



**Fecomércio
Sesc**

Big Data

Setembro
2025



O conjunto de Dados da aula passada

Big Data

Estamos usando uma API que contém domínios, nomes e países da maioria das universidades do mundo.

<https://github.com/Hipo/university-domains-list>

Big Data

```
import requests  
import sqlite3
```

```
url = "http://universities.hipolabs.com/search?country=Brazil"
```

```
# Acessando o link da internet  
response = requests.get(url)  
response.raise_for_status()  
universities = response.json()
```

Big Data

```
# Criar o banco e se conectar nele  
con = sqlite3.connect("universidades.db")  
c = con.cursor()
```

Big Data

```
# Criar a tabela no banco
c.execute("""
CREATE TABLE IF NOT EXISTS universities
(
    id INTEGER PRIMARY KEY,
    name TEXT,
    country TEXT,
    state_province TEXT,
    web_pages TEXT,
    domains TEXT
);
""")
```

Big Data

for university in universities:

```
c.execute("INSERT INTO universities (name, country, state_province,  
web_pages, domains) VALUES (?, ?, ?, ?, ?);",  
        (university.get('name'),  
        university.get('country'),  
        university.get('state-province'),  
        ', '.join(university.get('web_pages', [])),  
        ', '.join(university.get('domains', []))))
```

```
con.commit()
```

```
con.close()
```

Montando o repositório

Big Data

Vamos criar um repositório no github para o projeto.

Atenção: o repositório deve conter um README.md que será atualizado em cada aula.

Big Data

Como vocês estão em um computador público façam o login no github em uma aba anônima.

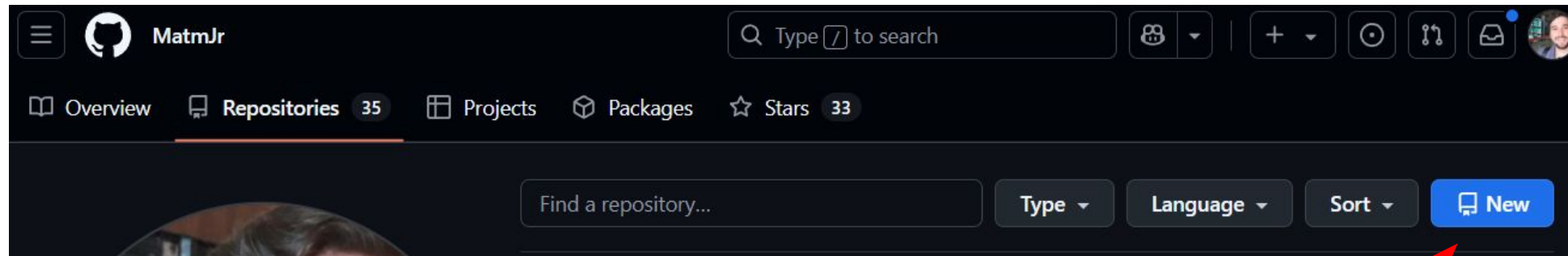
Big Data

Acesse os seus repositórios no github:



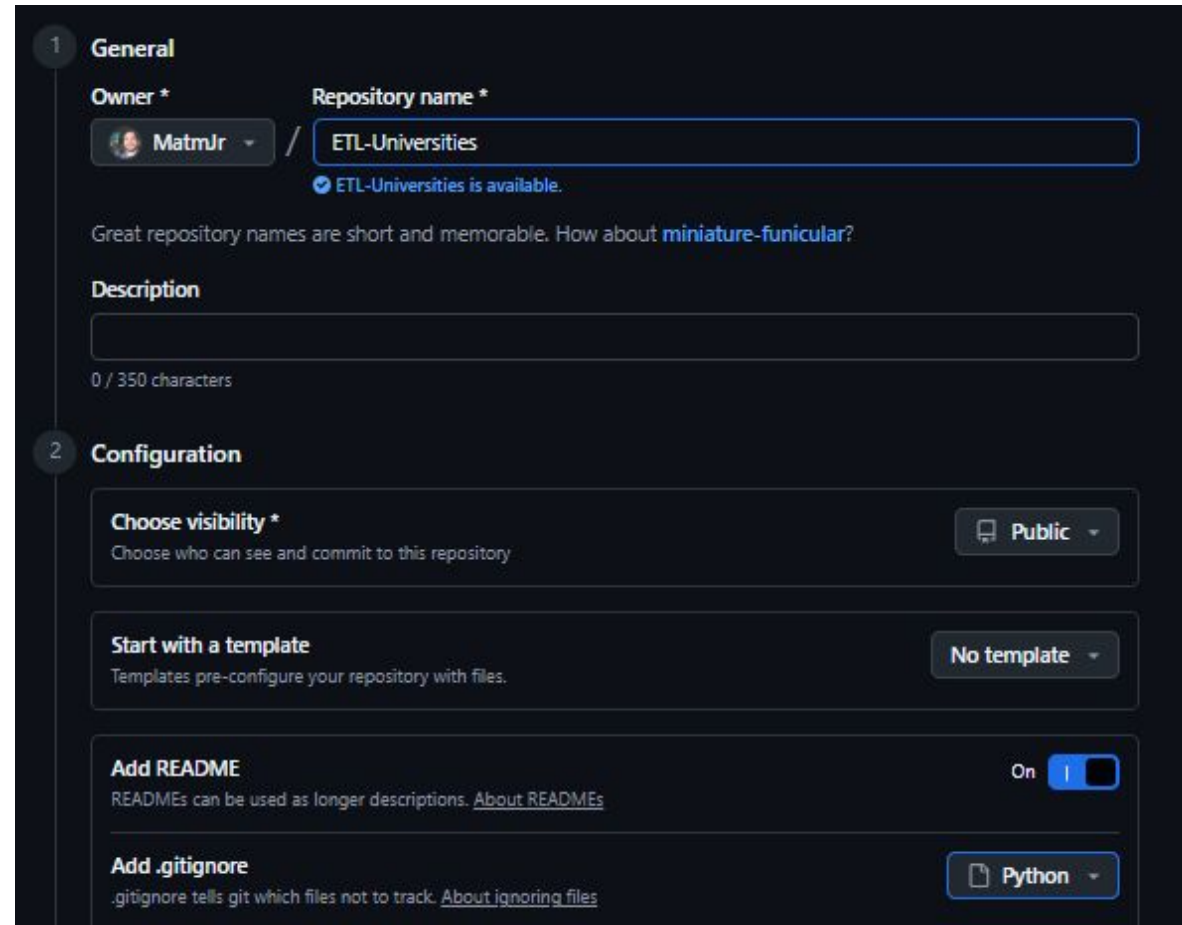
Big Data

Clique no botão “Novo”:



Big Data

Use as
configurações
a seguir:



The screenshot displays the GitHub repository creation process, divided into two main sections: General and Configuration.

1 General

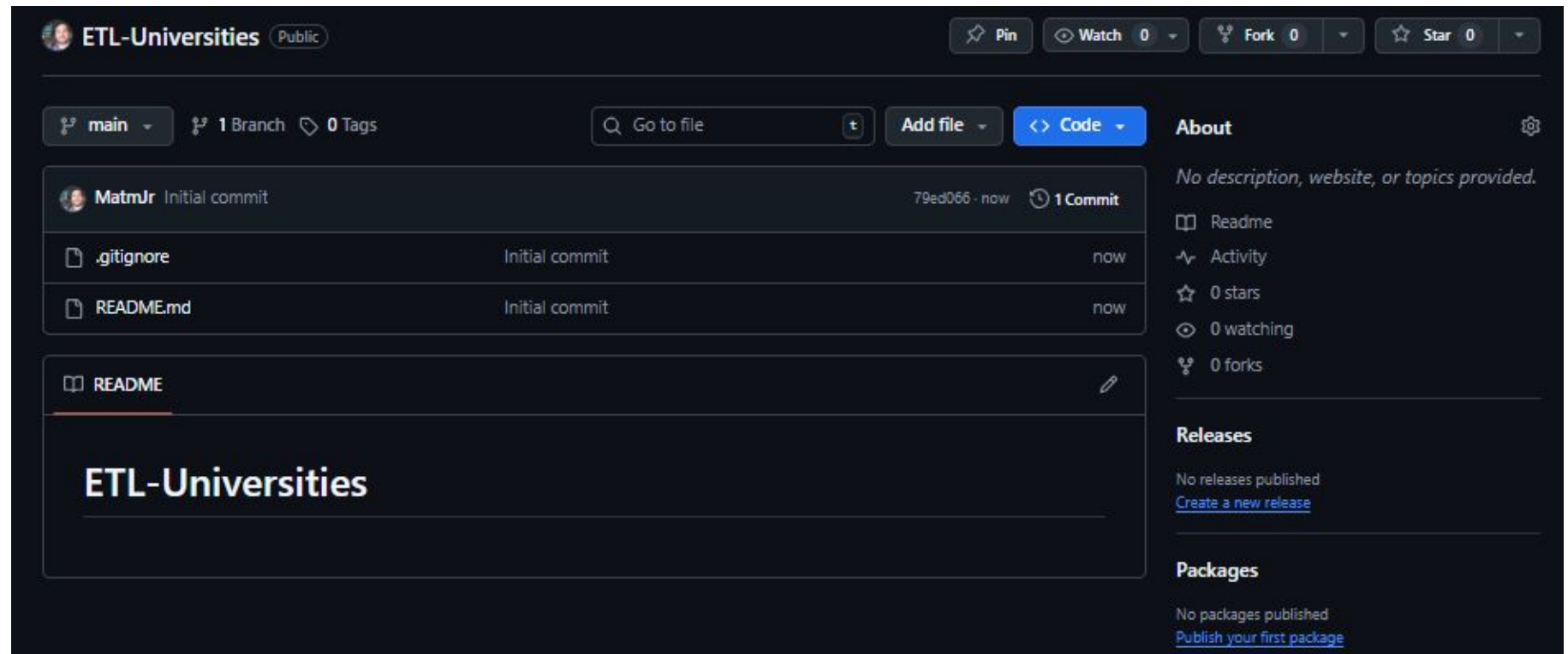
- Owner ***: MatmJr
- Repository name ***: ETL-Universities
- ✔ ETL-Universities is available.
- Great repository names are short and memorable. How about [miniature-funicular](#)?
- Description**: (Empty text box)
- 0 / 350 characters

2 Configuration

- Choose visibility ***: Choose who can see and commit to this repository. **Public** (selected)
- Start with a template**: Templates pre-configure your repository with files. **No template** (selected)
- Add README**: READMEs can be used as longer descriptions. [About READMEs](#). **On** (toggle switch)
- Add .gitignore**: .gitignore tells git which files not to track. [About ignoring files](#). **Python** (selected)

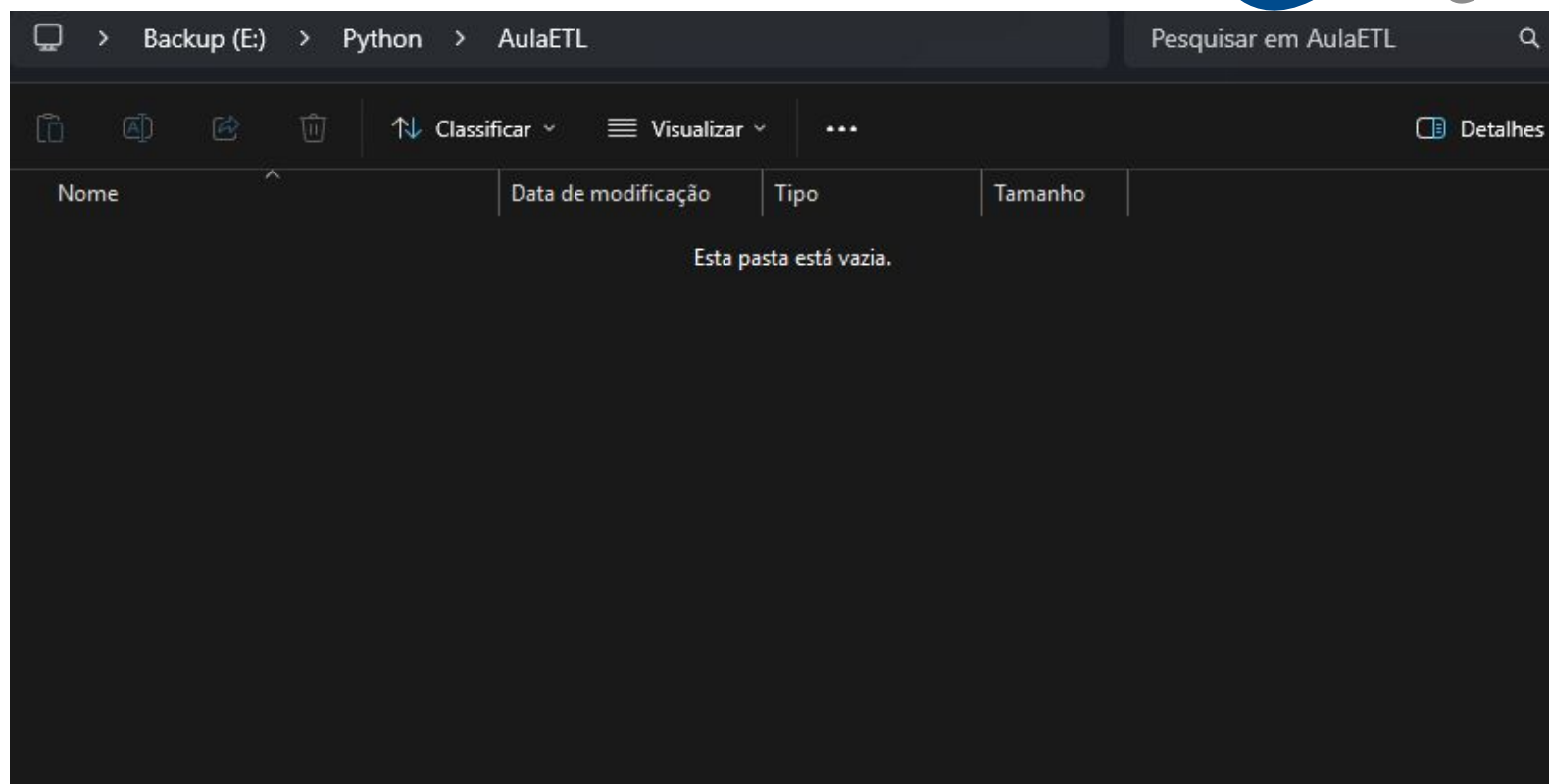
Big Data

Após criar o repositório, faça um clone na sua máquina:



Big Data

Crie uma pasta de trabalho. **Obs:** nos computadores do SENAC as pastas devem ser criadas em Downloads ou Documentos.



Big Data



No terminal que abriu digite:

```
PS E:\Python\AulaETL> git clone sua_url_do_github|
```

OBS.: Se aparecer um erro dizendo que git não é um comando conhecido, você deve instalar o GIT no computador

<https://git-scm.com/downloads>

Big Data

Caso dê tudo certo...

```
Cloning into 'etlBCB'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.
```

Big Data

se você digitar dir (ou ls no Linux) no terminal:

```
PS E:\Python\AulaETL> dir

Diretório: E:\Python\AulaETL

Mode                LastWriteTime         Length Name
----                -
d-----          20/03/2025   20:00             etlBCB
```

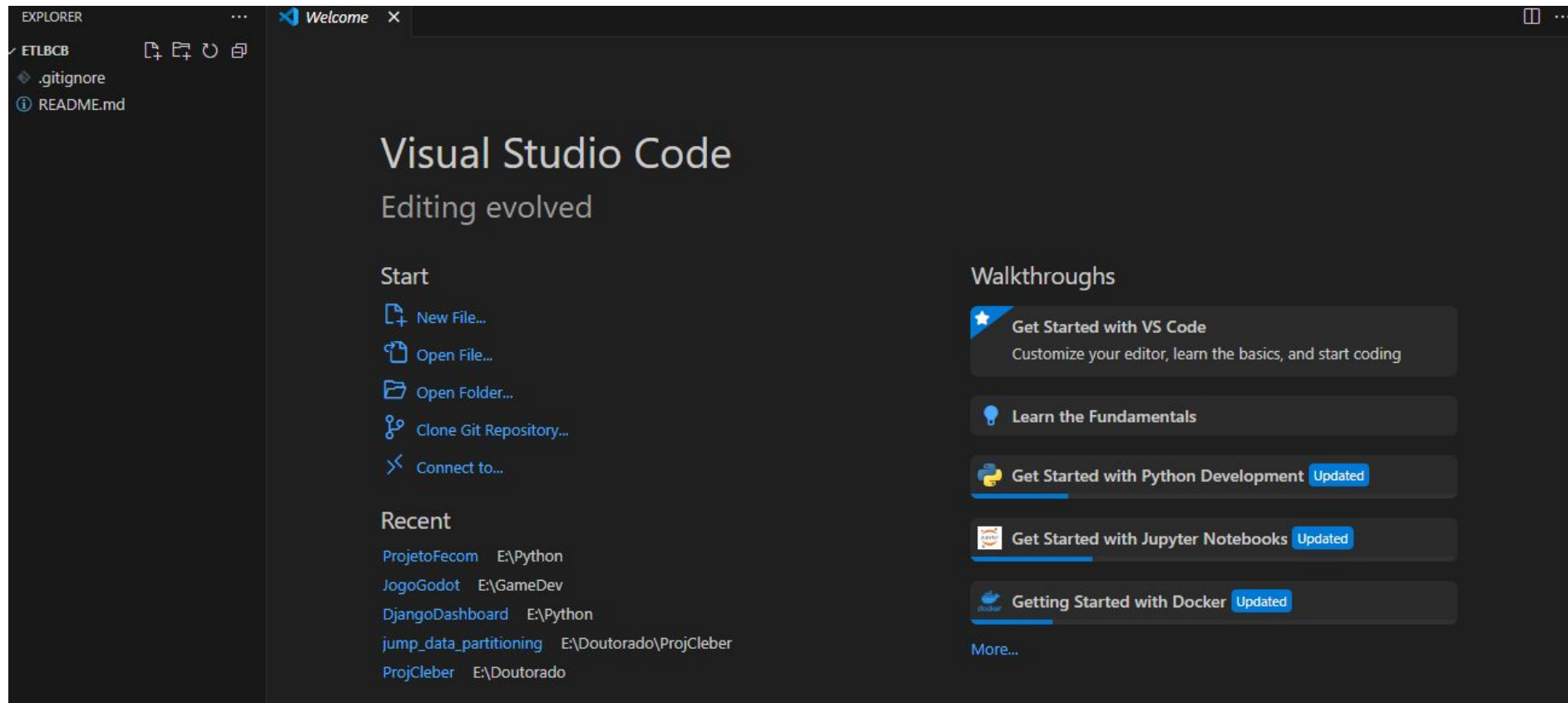
Big Data

Agora digite cd e o nome da pasta que foi criada

```
PS C:\Users\Marco\Desktop\etl> cd .\ETL-Universities\  
PS C:\Users\Marco\Desktop\etl\ETL-Universities> code .
```

E agora digite: **code** .

Big Data



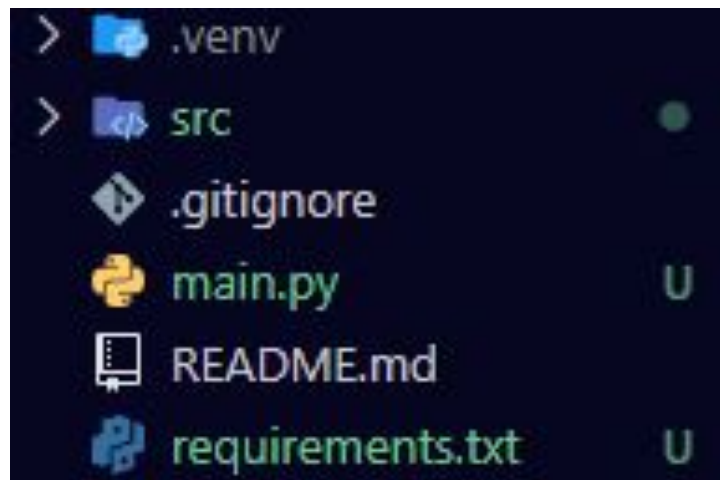
Big Data

Crie um arquivo chamado requirements.txt e adicione 2 linhas neste documento:

```
requests  
pandas
```

Big Data

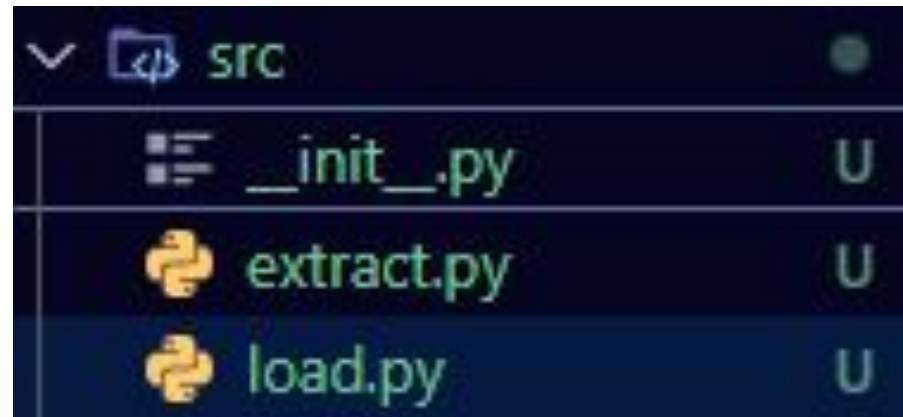
Crie um diretório chamado src e um arquivo chamado main.py fora do src e **execute o processo para criar um venv.**



Construindo uma Solução com POO

Big Data

Dentro do diretório src crie três arquivos: `__init__.py`, [extract.py](#) e [load.py](#)



Big Data

Na camada de extract teremos uma classe responsável por acessar a url e serializar os dados do JSON.

Big Data

[extract.py](#)

```
import requests
```

```
class extract():  
    def __init__(self):  
        pass
```

Big Data

```
def extract_country(self, country):
```

```
    url=f"http://universities.hipolabs.com/search?country={country}"
```

```
    response = requests.get(url)
```

```
    response.raise_for_status()
```

```
    universities = response.json()
```

```
    return universities
```

Big Data



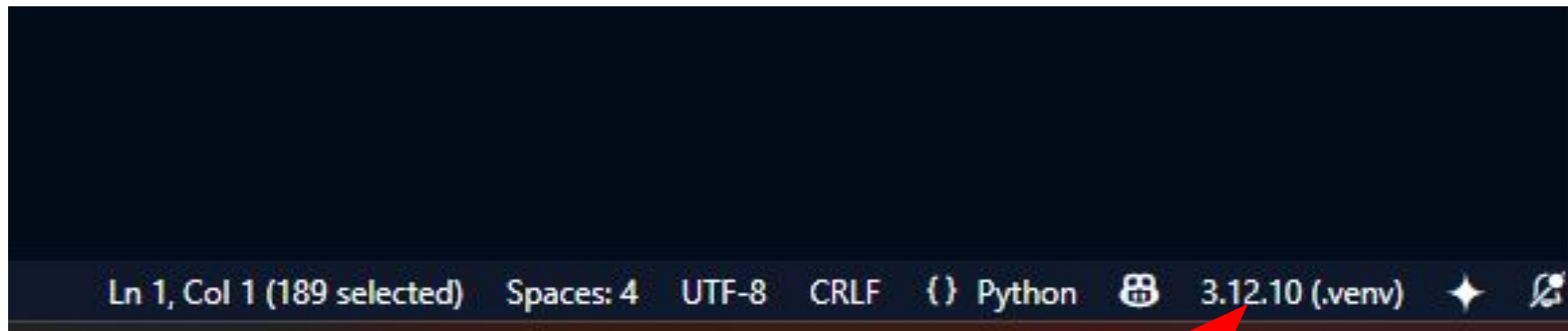
Testando a classe, vamos para o arquivo [main.py](#):

```
from src.extract import extract
```

```
Extract = extract()  
br = Extract.extract_country('Brazil')  
print(type(br))  
print(br)
```

Big Data

Observação: verifique se o venv está ativo no ambiente antes de executar o código.



Big Data

Temos um método que retorna uma lista com dicionários, sendo assim, precisamos criar um método que pegue essa lista e transforme em dados em um banco. Como isso se trata de uma nova responsabilidade é uma boa prática criarmos uma nova classe.

Big Data

[load.py](#)

```
import sqlite3
```

```
class load():  
    def __init__(self):  
        pass
```

Big Data

```
def create_sqlite_table(self, universities_list, db_name, table_name):  
  
    # Criar o banco e se conectar nele  
    con = sqlite3.connect(f"{db_name}.db")  
    c = con.cursor()
```


Big Data

```
c.execute(f'''
    CREATE TABLE IF NOT EXISTS {table_name}
    (
        id INTEGER PRIMARY KEY,
        name TEXT,
        country TEXT,
        state_province TEXT,
        web_pages TEXT,
        domains TEXT
    );
''')
```

Big Data

```
for university in universities_list:
    c.execute(f'''INSERT INTO {table_name} (name, country,
state_province, web_pages, domains) VALUES (?, ?, ?, ?, ?);''',
        (university.get('name'),
        university.get('country'),
        university.get('state-province'),
        ', '.join(university.get('web_pages', [])),
        ', '.join(university.get('domains', []))))

con.commit()
con.close()
```

Big Data

Testando a classe, vamos para o arquivo [main.py](#):

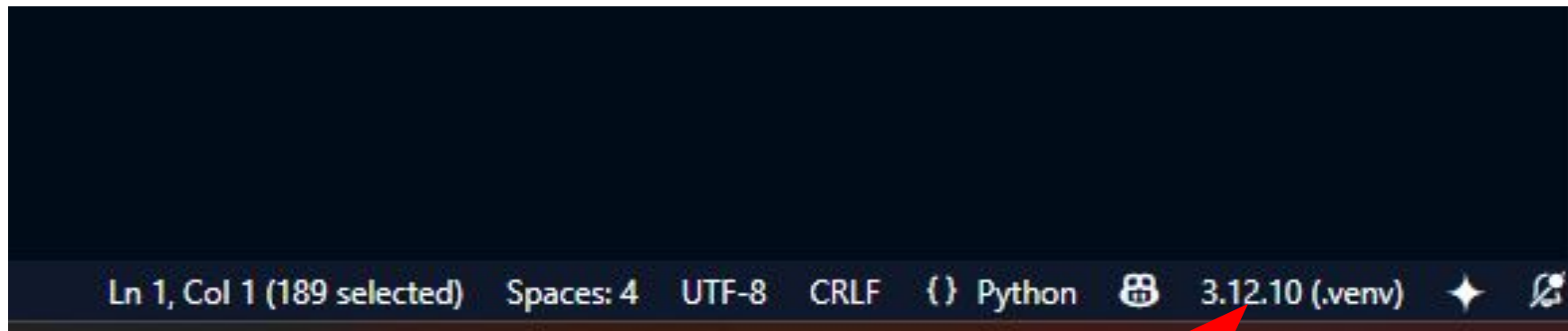
```
from src.extract import extract  
from src.load import load
```

```
Extract = extract()  
Load = load()
```

```
br = Extract.extract_country('Brazil')  
Load.create_sqlite_table(br,'universities', 'universidades_br')
```

Big Data

Observação: verifique se o venv está ativo no ambiente antes de executar o código.



Big Data



Podemos agora extrair os dados de diversos países

```
ch = Extract.extract_country('China')
Load.create_sqlite_table(ch, 'universities', 'universidades_ch')

fr = Extract.extract_country('France')
Load.create_sqlite_table(fr, 'universities', 'universidades_fr')

it = Extract.extract_country('Italy')
Load.create_sqlite_table(it, 'universities', 'universidades_it')
```

Big Data

TABLES		id	name	country	state_pro...	web_pages	domains
> universidades_br		Filter...	Filter...	Filter...	Filter...	Filter...	Filter...
> universidades_ch							
> universidades_fr							
> universidades_it							
	1	NULL	Universidade Comunitária da Região de Chapecó - Unoc...	Brazil	NULL	https://unochapeco.edu.br	unochapeco.edu.br
	2	NULL	Centro Universitário de Brasília, UNICEUB	Brazil	NULL	https://www.uniceub.br	sempreub.com, uniceub.br
	3	NULL	Centro Universitário Barao de Maua	Brazil	NULL	http://www.baraodemaua.br/	baraodemaua.br
	4	NULL	Universidade Braz Cubas	Brazil	NULL	http://www.brazcubas.br/	brazcubas.br
	5	NULL	Universidade Candido Mendes	Brazil	NULL	http://www.candidomendes.br/	candidomendes.br
	6	NULL	Universidade Castelo Branco	Brazil	NULL	http://www.castelobranco.br/	castelobranco.br
	7	NULL	Centro Universitário Claretiano	Brazil	NULL	http://www.claretiano.edu.br/	claretiano.edu.br
	8	NULL	Centro Regional Universitário de Espírito Santo do Pinhal	Brazil	NULL	http://www.creupi.br/	creupi.br
	9	NULL	EMESCAM - Escola Superior de Ciências da Santa Casa d...	Brazil	NULL	http://www.emescam.br/	emescam.br
	10	NULL	Universidade Federal de São Paulo	Brazil	NULL	http://www.epm.br/	epm.br
	11	NULL	Universidade Estácio de Sá	Brazil	NULL	http://www.estacio.br/	estacio.br
	12	NULL	FAAP - Fundação Armando Alvares Penteado	Brazil	NULL	http://www.faap.br/	faap.br
	13	NULL	Faculdades Integradas Curitiba	Brazil	NULL	http://www.faculadadescuritiba.br/	faculadadescuritiba.br
	14	NULL	FAE Business School - Faculdade de Administração e Eco...	Brazil	NULL	http://www.fae.edu/	fae.edu
	15	NULL	Fundação Educacional de Ituverava	Brazil	NULL	http://www.feituverava.com.br/	feituverava.com.br

Melhorando a estrutura do código

Big Data

Quando estamos criando uma solução nossa devemos ir documentando o nosso código, uma forma de fazer isso é por meio de tipagem e docstring dos métodos

```
def extract_country(self, country: str) -> list[dict]:  
    """  
    Método responsável por acessar a url e transformar o json em uma lista de dicionários.  
  
    Args:  
        country: str  
    """
```


Big Data

```
def create_sqlite_table(self, universities_list: list[dict], db_name: str, table_name: str):  
    """  
    Método responsável por criar um banco SQLite e adicionar tabelas nele.  
  
    Args:  
        universities_list: list[dict]  
        db_name: str  
        table_name: str  
    """
```

Big Data

O arquivo `__init__.py` pode expor os métodos ou classes definidos em outros módulos do pacote. Assim, o diretório `src` pode funcionar como um módulo que contém as classes implementadas, se usarmos no `__init__.py` algo como:

```
from .extract import extract  
from .load import load
```

Big Data

Assim na nossa main podemos chamar as classes direto do src, como é feito em outras linguagens:

```
from src import extract
from src import load

Extract = extract()
Load = load()

br = Extract.extract_country('Brazil')
print(br)
```

Big Data

Por fim, podemos utilizar a biblioteca Black para padronizar a formatação do código, garantindo indentação e estilo consistentes, de acordo com as convenções adotadas pela comunidade Python.

Big Data

Abra um novo terminal no vscode e digite cmd:

```
PS C:\Users\Marco\Desktop\etl\ETL-Universities> cmd
Microsoft Windows [versão 10.0.26100.4946]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Marco\Desktop\etl\ETL-Universities>
```

Big Data



Entre no venv criado no início da aula

```
C:\Users\Marco\Desktop\etl\ETL-Universities>.venv\Scripts\activate
```

```
(.venv) C:\Users\Marco\Desktop\etl\ETL-Universities>
```

Big Data

Instale a biblioteca black:

```
(.venv) C:\Users\Marco\Desktop\etl\ETL-Universities>pip install black
```

Big Data

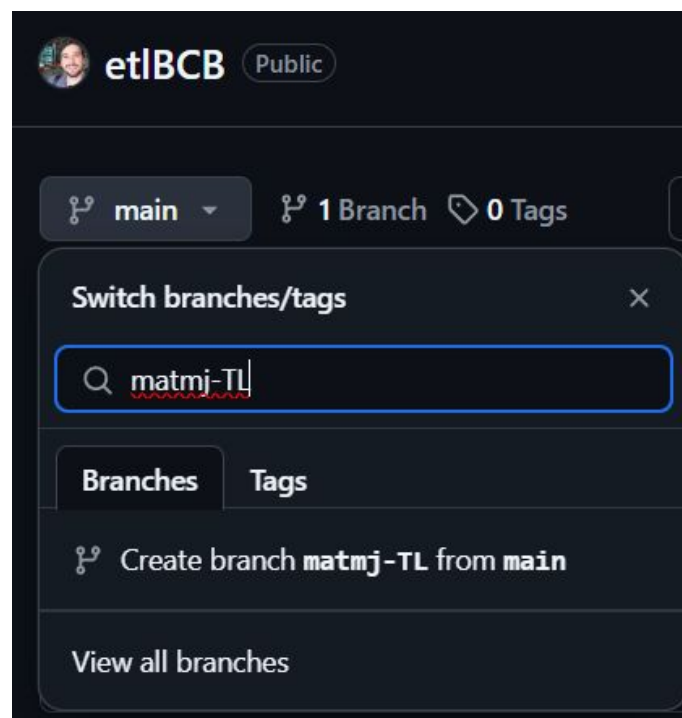
Para executar a biblioteca e formatar o código use o comando:

```
(.venv) C:\Users\Marco\Desktop\etl\ETL-Universities>black .  
reformatted C:\Users\Marco\Desktop\etl\ETL-Universities\main.py  
reformatted C:\Users\Marco\Desktop\etl\ETL-Universities\src\__init__.py  
reformatted C:\Users\Marco\Desktop\etl\ETL-Universities\src\extract.py  
reformatted C:\Users\Marco\Desktop\etl\ETL-Universities\src\load.py  
  
All done! ✨ 🍰 ✨  
4 files reformatted.
```


Salvando no Github

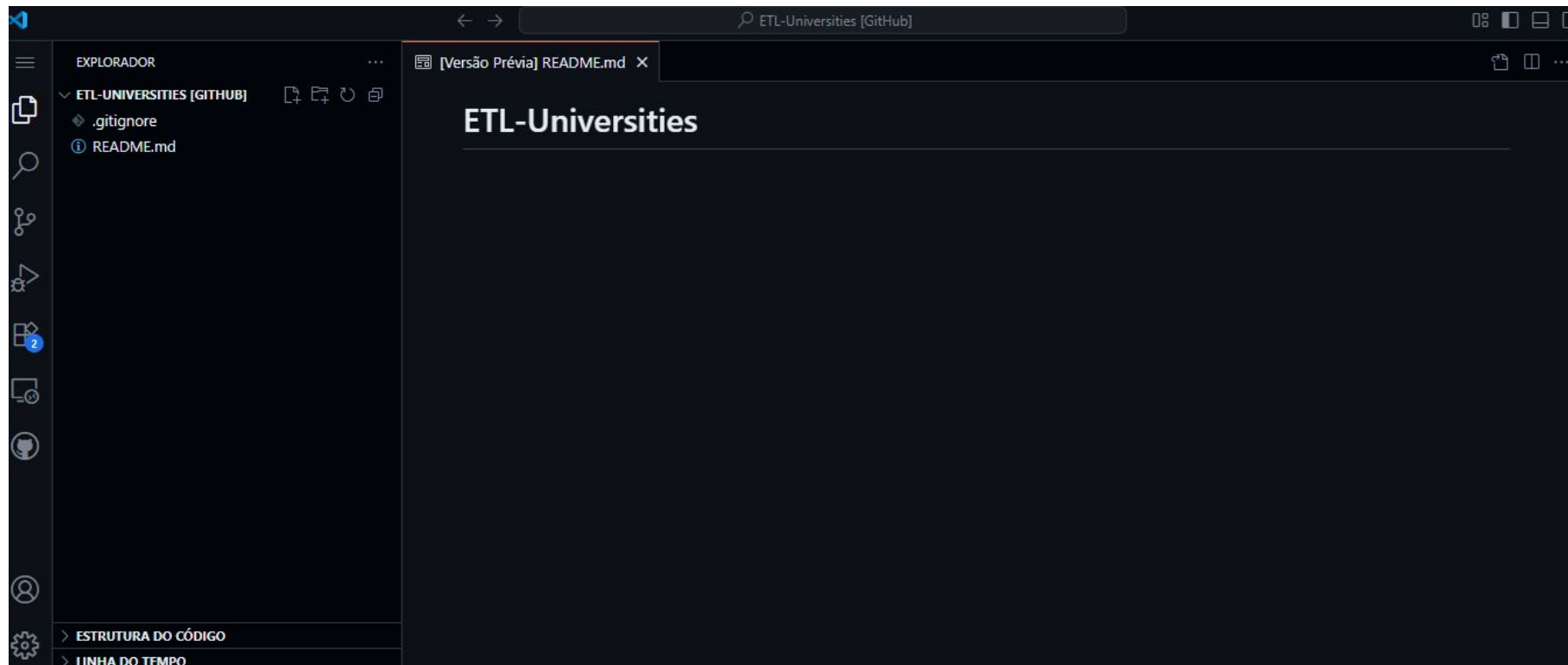
Big Data

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:



Big Data

Aperte o botão “.” no teclado dentro do repositório

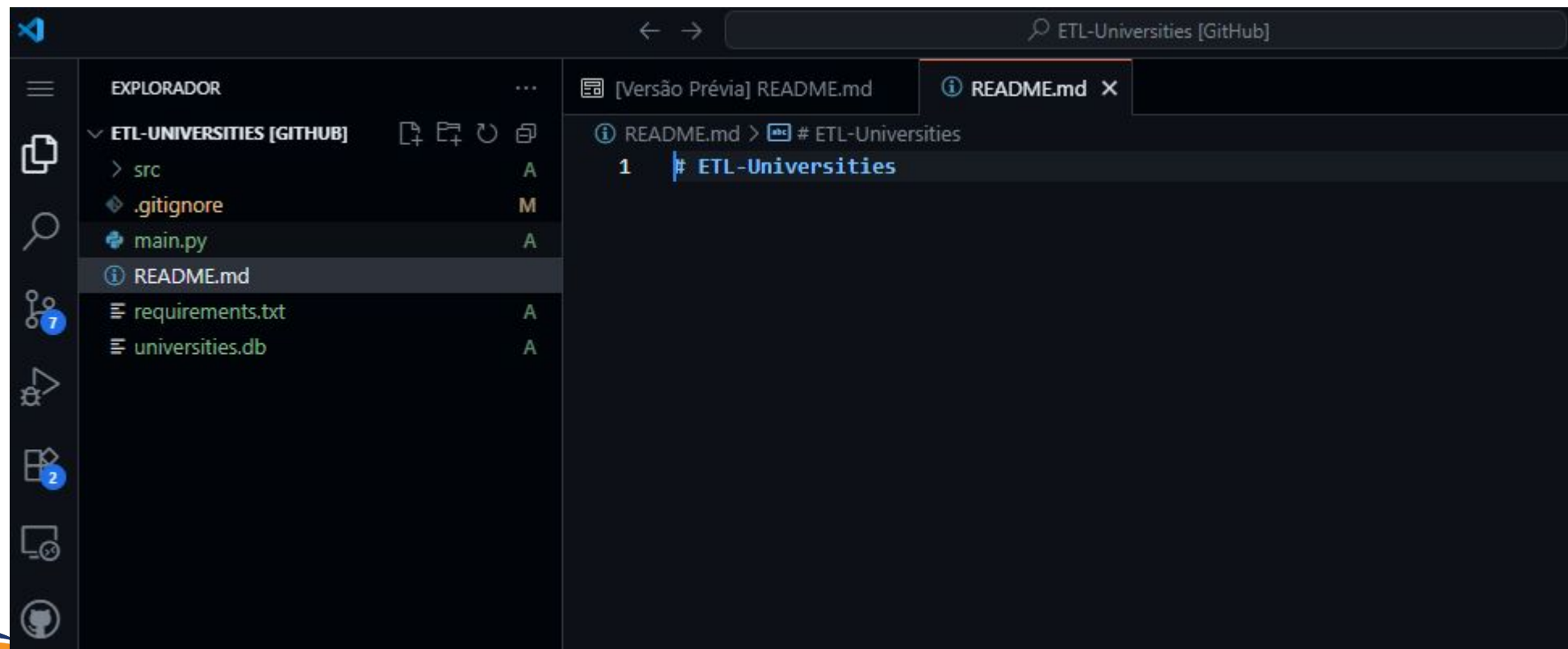


Big Data

Vá na pasta do projeto e selecione todos os arquivos, menos o .venv e arraste tudo para dentro do repositório

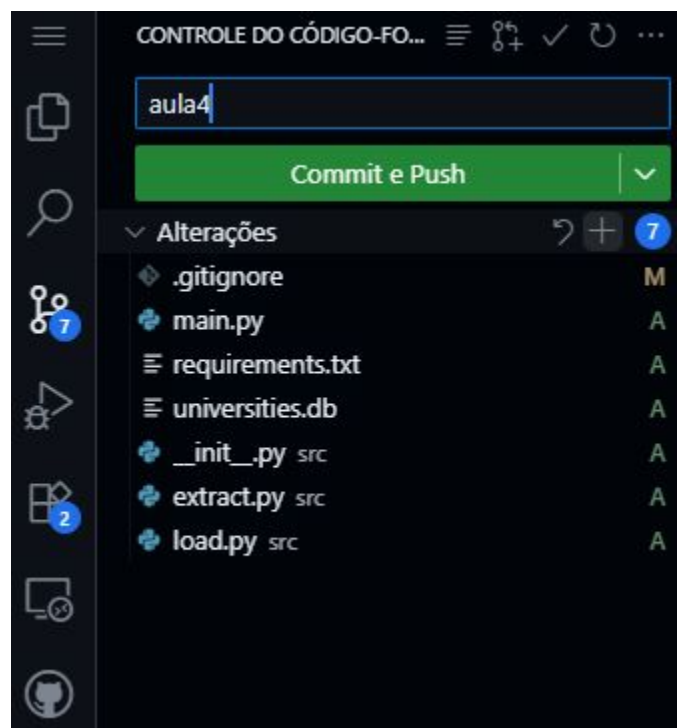
src	03/09/2025 11:43	Pasta de arquivos	
.gitignore	03/09/2025 10:46	Arquivo Fonte Git ...	5 KB
main.py	03/09/2025 12:43	Arquivo PY	1 KB
README.md	03/09/2025 10:46	Arquivo Fonte Ma...	1 KB
requirements.txt	03/09/2025 11:04	Documento de Te...	1 KB
universities.db	03/09/2025 12:09	Data Base File	148 KB

Big Data



Big Data

Na aba de controle de código-fonte digite uma mensagem para o commit, clique no “+” que está em alterações e depois clique em “Commit e Push”.



Dúvidas?



Marco Mialaret, MSc

Telefone:

81 98160 7018

E-mail:

marcomialaret@gmail.com

