repositório:** [Clique aqui](#);
- **Clone o repositório em sua máquina, junto ao código, os arquivos individuais de cada uma das dez moedas também serão transferidos para a sua máquina;**

- Abra o [Colab](#) ou se preferir, trate-o no VSCode, ele também será utilizado mais a frente. Caso opte pelo tratamento rápido pelo Colab, siga esta etapa:

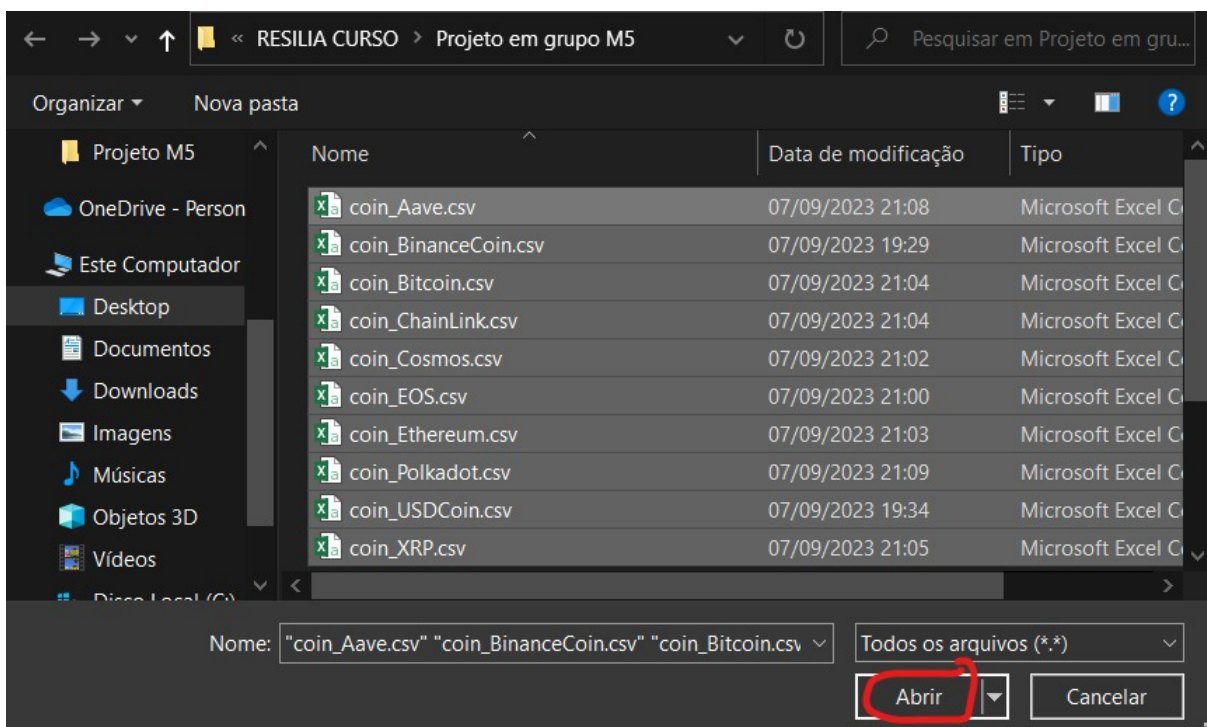
#1



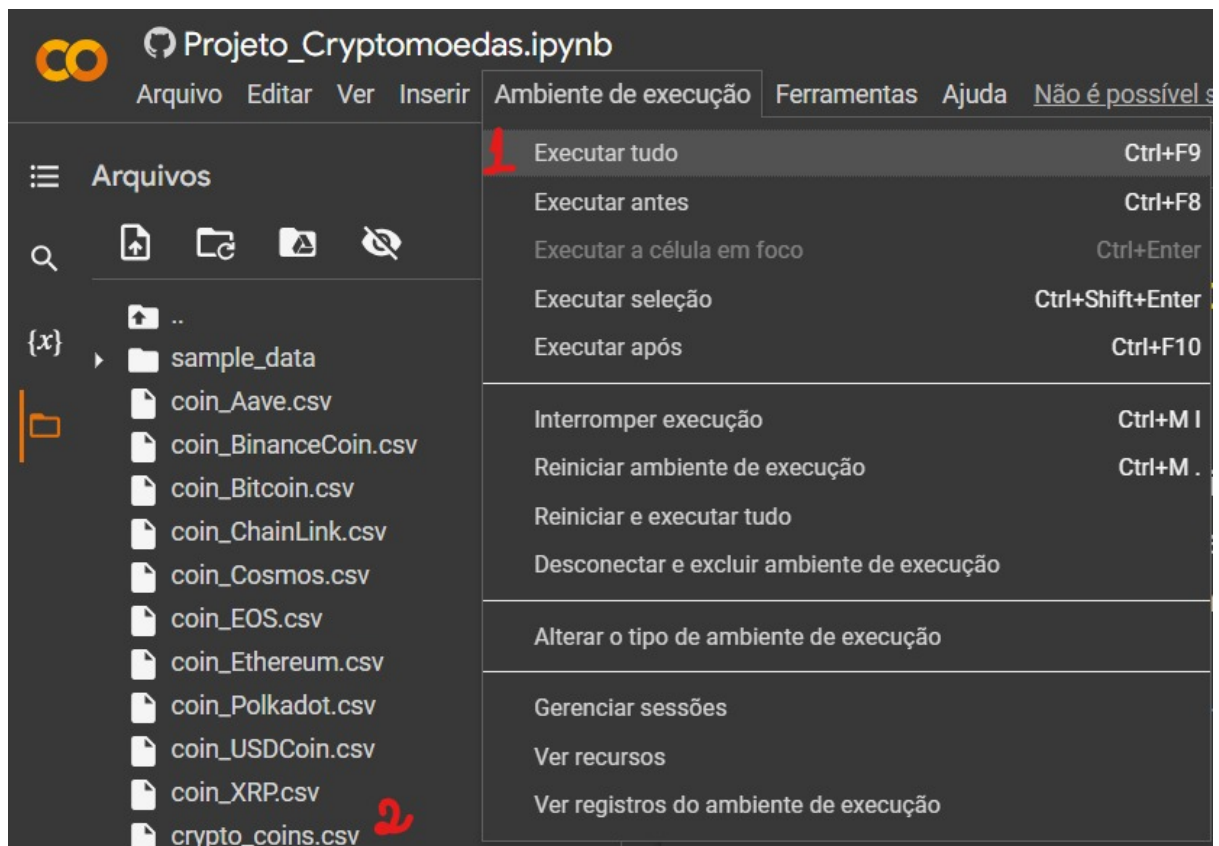
#2



#3 Localize a pasta em que se encontra os arquivos e faça o upload



#4 Após carregado, vá em *Ambiente de Execução* > *Executar tudo* e ele irá gerar o arquivo *crypto_coins.csv* que você irá utilizar no *PostgreSQL*;



#5



- Após fazer o download do arquivo, antes de executar o script .py, é necessário abrir o pgAdmin, criar o Banco de Dados para as criptos e logo após, a entidade onde serão inseridos os dados (*vide seção 2*).

- Após criar o BD, insira este comando para criar a tabela:

```
CREATE TABLE criptomoedas (
  Id SERIAL PRIMARY KEY,
  Nome VARCHAR(255),
  Simbolo VARCHAR(10),
  Data TIMESTAMP,
  Alto NUMERIC,
  Baixo NUMERIC,
  Aberto NUMERIC,
  Fechado NUMERIC,
  Volume BIGINT,
  Valor_acao NUMERIC
);
```

- Confira a tabela com `SELECT * FROM criptomoedas` para ter verificar as colunas criadas
- Agora, o script pode ser rodado para conectar ao Banco de Dados e inserir o arquivo.

>>> É recomendado que, ao rodar o `script_postgre.py`, o arquivo gerado no tratamento dos dados do Colab esteja dentro da mesma pasta que o arquivo do script, tanto no Windows quanto no Colab, para facilitar o acesso.

- Ao abrir o script, substitua o campo com as informações do seu banco de dados (*vide seção 2.1*):

```
#Configurar banco de dados postgresql
dbname = 'banco_de_dados'
user = 'seu_usuario'
password = 'sua_senha'
host = 'localhost'
port = '5432'
table_name = 'nome_tabela'

caminho_csv = 'crypto_coins.csv'
```

(O arquivo está na pasta [Scripts](#) > `script_postgre.py`)

- Após executar o `scrip_postgre.py`, os dados estarão em seu banco de dados

```
s/Python/Python311/python.exe "c:/Users/Nifrias/Desktop/Arq
Dados inseridos com sucesso!
PS C:\Users\Nifrias\Desktop\Arquivos Python\Projeto M5> □
```

- Volte ao *pgAdmin*, realize a consulta *SELECT * FROM nome_da_tabela* e verifique os valores inseridos
- Abra o *Power BI* > Obter Dados > Mais... > digite *postgre* na barra, clique na primeira opção > Preencha as informações do banco de dados que você criou, localhost, usuario e senha > selecione seu arquivo *public.seu_banco* e os dados estarão disponíveis para serem manipulados. (*vide seção 3*)