

Data Science – Princípios e Técnicas

Março
2025



Data Science



Onde me encontrar:

<https://www.linkedin.com/in/marco-mialaret-junior/>

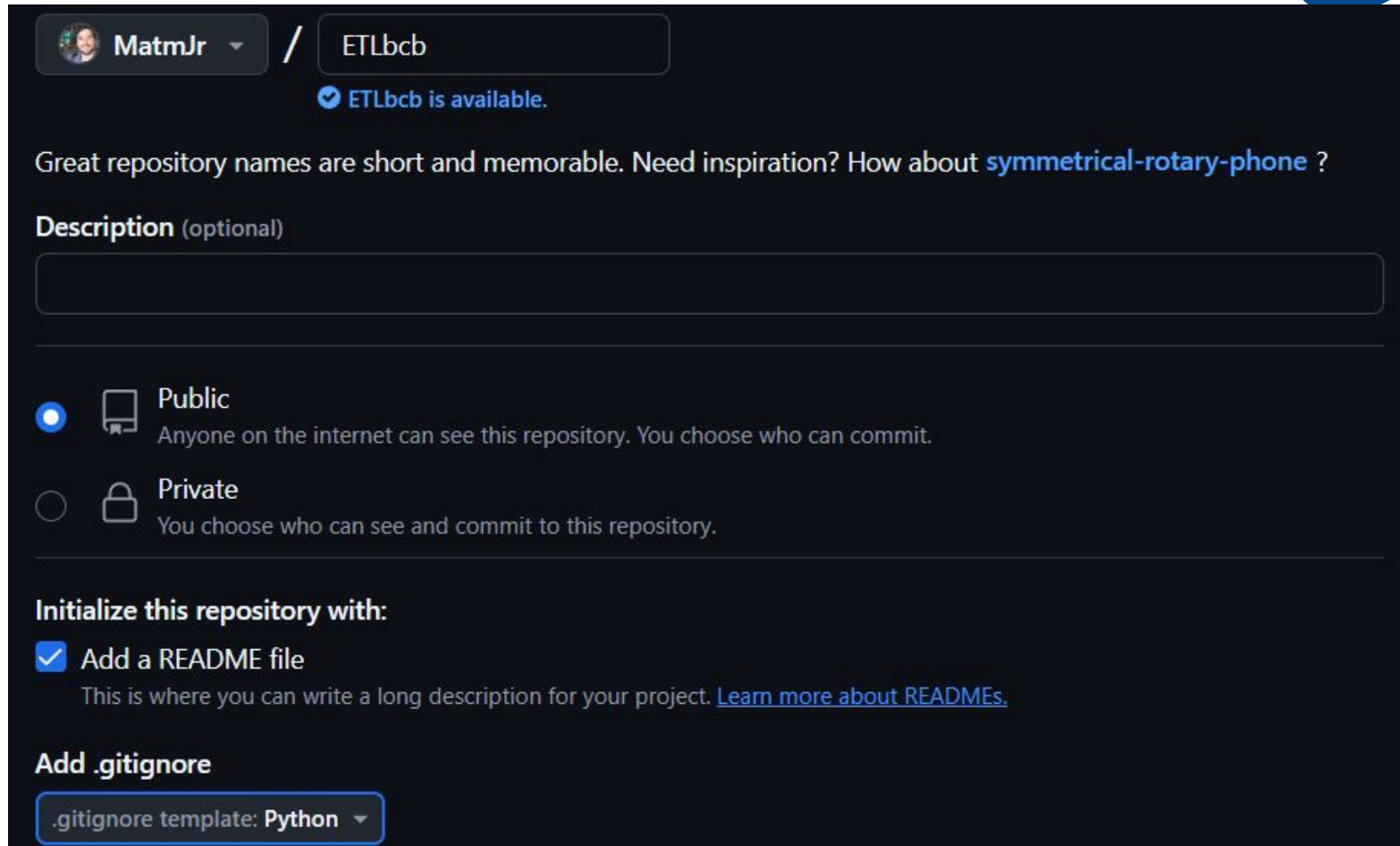
e

<https://github.com/MatmJr>

O repositório da última aula

Engenharia de Software

Na aula passada criamos um repositório.



The screenshot shows the GitHub repository creation page. At the top, the user 'MatmJr' is logged in, and the repository name 'ETLbcb' is entered. A message indicates 'ETLbcb is available.' Below this, a prompt suggests repository names should be short and memorable, with an example 'symmetrical-rotary-phone'. A 'Description (optional)' text box is provided. The 'Public' option is selected, with a note that anyone on the internet can see and commit to the repository. The 'Private' option is also visible. Under 'Initialize this repository with:', the 'Add a README file' checkbox is checked, with a link to learn more about READMEs. The 'Add .gitignore' section shows a dropdown menu with 'Python' selected.

MatmJr / ETLbcb

✔ ETLbcb is available.

Great repository names are short and memorable. Need inspiration? How about [symmetrical-rotary-phone](#) ?

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:

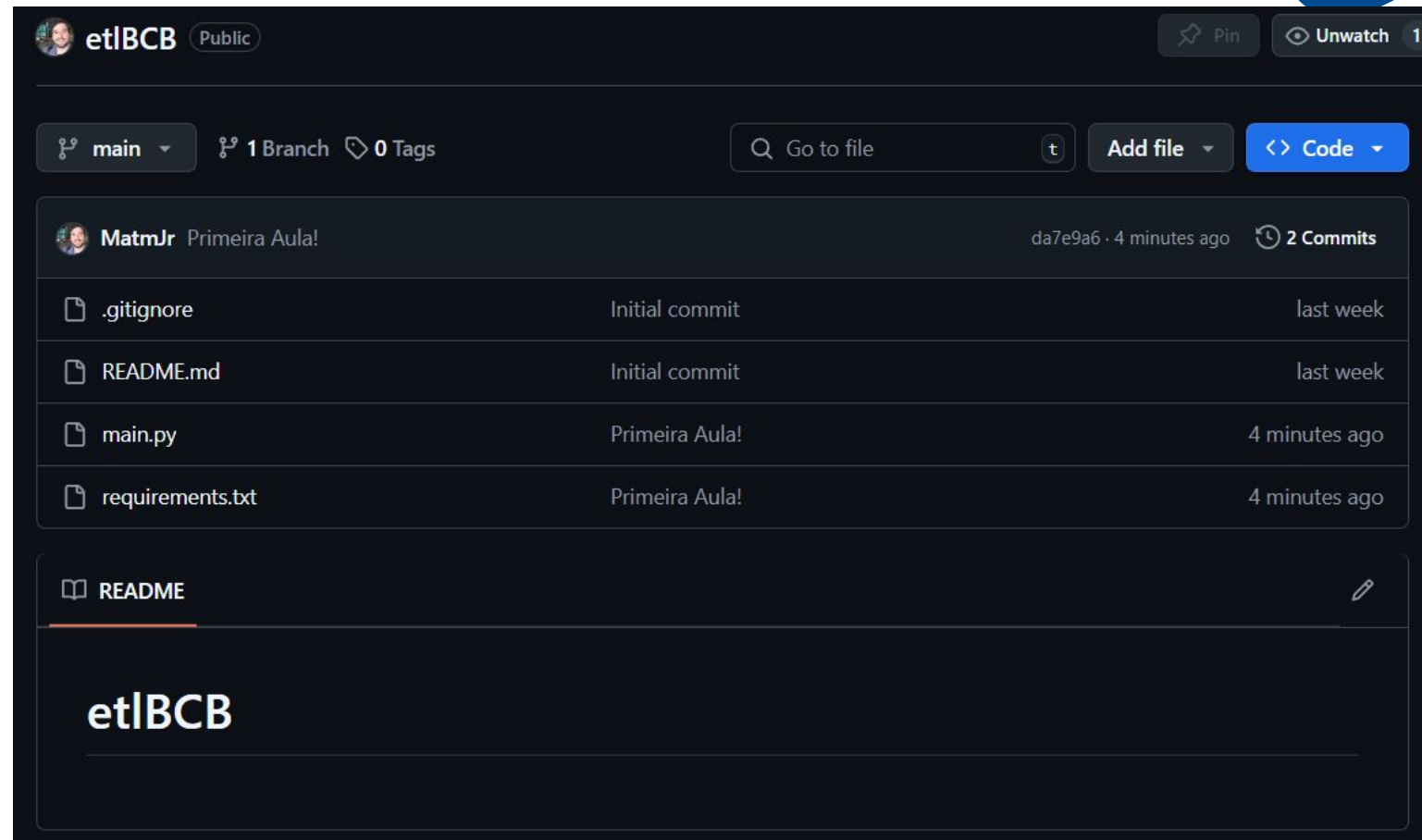
☒ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: Python

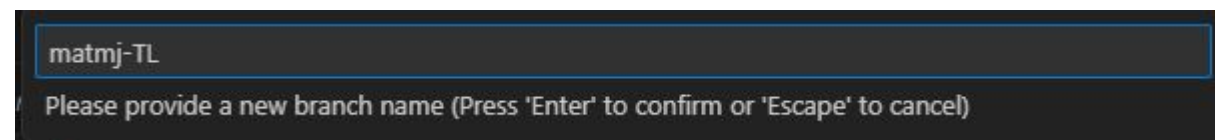
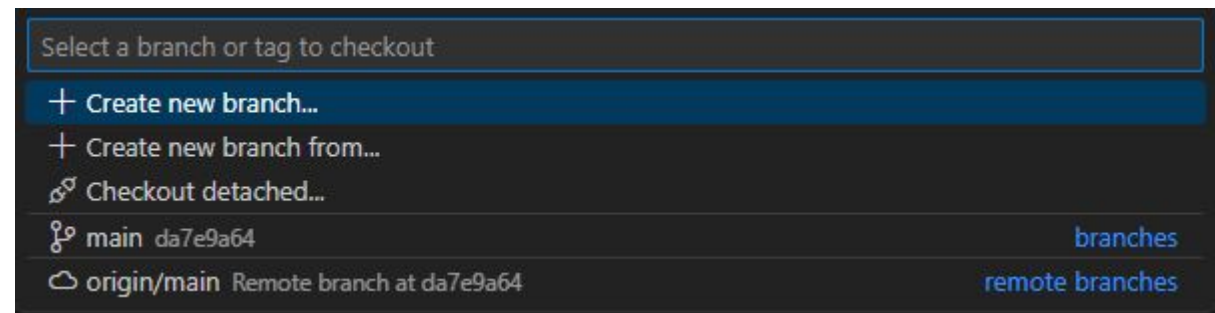
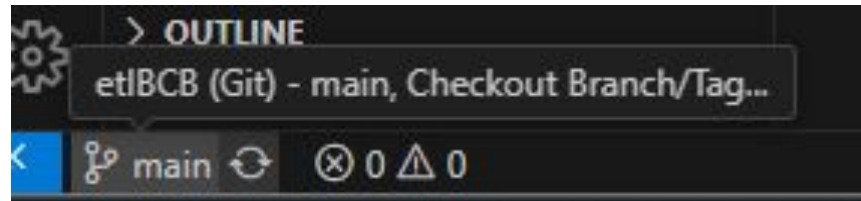
Engenharia de Software

Vamos
começar a
aula clonando
o repositório
construído na
aula passada.



Engenharia de Software

Crie uma nova branch no seu repositório.



Data Science

Na aula passada fizemos a extração dos dados, agora vamos fazer algumas transformações e depois vamos salvá-los para cobrir todas as etapas do ETL.

Transformação (Transform)

Data Science

Na aula passada criamos uma função que acessava a API do banco central, mas os dados estavam sem nenhum tipo de formatação. Primeiro vamos fazer algumas melhorias estéticas na nossa função.

Data Science

Código da aula passada:

```
def etlBcB(data):  
    """  
    Função para extrair os dados da API do Banco Central.  
    """  
    url = f"https://olinda.bcb.gov.br/olinda/servico/MPV_DadosAbertos/versa  
    req = requests.get(url)  
    print("Status Code:", req.status_code)  
    data = req.json()  
  
    df = pd.json_normalize(data["value"])  
    print(df)
```

Data Science



Começaremos “typando” a função e a sua DocString.

```
def etlBcB(date: str) -> pd.DataFrame:
    """
    Função para extrair os dados da API do Banco Central.

    Atributo:
    String - AAAAT - A ano e T trimestres (1-4)

    Saída:
    DataFrame com os dados econômicos dos meios de Pagamento.
    """
```

Data Science

Vamos retirar o print da função e inserir um retorno para ela.

```
url = f"https://olinda.bcb.gov.br/olinda/servico/MPV_DadosAbertos/versa
req = requests.get(url)
print("Status Code:", req.status_code)
data = req.json()

df = pd.json_normalize(data["value"])
return df
```

Data Science

Mas qual a vantagem de adicionar um retorno?

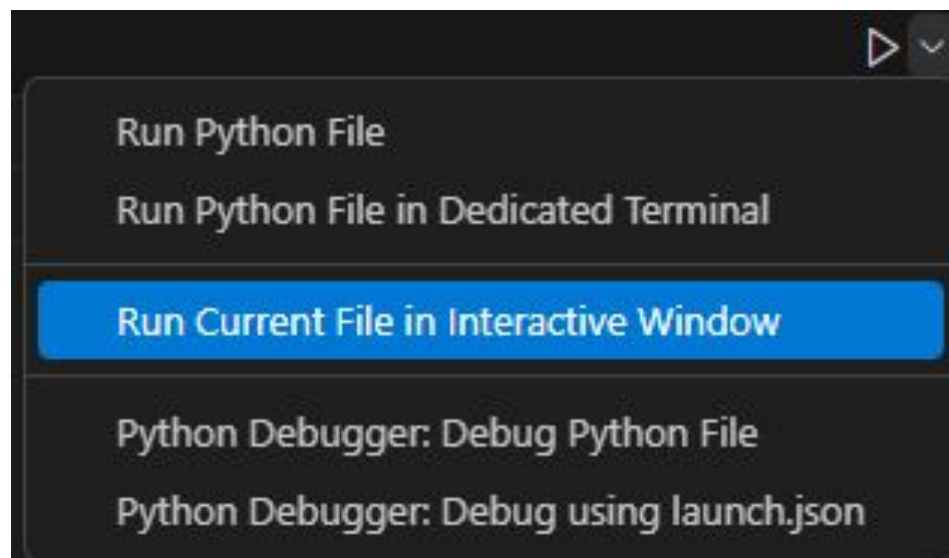
Data Science

```
df = etlBcB("20191")  
print(df)
```

Podemos atribuir a saída da função a uma variável.

Data Science

Perceba que todas as vezes que vamos executar a função precisamos acessar o link da API, para melhorar a exploração dos dados vamos usar a execução interativa do VSCode.



Data Science

Connected to .venv (Python 3.12.9)

✓ `import requests ...`

Status Code: 200

	datatrimestre	valorPix	valorTED	valorTEC	valorCheq
0	2024-09-30	6975429.47	11180217.68	0.00	194330.
1	2024-06-30	6284296.05	10662555.73	0.00	221553.
2	2024-03-31	5429305.43	9651625.56	925.26	190457.
3	2023-12-31	5300155.00	10090921.81	5787.89	221424.
4	2023-09-30	4469529.57	9960251.89	4156.38	218121.
5	2023-06-30	3900377.43	10657463.13	4770.74	228408.
6	2023-03-31	3447198.28	9919602.84	4535.17	240686.
7	2022-12-31	3342913.94	10382121.63	5609.53	242259.
8	2022-09-30	2936413.95	10437763.16	3735.46	267090.
9	2022-06-30	2543384.40	10449574.67	4786.14	271008.
10	2022-03-31	2067826.55	9507158.34	3865.81	255455.
11	2021-12-31	1916418.68	9807293.11	5436.19	264298.

Data Science

```
df.info()
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 23 entries, 0 to 22
```

```
Data columns (total 27 columns):
```

#	Column	Non-Null Count	Dtype
---	-----	-----	-----
0	datatrimestre	23 non-null	object
1	valorPix	23 non-null	float64
2	valorTED	23 non-null	float64
3	valorTEC	23 non-null	float64
4	valorCheque	23 non-null	float64
5	valorBoleto	23 non-null	float64
6	valorDOC	23 non-null	float64
7	valorCartaoCredito	23 non-null	float64

Data Science

Note que a coluna datatrimestre não está no formato de data e isso pode trazer problemas na hora de construir um banco de dados.

Data Science

No pandas temos um método para converter strings em datetime, então podemos converter os valores presentes no atributo 'datatrimestre'.

```
pd.to_datetime(df['datatrimestre'])
```

✓ 0.0s

0	2024-09-30
1	2024-06-30
2	2024-03-31
3	2023-12-31
4	2023-09-30
5	2023-06-30

Data Science

Mas os valores foram convertidos no dataframe?

Data Science

Não, para fazer isso precisamos atribuir a conversão ao atributo em questão

```
df['datatrimestre'] = pd.to_datetime(df['datatrimestre'])
```

✓ 0.0s

Data Science

```
df.info()
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 23 entries, 0 to 22
```

```
Data columns (total 27 columns):
```

#	Column	Non-Null Count	Dtype
0	datatrimestre	23 non-null	datetime64[
1	valorPix	23 non-null	float64
2	valorTED	23 non-null	float64

Data Science



Por fim, podemos adicionar essa operação a nossa função.

```
url = f"https://olinda.bcb.gov.br/olinda/servico/MPV_DadosA
req = requests.get(url)
print("Status Code:", req.status_code)
data = req.json()

df = pd.json_normalize(data["value"])

df['datatrimestre'] = pd.to_datetime(df['datatrimestre'])
return df
```

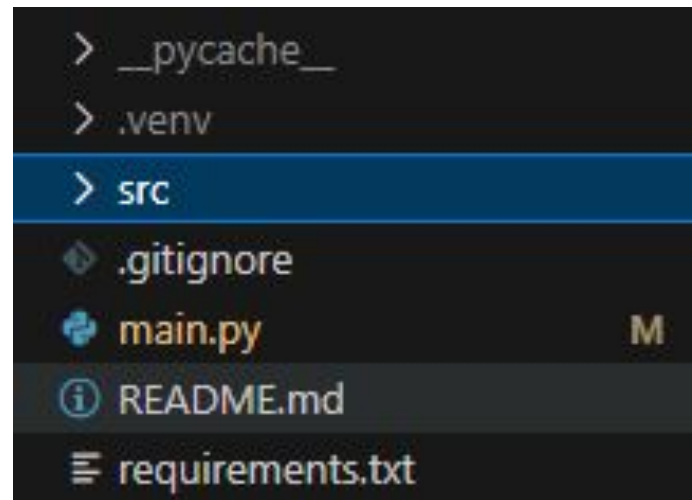
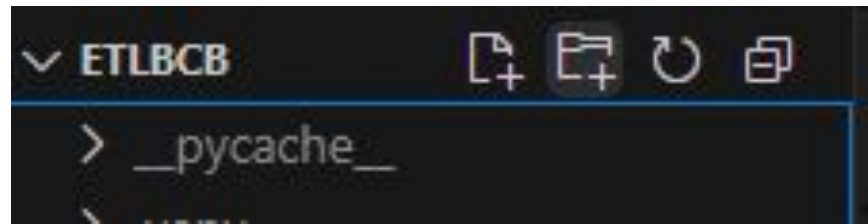
Carregamento (LOAD)

Engenharia de Software

Antes de criar métodos para salvar os dados, vamos gastar uns minutos melhorando a estrutura do projeto.

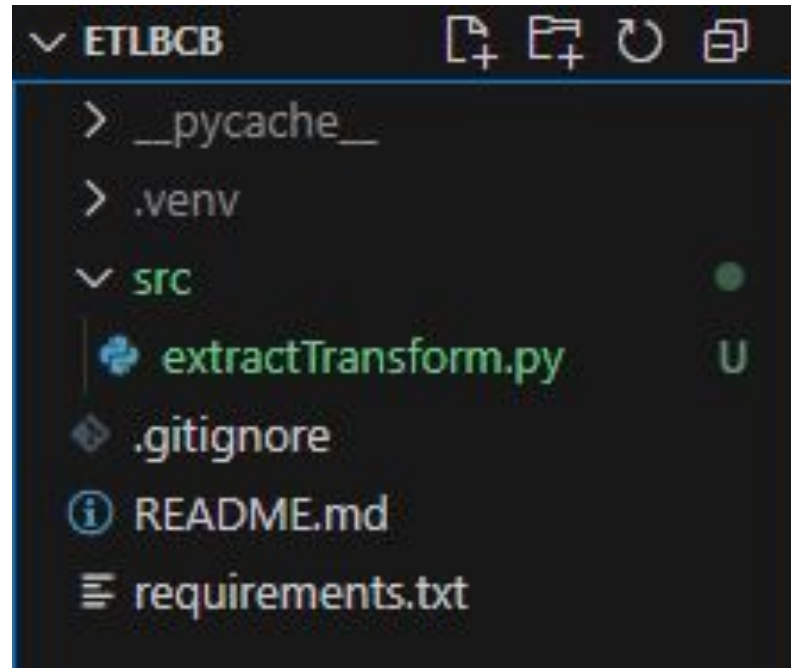
Data Science

Crie um novo diretório chamado de src



Data Science

Troque o nome do arquivo main.py para extractTransform.py e o coloque na pasta src.



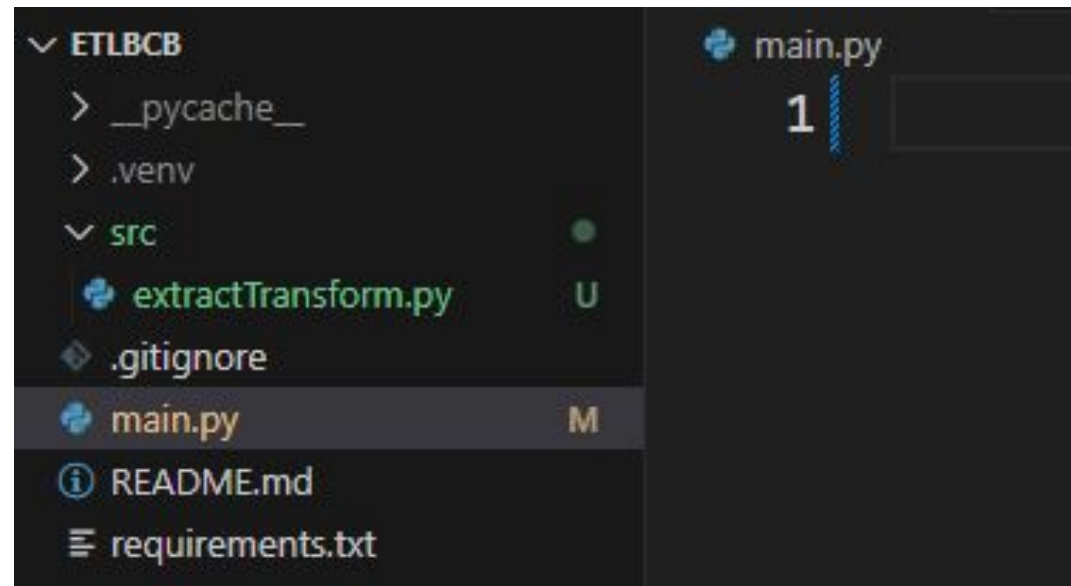
Data Science

Deixe apenas a função no arquivo extractTransform.py

```
extractTransform.py U X
src > extractTransform.py > ...
1 import requests
2 import pandas as pd
3
4 def etlBcB(date: str) -> pd.DataFrame:
5     """
6     Função para extrair os dados da API do Banco Central.
7
8     Atributo:
9     AAAAT - String - A ano e T trimestres (1-4) apartir de
10
11     Saída:
12     DataFrame com os dados econômicos dos meios de Pagamento.
13     """
14     url = f"https://olinda.bcb.gov.br/olinda/servico/MPV_DadosAbertos/versao/v1/odata/MeiosdePagamentosT"
15     req = requests.get(url)
16     print("Status Code:", req.status_code)
17     data = req.json()
18
19     df = pd.json_normalize(data["value"])
20
21     df['datatrimestre'] = pd.to_datetime(df['datatrimestre'])
22     return df
```

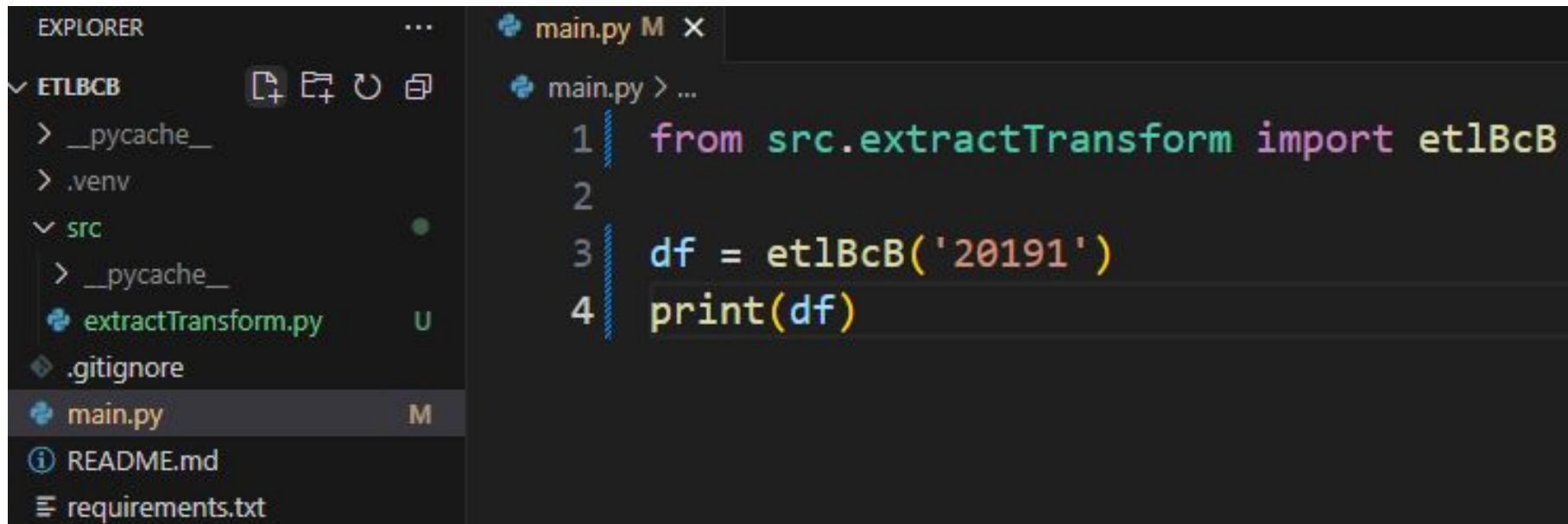
Data Science

Crie um novo arquivo chamado de main.py na raiz do projeto.



Data Science

O python permite que os arquivos sejam usados como módulo. Assim, podemos fazer o seguinte na nova main:

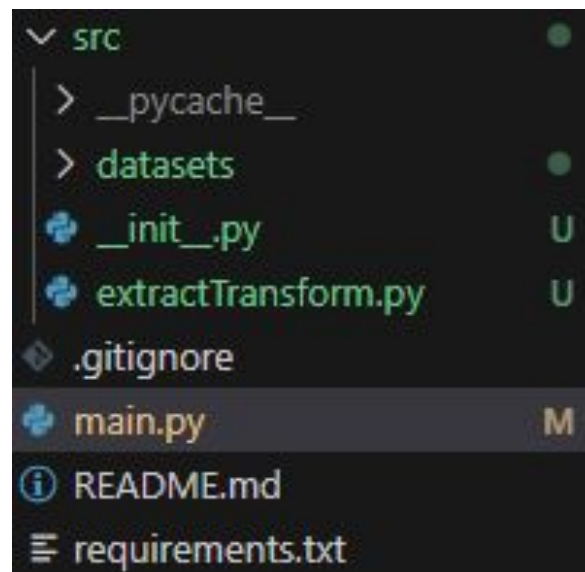


The screenshot shows a code editor interface. On the left, the 'EXPLORER' panel displays a file tree for a project named 'ETLBCB'. The tree includes a '__pycache__' directory, a '.venv' directory, a 'src' directory (which contains another '__pycache__' directory, 'extractTransform.py', and '.gitignore'), and files named 'main.py', 'README.md', and 'requirements.txt'. The 'main.py' file is selected. On the right, the editor window shows the content of 'main.py', which contains four lines of Python code:

```
1 from src.extractTransform import etlBcB
2
3 df = etlBcB('20191')
4 print(df)
```

Data Science

A primeira forma que vamos salvar os dados é convertendo em csv. Antes, vamos criar um diretório chamado datasets e um arquivo vazio chamado **init.py** no src.



```
▼ src
  > __pycache__
  > datasets
  + __init__.py
  + extractTransform.py
  .gitignore
  main.py
  README.md
  requirements.txt
```

The screenshot shows a file explorer interface with a dark background. The 'src' directory is expanded, showing its contents. The 'datasets' directory is highlighted in green. The 'main.py' file is highlighted in orange and marked with a yellow 'M' icon, indicating it has been modified. Other files include '__init__.py', 'extractTransform.py', '.gitignore', 'README.md', and 'requirements.txt'.

Data Science

o método que converte o dataframe em csv não vai ficar na main, vamos criar um arquivo chamado load.py no src e colocar o método como uma função.

```
import pandas as pd

def saveCsv(df: pd.DataFrame, nome_arquivo: str, separador: str, decimal: str):
    df.to_csv(f'src/datasets/{nome_arquivo}.csv', decimal=decimal, sep=separador)
```


Data Science

Devemos importar o load na main e executar o método.

```
main.py M x
main.py > ...
1 from src.extractTransform import etlBcB
2 from src.load import saveCsv
3
4 df = etlBcB('20191')
5
6 saveCsv(df, "meiosPagamentosTri", ";", ".")
7
```

Data Science

Devemos importar o load na main e executar o método.

```
main.py M x
main.py > ...
1 from src.extractTransform import etlBcB
2 from src.load import saveCsv
3
4 df = etlBcB('20191')
5
6 saveCsv(df, "meiosPagamentosTri", ";", ".")
7
```

Melhorando a documentação

Data Science

Vamos melhorar o README com as informações do nosso projeto. Não esqueça de pegar as informações dos atributos na documentação da API.

<https://dadosabertos.bcb.gov.br/dataset/estatisticas-meios-pagamentos>

Subindo para o GitHub

Data Science

Para as pessoas que estiverem trabalhando no computador pessoal. Abra o terminal e use os comandos:

```
git add .
```

```
PS E:\Python\etlBCB> git commit -m "Segunda Aula"
[matmj-TL 5aca8af] Segunda Aula
5 files changed, 55 insertions(+), 13 deletions(-)
create mode 100644 src/__init__.py
create mode 100644 src/datasets/meiosPagamentosTri.csv
create mode 100644 src/extractTransform.py
create mode 100644 src/load.py
```

```
git push
```

Data Science



Vai surgir um erro, pois criamos a branch localmente:

```
fatal: The current branch matmj-TL has no upstream branch.  
To push the current branch and set the remote as upstream, use
```

```
git push --set-upstream origin matmj-TL
```

```
To have this happen automatically for branches without a tracking  
upstream, see 'push.autoSetupRemote' in 'git help config'.
```

Data Science

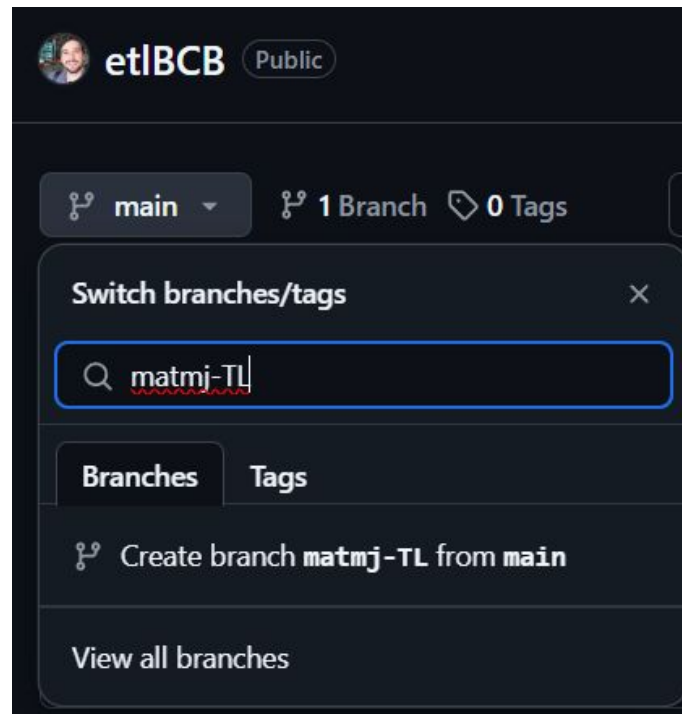


Mas, basta executar o código indicado no erro

```
git push --set-upstream origin matmj-TL
```

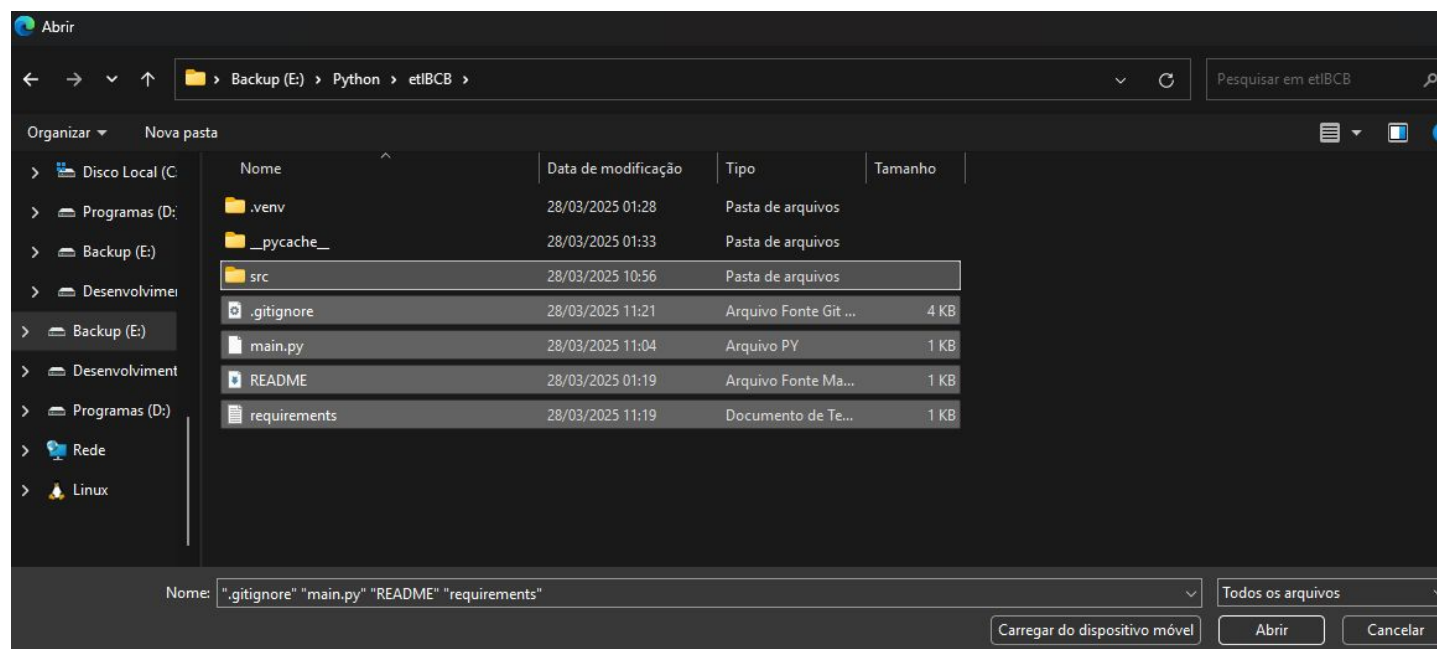

Data Science

Para as pessoas que estiverem trabalhando no computador da faculdade. Abra o github pelo navegador, acesse o repositório da aula e crie uma nova branch:



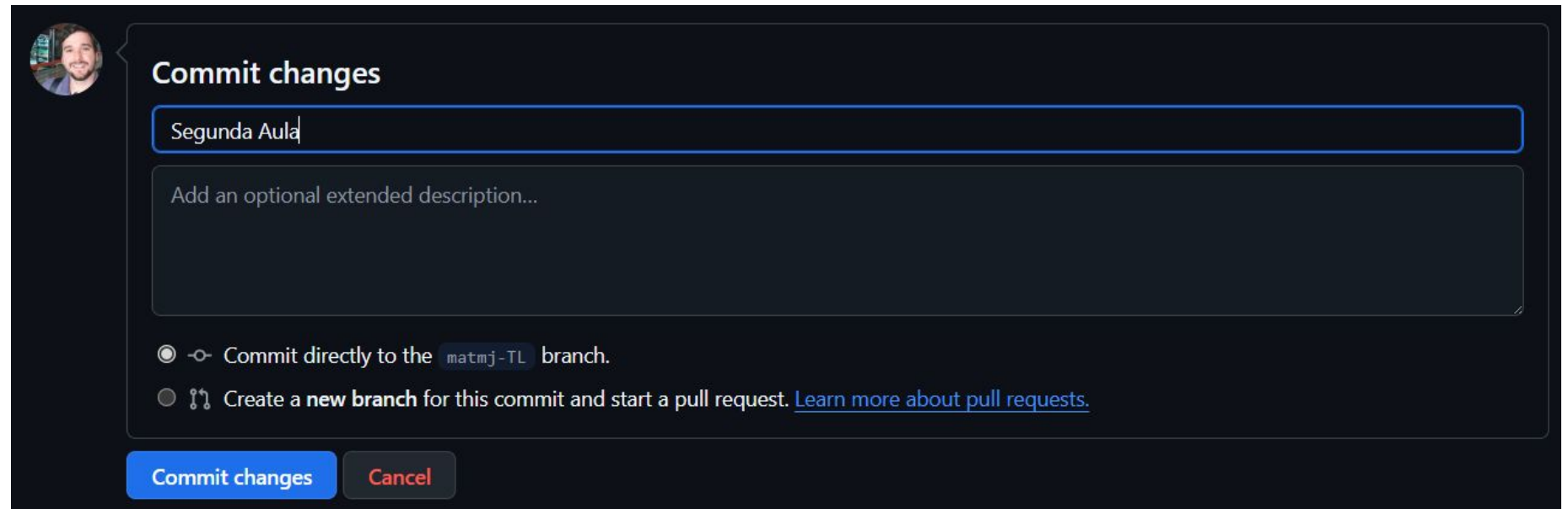
Data Science

Ao lado do botão code tem um “+”, clique nele e escolha a opção Upload files, lembre de não selecionar o .venv:




Data Science

Faço o commit



A screenshot of a GitHub commit dialog box. The dialog has a dark background. At the top left is a circular profile picture of a man. To its right is the title "Commit changes". Below the title is a text input field containing "Segunda Aula". Underneath the input field is a larger text area with the placeholder text "Add an optional extended description...". At the bottom of the dialog, there are two radio button options. The first option is selected and is labeled "Commit directly to the `matmj-TL` branch." The second option is labeled "Create a **new branch** for this commit and start a pull request. [Learn more about pull requests.](#)". At the very bottom of the dialog are two buttons: "Commit changes" in blue and "Cancel" in red.

 **Commit changes**

Segunda Aula

Add an optional extended description...

☒ Commit directly to the `matmj-TL` branch.

☐ Create a **new branch** for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes **Cancel**

Dúvidas?



Marco Mialaret, MSc

Telefone:

81 98160 7018

E-mail:

marcomialaret@gmail.com

