

## Data Science – Princípios e Técnicas

Abril  
2025



# Data Science



Onde me encontrar:

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

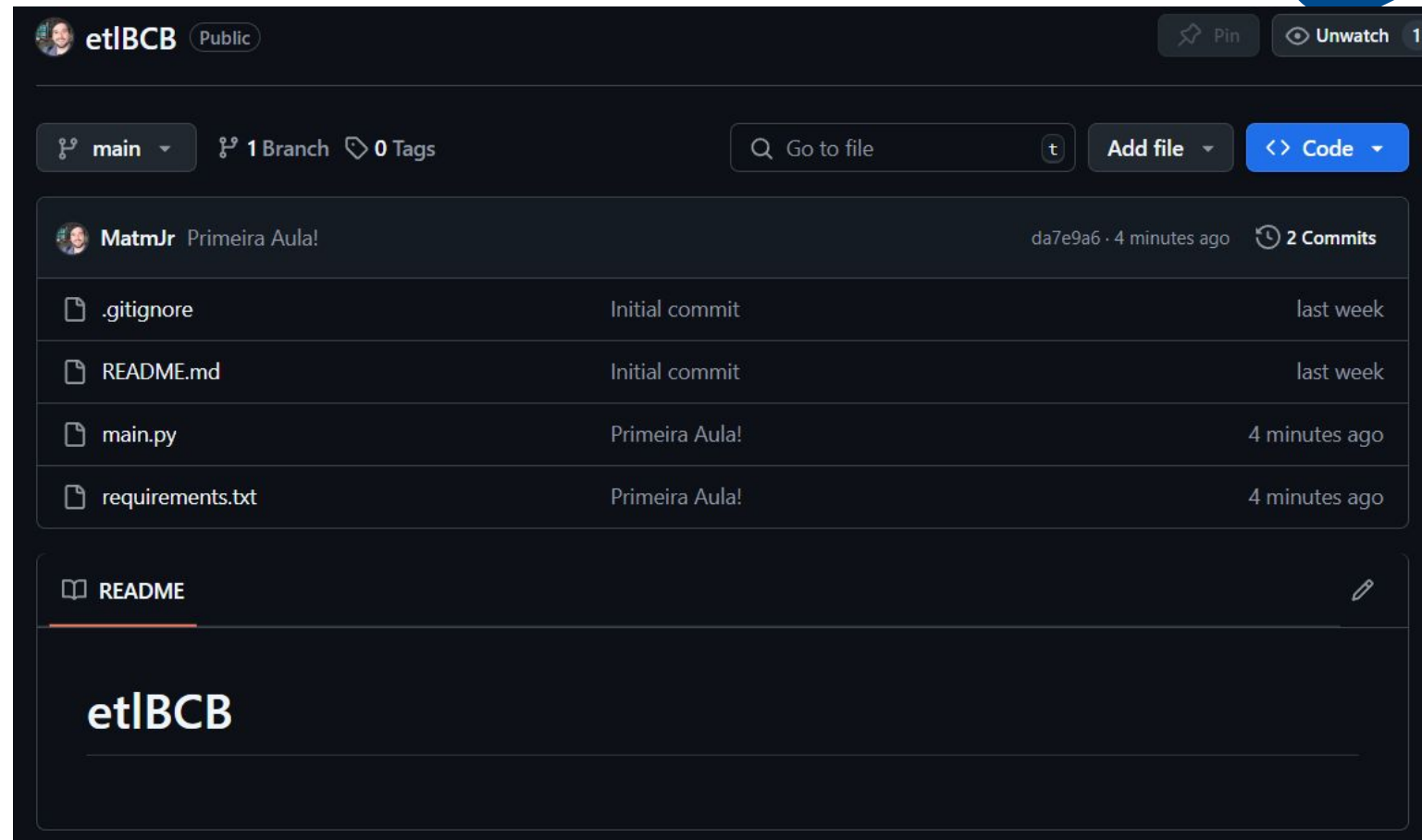
e

<https://github.com/MatmJr>

# O repositório da última aula

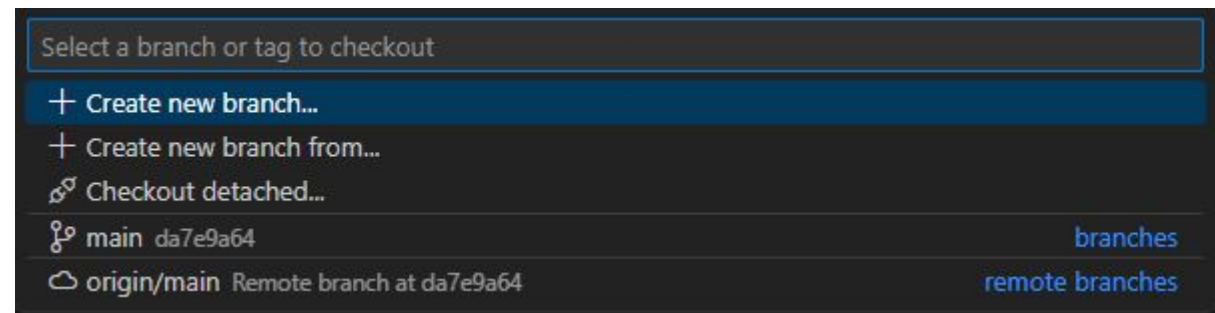
# Engenharia de Software

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.

# O SQLite

# Data Science

---

O SQLite é um banco de dados relacional que armazena informações em arquivos (geralmente em formato de tabela) sem depender de um servidor, o que o torna autônomo e oferece maior flexibilidade para o desenvolvimento de ferramentas e otimização dos processos empresariais.



# Data Science

---

O MySQL é popular por oferecer armazenamento em servidor e maior segurança com várias camadas de autenticação, embora exija mais configuração. Já o SQLite é mais indicado para aplicações móveis e se destaca pela simplicidade de uso, facilitando o manuseio e reduzindo riscos por erro humano.

# Data Science

---

Uma das vantagens do SQLite é sua ampla compatibilidade com sistemas operacionais como Windows, Linux, macOS, iOS e Android, facilitando o desenvolvimento de aplicações multiplataforma sem perda de performance. Além disso, ele pode ser usado com diversas linguagens de programação, desde que haja integração com uma biblioteca em linguagem C, padrão do SQLite.

# Data Science

---

O SQLite se destaca por sua independência e simplicidade, dispensando servidores e ferramentas auxiliares, o que facilita sua implementação, reduz custos e torna o sistema mais leve e ágil. Ele pode ser acessado com uma simples conexão, permitindo o uso prático de tabelas, cópias e reorganizações sem comandos complexos. Por isso, é uma ótima opção para quem desenvolve sites leves e aplicações móveis.

# Salvando os dados em SQLite

# Data Science

---

O SQLite já vem no Python e também existe um método para salvar no formato adequado no pandas. Assim, a função para salvar. Implemente a função a seguir no arquivo chamado load que criamos na aula passada.

# Data Science

---

```
import sqlite3
```

```
def salvarSQLite(df: pd.DataFrame, nome_banco: str, nome_tabela: str):  
    conn = sqlite3.connect(nome_banco)  
  
    df.to_sql(nome_tabela, conn, if_exists='replace', index=False)  
  
    conn.close()  
    return
```

# Data Science

---

o load.py deve ficar assim:

```
import pandas as pd
import sqlite3

def salvarCsv(df: pd.DataFrame, nome_arquivo: str, separador: str, dec:str):
    df.to_csv(nome_arquivo, sep=separador, decimal=dec)
    return

def salvarSQLite(df: pd.DataFrame, nome_banco: str, nome_tabela: str):
    conn = sqlite3.connect(nome_banco)

    df.to_sql(nome_tabela, conn, if_exists='replace', index=False)

    conn.close()
    return
```

# Data Science

---

a main.py deve ficar assim:

```
import pandas as pd
from src.extractTransform import requestApiBcb
from src.load import salvarCsv, salvarSQLite

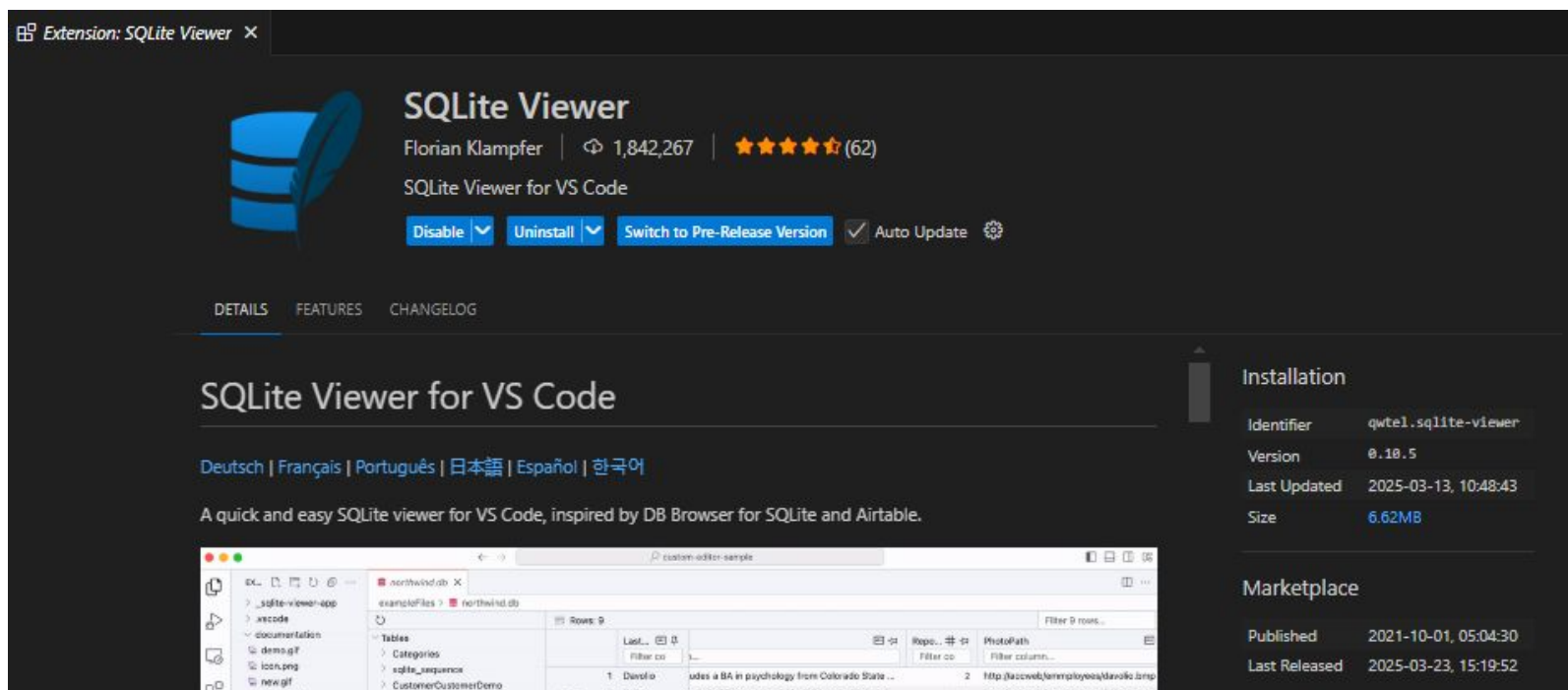
dadosBcb = requestApiBcb('20191')
salvarCsv(dadosBcb, "./src/datasets/meiosPagamentosTri.csv", ";",".")

salvarSQLite(dadosBcb, "src/datasets/dadosbcb.db", "meiosPagamentosTri")
```



# Data Science

Instalando o plugin SQLite viewer podemos visualizar os dados.



# Data Science

dadosbcb.db U X

src > datasets > dadosbcb.db

Filter 1 tables... Rows: 24 Filter 24 rows... Upgrade to PRO

TABLES		datatrimestre	valorPix	valorTED	valorTEC	valorCheq...	valorBoleto
> meiosPagamentosTri		Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
	1	2024-12-31 00:00:00	7714115.37	11647265.67	0	187486.45	2529891.48
	2	2024-09-30 00:00:00	6975429.47	11180217.68	0	194330.6	2444696.19
	3	2024-06-30 00:00:00	6284296.05	10662555.73	0	221553.82	2360557.98
	4	2024-03-31 00:00:00	5429305.43	9651625.56	925.26	190457.25	2248972.68
	5	2023-12-31 00:00:00	5300155	10090921.81	5787.89	221424.78	2317765.95
	6	2023-09-30 00:00:00	4469529.57	9960251.89	4156.38	218121.88	2296883.56
	7	2023-06-30 00:00:00	3900377.43	10657463.13	4770.74	228408.82	2249059.73
	8	2023-03-31 00:00:00	3447198.28	9919602.84	4535.17	240686.19	2211523.05
	9	2022-12-31 00:00:00	3342913.94	10382121.63	5609.53	242259.61	2259674.59
	10	2022-09-30 00:00:00	2936413.95	10437763.16	3735.46	267090.22	2248416.63
	11	2022-06-30 00:00:00	2543384.4	10449574.67	4786.14	271008.09	2210099.22
	12	2022-03-31 00:00:00	2067826.55	9507158.34	3865.81	255455.43	2032317.14
	13	2021-12-31 00:00:00	1916418.68	9807293.11	5436.19	264298.16	2072053.93
	14	2021-09-30 00:00:00	1556916.22	9257046.28	3684.13	271120.34	2001074
	15	2021-06-30 00:00:00	1105735.2	8566787.27	3894.81	264944.31	1839452.36
	16	2021-03-31 00:00:00	625046.52	7894414.16	4352.44	253527.92	1769759.68
	17	2020-12-31 00:00:00	149894.91	8036705.32	5137.42	278295.37	1919100.6
	18	2020-09-30 00:00:00	0	7503982.24	4159.6	259334.88	1959819.72
	19	2020-06-30 00:00:00	0	6267004.16	4647.32	237873.12	1366623.21
	20	2020-03-31 00:00:00	0	6652135.13	4829.56	318844.92	1526486.76
	21	2019-12-31 00:00:00	0	6560042.73	6343.77	372693.66	1579497.09
	22	2019-09-30 00:00:00	0	6285701.44	5469.66	385735.21	1484836.27
	23	2019-06-30 00:00:00	0	5700739.79	5506.27	388525.47	1409618.75
	24	2019-03-31 00:00:00	0	5039380.73	4794.82	375916.28	1468066.32

# Preparando o ambiente para salvar no Mysql

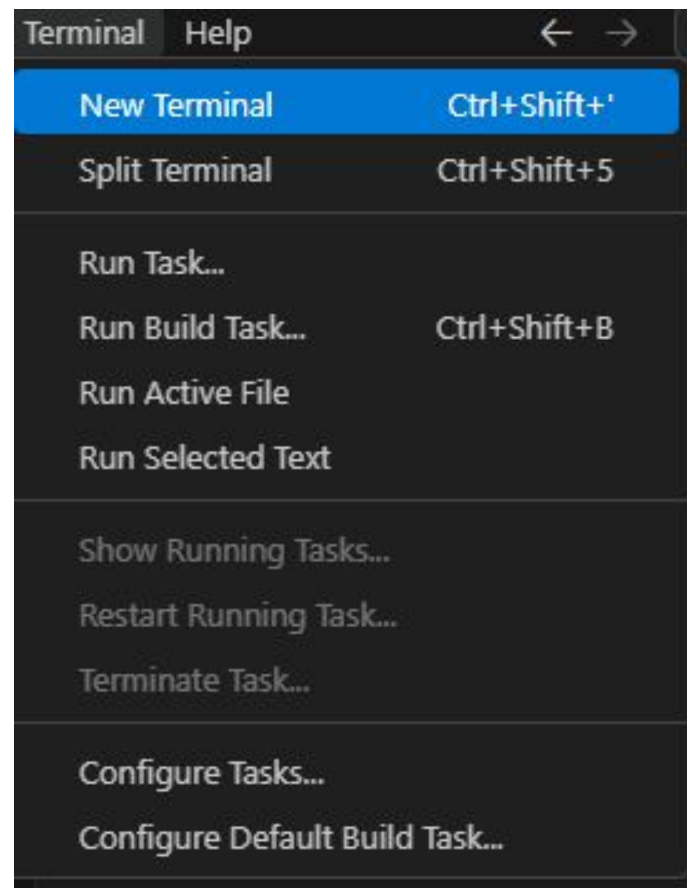
# Engenharia de Software

---

Você pode usar uma função semelhante para salvar um DataFrame do pandas diretamente em um banco MySQL, mas precisa usar o SQLAlchemy como "ponte" entre o pandas e o MySQL.

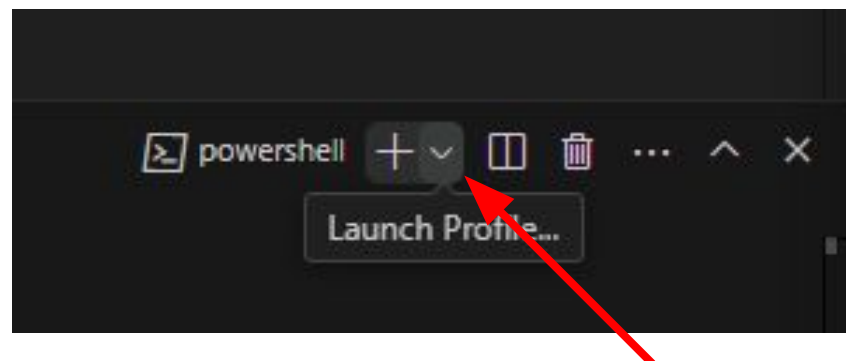
# Data Science

Abra o terminal:

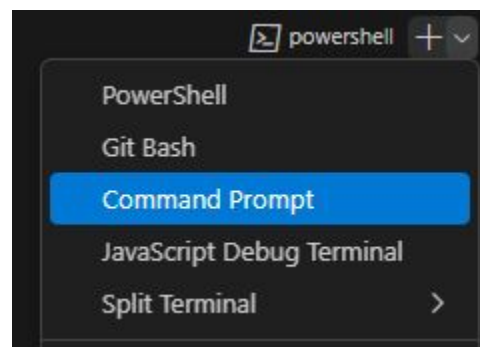


# Data Science

Clique na seta para baixo na parte superior direita do terminal



Selecione a opção Command Prompt



# Data Science



No terminal aberto execute o comando:

`.venv\Scripts\activate`

```
Microsoft Windows [versão 10.0.26100.2894]  
(c) Microsoft Corporation. Todos os direitos reservados.  
  
C:\Users\marco\OneDrive\Área de Trabalho\aula>.venv\Scripts\activate  
  
(.venv) C:\Users\marco\OneDrive\Área de Trabalho\aula>
```

# Data Science

---

No linux é um pouco diferente:

```
source .venv/bin/activate
```



# Data Science

---

Vamos instalar as primeiras bibliotecas (dependências)  
python:

```
pip install sqlalchemy pymysql
```

# Preparando o ambiente para salvar no Mysql

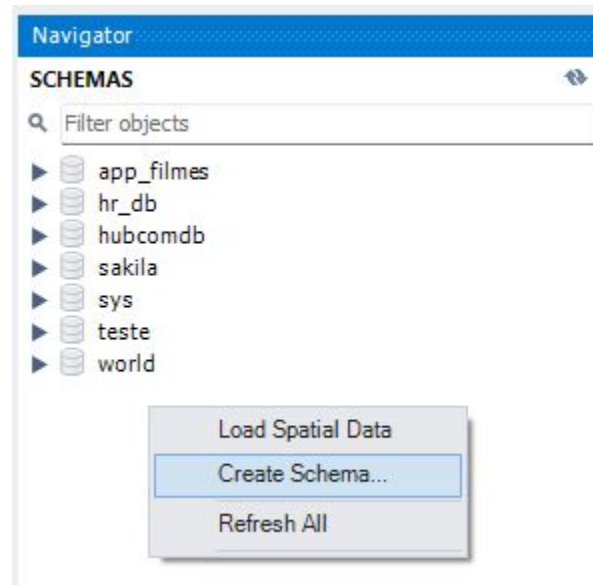
# Data Science

---

Antes de implementar a próxima função verifique se o seu computador possui o Workbench instalado. Caso esteja instalado crie um schema chamado etlbc.

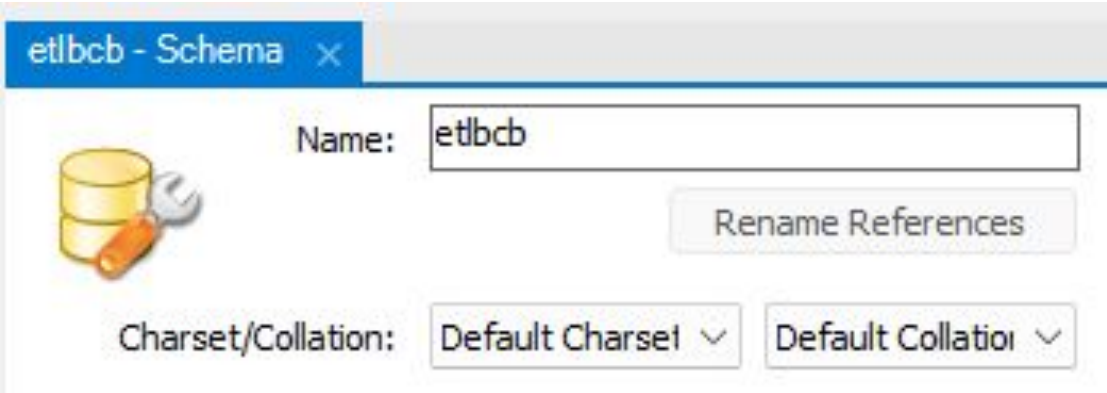
# Data Science

Clique com o botão do lado direito e escolha a opção create schema.




# Data Science

Clique com o botão do lado direito e escolha a opção create schema.



etlccb - Schema

Name:



Charset/Collation:

# Data Science



A implementação da função no load.py é tranquila

```
from sqlalchemy import create_engine
```

```
def salvarMySQL(  
    df: pd.DataFrame, usuario: str, senha: str, host: str, banco: str, nome_tabela: str  
    ):  
    engine = create_engine(f"mysql+pymysql://{usuario}:{senha}@{host}/{banco}")  
  
    df.to_sql(nome_tabela, con=engine, if_exists='replace', index=False)  
  
    return
```

# Data Science

```
import pandas as pd
import sqlite3
from sqlalchemy import create_engine

def salvarCsv(df: pd.DataFrame, nome_arquivo: str, separador: str, dec:str):

    df.to_csv(nome_arquivo, sep=separador, decimal=dec)

    return

def salvarSQLite(df: pd.DataFrame, nome_banco: str, nome_tabela: str):

    conn = sqlite3.connect(nome_banco)

    df.to_sql(nome_tabela, conn, if_exists='replace', index=False)

    conn.close()
    return

def salvarMySQL(
    df: pd.DataFrame, usuario: str, senha: str, host: str, banco: str, nome_tabela: str
):
    engine = create_engine(f"mysql+pymysql://{usuario}:{senha}@{host}/{banco}")

    df.to_sql(nome_tabela, con=engine, if_exists='replace', index=False)

    return
```

# Data Science

---

a main.py deve ficar assim:

```
import pandas as pd
from src.extractTransform import requestApiBcb
from src.load import salvarCsv, salvarSQLite, salvarMySQL

dadosBcb = requestApiBcb('20191')
salvarCsv(dadosBcb, "./src/datasets/meiosPagamentosTri.csv", ";", ".")

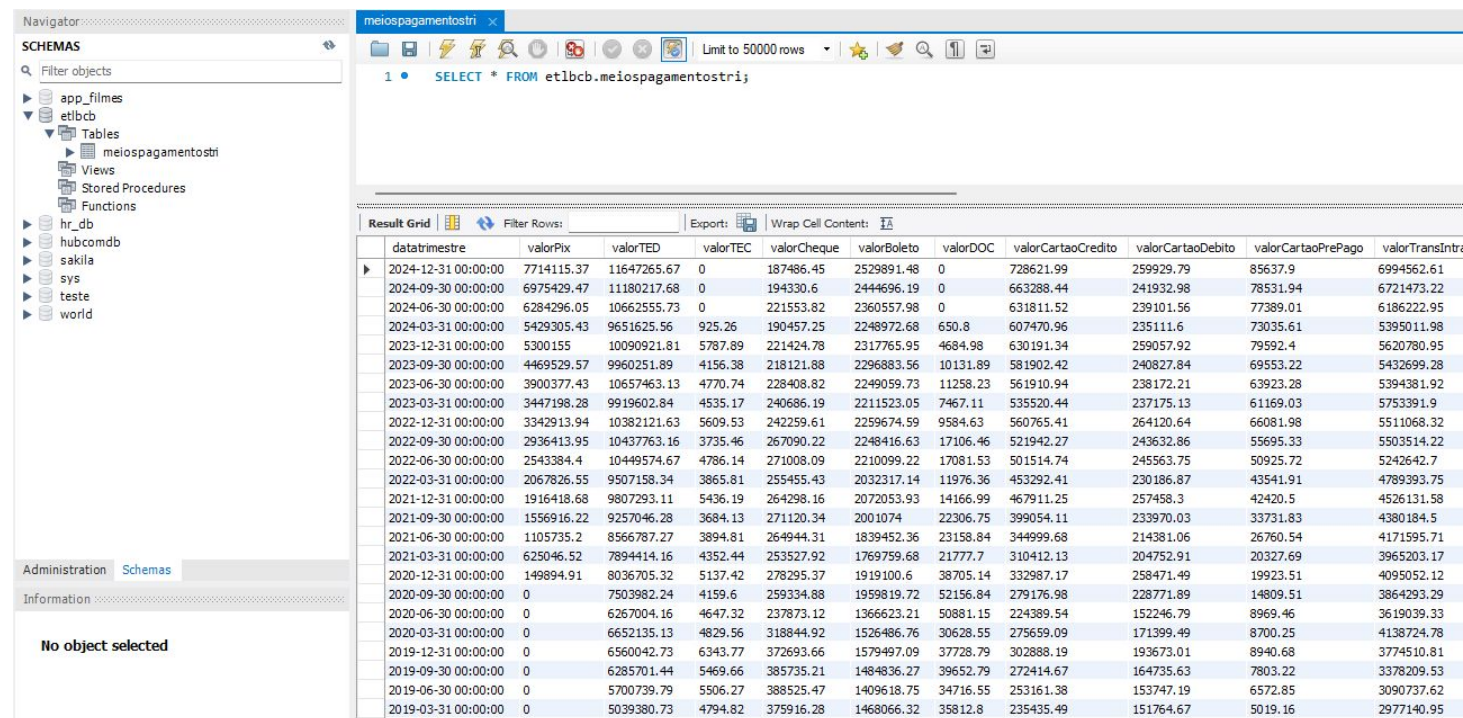
salvarSQLite(dadosBcb, "src/datasets/dadosbcb.db", "meiosPagamentosTri")

salvarMySQL(dadosBcb, "root", "root", "localhost", "etlbcb", "meiospagamentostri")
```



# Data Science

Ao executar o código será criada uma nova tabela no Workbench:



The screenshot displays the Oracle SQL Developer interface. On the left, the 'Navigator' pane shows the 'SCHEMAS' tree with 'meiospagamento' selected. The main window shows a SQL query: `SELECT * FROM etlbc.meiospagamento;`. Below the query, the 'Result Grid' shows a table with 11 columns: `datatimestre`, `valorPix`, `valorTED`, `valorTEC`, `valorCheque`, `valorBoleto`, `valorDOC`, `valorCartaoCredito`, `valorCartaoDebito`, `valorCartaoPrePago`, and `valorTransIntr`. The table contains 20 rows of data, representing transactions from 2019 to 2024.

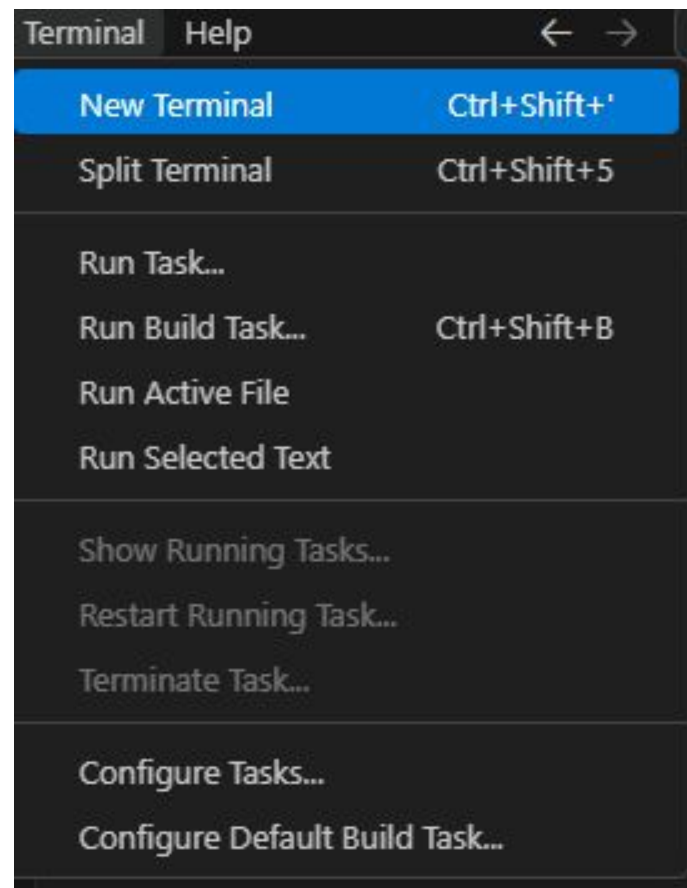
datatimestre	valorPix	valorTED	valorTEC	valorCheque	valorBoleto	valorDOC	valorCartaoCredito	valorCartaoDebito	valorCartaoPrePago	valorTransIntr
2024-12-31 00:00:00	7714115.37	11647265.67	0	187486.45	2529891.48	0	728621.99	259929.79	85637.9	6994562.61
2024-09-30 00:00:00	6975429.47	11180217.68	0	194330.6	2444696.19	0	663288.44	241932.98	78531.94	6721473.22
2024-06-30 00:00:00	6284296.05	10662555.73	0	221553.82	2360557.98	0	631811.52	239101.56	77389.01	6186222.95
2024-03-31 00:00:00	5429305.43	9651625.56	925.26	190457.25	2248972.68	650.8	607470.96	235111.6	73035.61	5395011.98
2023-12-31 00:00:00	5300155	10090921.81	5787.89	221424.78	2317765.95	4684.98	630191.34	259057.92	79592.4	5620780.95
2023-09-30 00:00:00	4469529.57	9960251.89	4156.38	218121.88	2296883.56	10131.89	581902.42	240827.84	69553.22	5432699.28
2023-06-30 00:00:00	3900377.43	10657463.13	4770.74	228408.82	2249059.73	11258.23	561910.94	238172.21	63923.28	5394381.92
2023-03-31 00:00:00	3447198.28	9919602.84	4535.17	240686.19	2211523.05	7467.11	535520.44	237175.13	61169.03	5753391.9
2022-12-31 00:00:00	3342913.94	10382121.63	5609.53	242259.61	2259674.59	9584.63	560765.41	264120.64	66081.98	5511068.32
2022-09-30 00:00:00	2936413.95	10437763.16	3735.46	267090.22	2248416.63	17106.46	521942.27	243632.86	55695.33	5503514.22
2022-06-30 00:00:00	2543384.4	10449574.67	4786.14	271008.09	2210099.22	17081.53	501514.74	245563.75	50925.72	5242642.7
2022-03-31 00:00:00	2067826.55	9507158.34	3865.81	255455.43	2032317.14	11976.36	453292.41	230186.87	43541.91	4789393.75
2021-12-31 00:00:00	1916418.68	9807293.11	5436.19	264298.16	2072053.93	14166.99	467911.25	257458.3	42420.5	4526131.58
2021-09-30 00:00:00	1556916.22	9257046.28	3684.13	271120.34	2001074	22306.75	399054.11	233970.03	33731.83	4380184.5
2021-06-30 00:00:00	1105735.2	8566787.27	3894.81	264944.31	1839452.36	23158.84	344999.68	214381.06	26760.54	4171595.71
2021-03-31 00:00:00	625046.52	7894414.16	4352.44	253527.92	1769759.68	21777.7	310412.13	204752.91	20327.69	3965203.17
2020-12-31 00:00:00	149894.91	8036705.32	5137.42	278295.37	1919100.6	38705.14	332987.17	258471.49	19923.51	4095052.12
2020-09-30 00:00:00	0	7503982.24	4159.6	259334.88	1959819.72	52156.84	279176.98	228771.89	14809.51	3864293.29
2020-06-30 00:00:00	0	6267004.16	4647.32	237873.12	1366623.21	50881.15	224389.54	152246.79	8969.46	3619039.33
2020-03-31 00:00:00	0	6652135.13	4829.56	318844.92	1526486.76	30628.55	275659.09	171399.49	8700.25	4138724.78
2019-12-31 00:00:00	0	6560042.73	6343.77	372693.66	1579497.09	37728.79	302888.19	193673.01	8940.68	3774510.81
2019-09-30 00:00:00	0	6285701.44	5469.66	385735.21	1484836.27	39652.79	272414.67	164735.63	7803.22	3378209.53
2019-06-30 00:00:00	0	5700739.79	5506.27	388525.47	1409618.75	34716.55	253161.38	153747.19	6572.85	3090737.62
2019-03-31 00:00:00	0	5039380.73	4794.82	375916.28	1468066.32	35812.8	235435.49	151764.67	5019.16	2977140.95

# Melhorando a estrutura do código

# Data Science

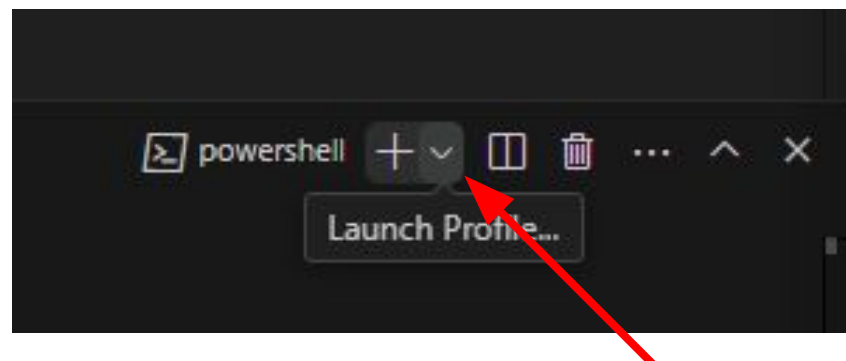
---

Abra o terminal:

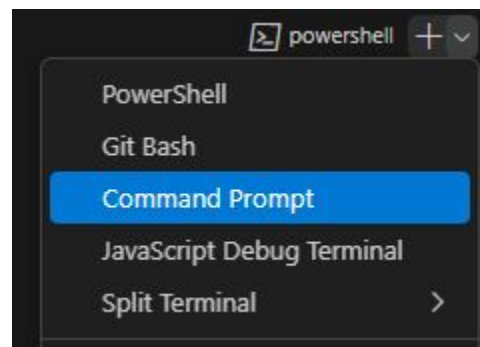


# Data Science

Clique na seta para baixo na parte superior direita do terminal



Selecione a opção Command Prompt



# Data Science



No terminal aberto execute o comando:

`.venv\Scripts\activate`

```
Microsoft Windows [versão 10.0.26100.2894]  
(c) Microsoft Corporation. Todos os direitos reservados.  
  
C:\Users\marco\OneDrive\Área de Trabalho\aula>.venv\Scripts\activate  
  
(.venv) C:\Users\marco\OneDrive\Área de Trabalho\aula>
```

# Data Science

---

No linux é um pouco diferente:

```
source .venv/bin/activate
```

# Data Science

---

Vamos instalar as primeiras bibliotecas (dependências)  
python:

```
pip install black
```

# Data Science

---

No terminal, verifique se você está na raiz do projeto e digite:  
`black .`



# Alterando o gitignore

# Data Science

---

A pasta datasets pode ficar muito grande, assim como o diretório .venv, sendo assim essas duas pastas devem ser ignoradas na hora de subir para o github:

# Data Science

---

Basta adicionar uma linha no github:

```
# Byte-compiled / optimized / DLL files
__pycache__/  
*.py[cod]  
*$py.class  
datasets
```

# 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

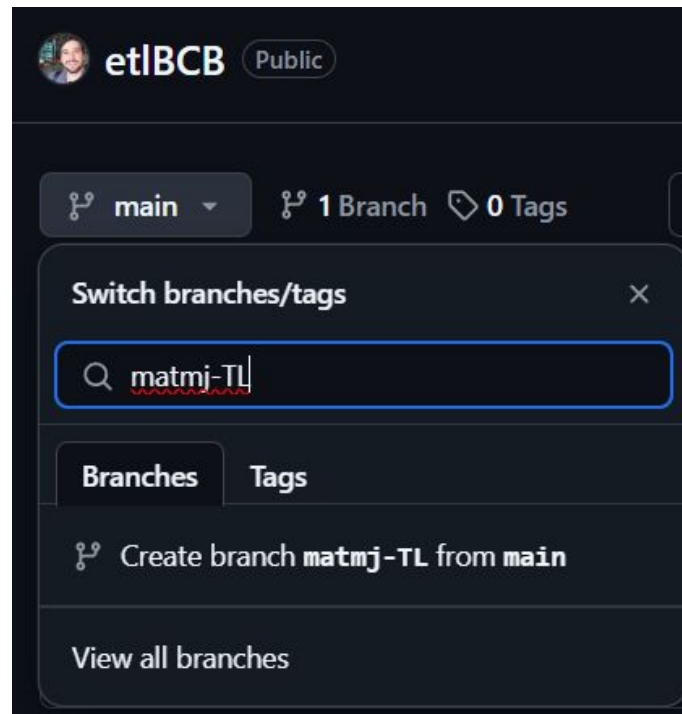


Mas, basta executar o código indicado no erro

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

# Data Science

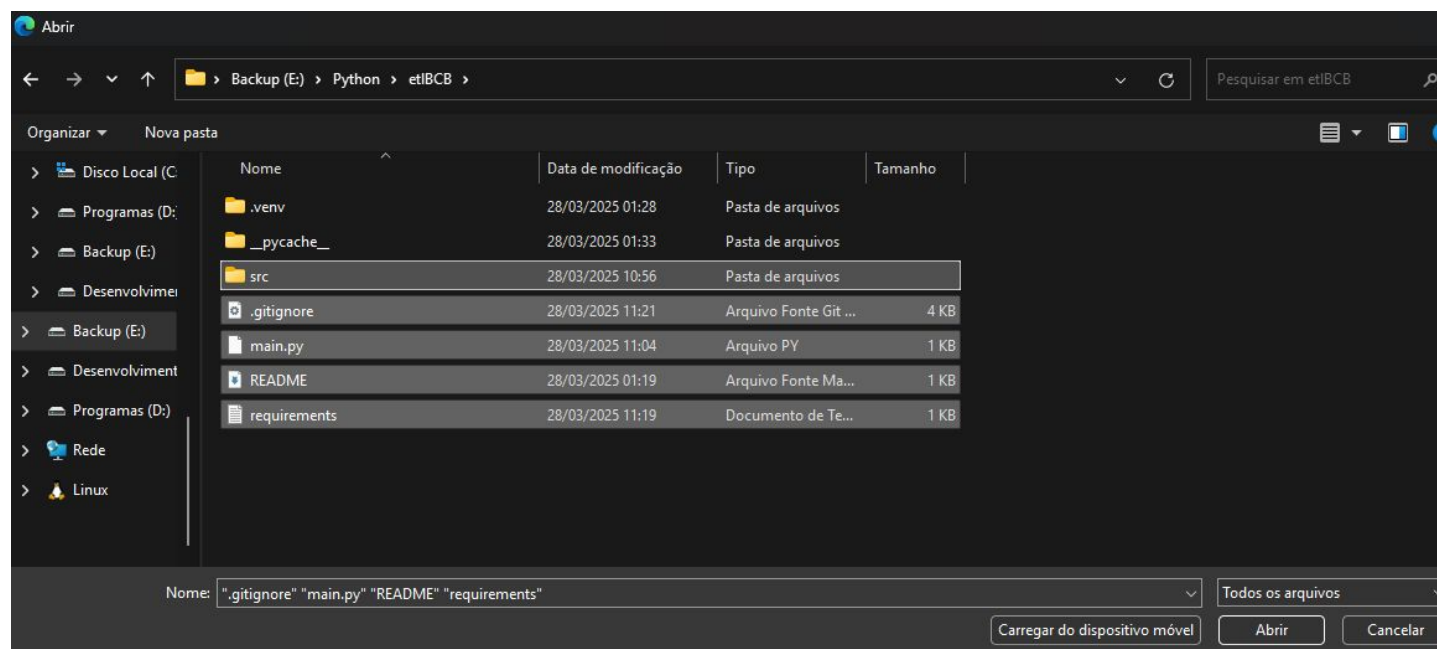
Para as pessoas que estiverem trabalhando no computador da faculdade. Abra o github pelo navegador, acesse o repositório da aula e crie uma nova branch (lembre de usar novos nomes):





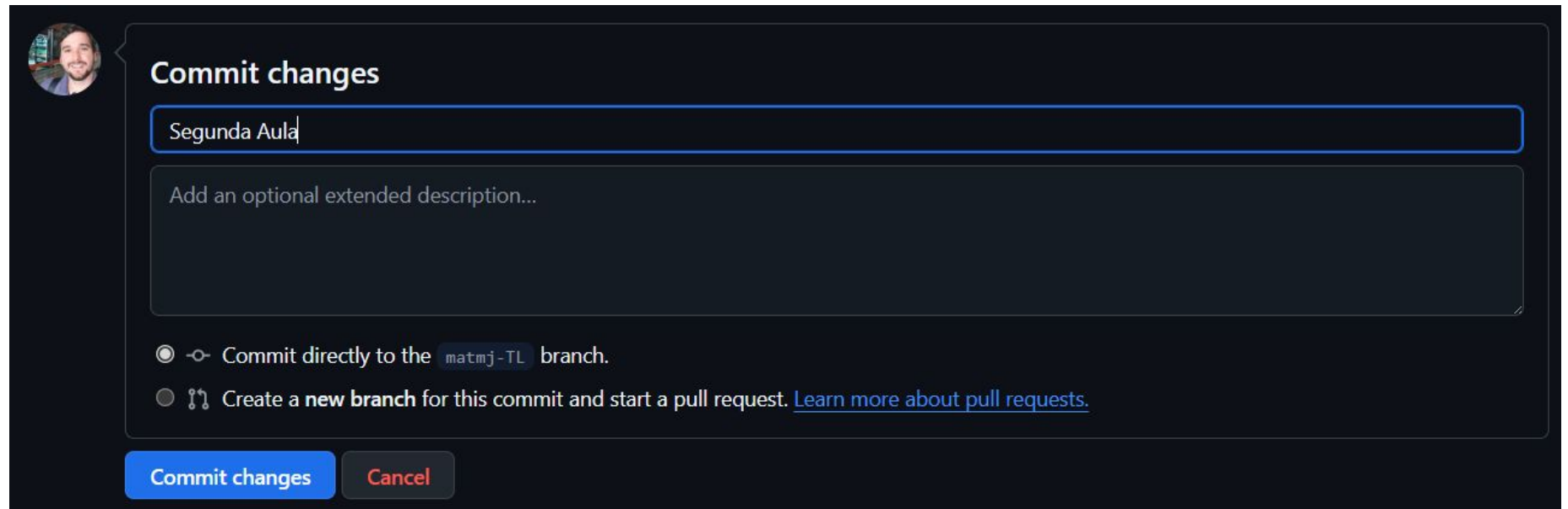
# Data Science

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 the text "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: a blue "Commit changes" button and a grey "Cancel" button.

 **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**

