

Universidade Comunitária da Região de Chapecó – Unochapecó

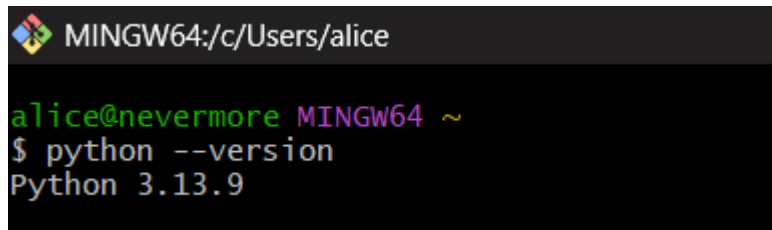
Escola Politécnica – Ciência da Computação/Sistemas de Informação

Banco de Dados II

Profa. Monica Tissiani De Toni Pereira

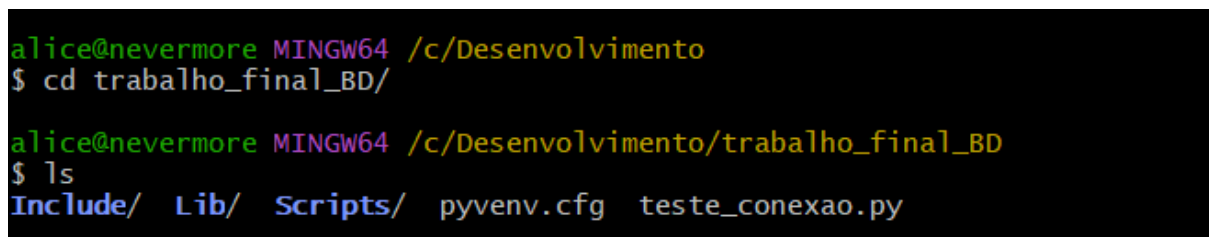
Alunos: Alice Botton Dal Paz, Anthony Guilherme Cazuni da Silva e Gabriel Henrique Robette Ferri

1. Instalação do Python

A terminal window with a dark background. The title bar shows a Windows icon and the path 'MINGW64:/c/Users/alice'. The prompt is 'alice@nevermore MINGW64 ~'. The user enters '\$ python --version' and the output is 'Python 3.13.9'.

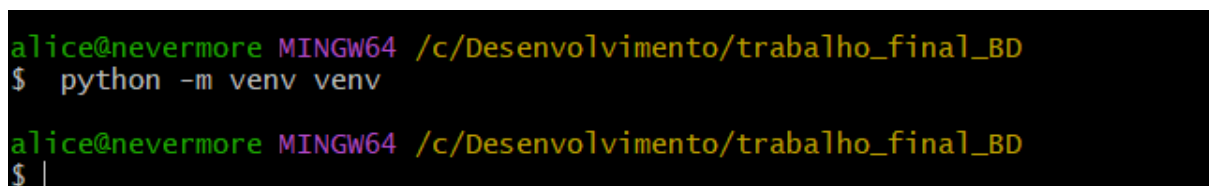
```
MINGW64:/c/Users/alice  
alice@nevermore MINGW64 ~  
$ python --version  
Python 3.13.9
```

2. Criação do diretório do projeto

A terminal window with a dark background. The prompt is 'alice@nevermore MINGW64 /c/Desenvolvimento'. The user enters '\$ cd trabalho_final_BD/'. The prompt changes to 'alice@nevermore MINGW64 /c/Desenvolvimento/trabalho_final_BD'. The user enters '\$ ls' and the output is 'Include/ Lib/ Scripts/ pyenv.cfg teste_conexao.py'.

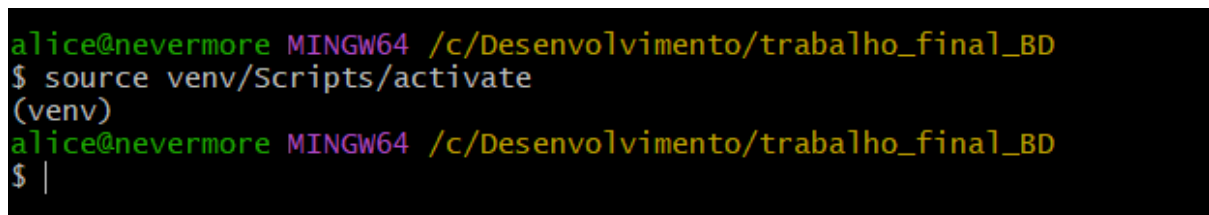
```
alice@nevermore MINGW64 /c/Desenvolvimento  
$ cd trabalho_final_BD/  
alice@nevermore MINGW64 /c/Desenvolvimento/trabalho_final_BD  
$ ls  
Include/ Lib/ Scripts/ pyenv.cfg teste_conexao.py
```

3. Criação do ambiente virtual (venv)

A terminal window with a dark background. The prompt is 'alice@nevermore MINGW64 /c/Desenvolvimento/trabalho_final_BD'. The user enters '\$ python -m venv venv'. The prompt changes to 'alice@nevermore MINGW64 /c/Desenvolvimento/trabalho_final_BD'. The user enters '\$ |' and the prompt changes to '\$ |'.

```
alice@nevermore MINGW64 /c/Desenvolvimento/trabalho_final_BD  
$ python -m venv venv  
alice@nevermore MINGW64 /c/Desenvolvimento/trabalho_final_BD  
$ |
```

4. Ativação do ambiente virtual

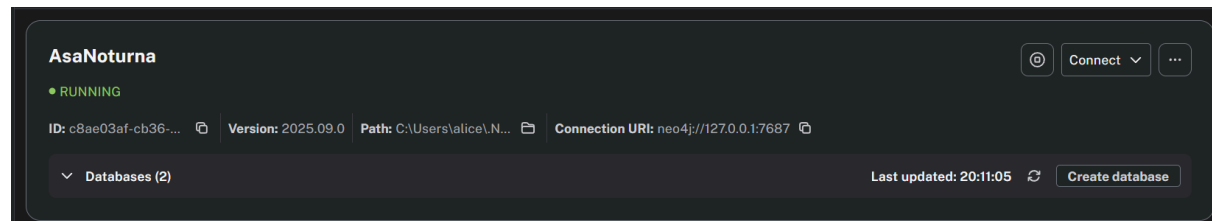
A terminal window with a dark background. The prompt is 'alice@nevermore MINGW64 /c/Desenvolvimento/trabalho_final_BD'. The user enters '\$ source venv/Scripts/activate'. The prompt changes to '(venv)'. The user enters 'alice@nevermore MINGW64 /c/Desenvolvimento/trabalho_final_BD'. The user enters '\$ |' and the prompt changes to '\$ |'.

```
alice@nevermore MINGW64 /c/Desenvolvimento/trabalho_final_BD  
$ source venv/Scripts/activate  
(venv)  
alice@nevermore MINGW64 /c/Desenvolvimento/trabalho_final_BD  
$ |
```

5. Instalação do Neo4j

```
(venv)
alice@nevermore MINGW64 /c/Desenvolvimento/trabalho_final_BD
$ pip install neo4j
Collecting neo4j
  Downloading neo4j-6.0.3-py3-none-any.whl.metadata (5.2 kB)
Collecting pytz (from neo4j)
  Using cached pytz-2025.2-py2.py3-none-any.whl.metadata (22 kB)
Downloading neo4j-6.0.3-py3-none-any.whl (325 kB)
Using cached pytz-2025.2-py2.py3-none-any.whl (509 kB)
Installing collected packages: pytz, neo4j
Successfully installed neo4j-6.0.3 pytz-2025.2
```

6. Inicialização da instância do Neo4j ativa

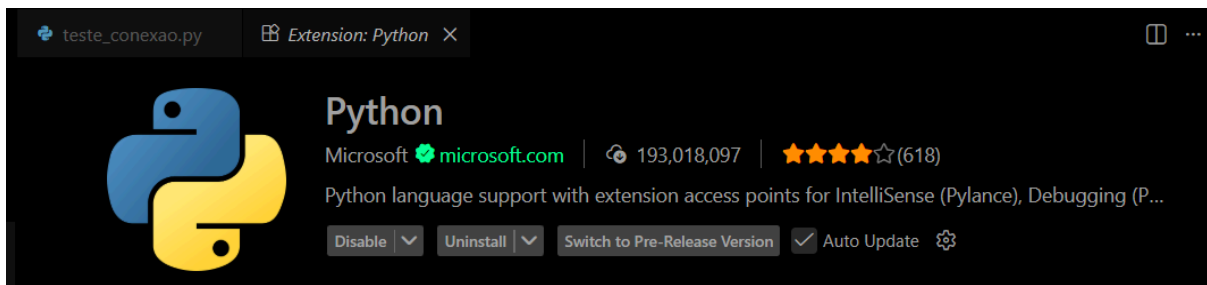


7. Escolha da IDE (VSCode)

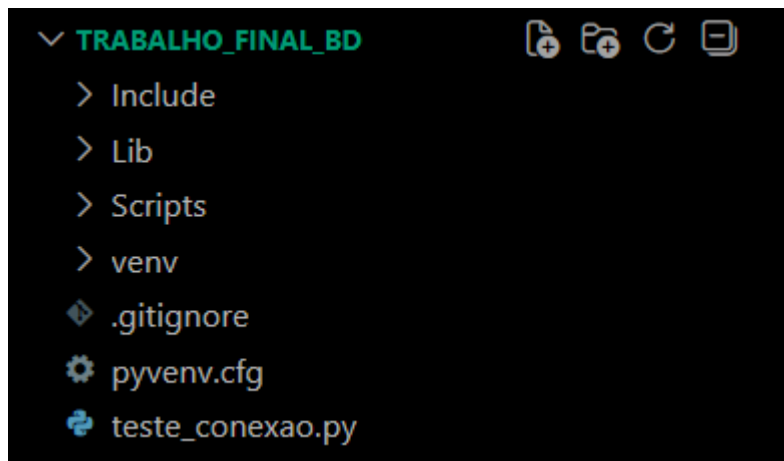
```
▼ TRABALHO_FINAL_BD
  > Include
  > Lib
  > Scripts
  > venv
  .gitignore
  pyvenv.cfg
  teste_conexao.py

teste_conexao.py
1  from neo4j import GraphDatabase
2
3  URI = "bolt://localhost:7687"
4  USUARIO = "neo4j"
5  SENHA = "trabalho"
6
7  try:
8      driver = GraphDatabase.driver(URI, auth=(USUARIO, SENHA))
9
10     driver.verify_connectivity()
11     print("Conexão bem-sucedida!")
12
13     with driver.session(database="neo4j") as session:
14         query = "MATCH (p:Pessoa) RETURN p.nome AS nome"
15         result = session.run(query)
16
17         print("Pessoas encontradas no banco:")
18         count = 0
19         for record in result:
20             print(record["nome"])
21             count += 1
22
23         if count == 0:
24             print("Nenhuma pessoa encontrada")
25
26 except Exception as e:
27     print(f"Ocorreu um erro: {e}")
28
29 finally:
30     if 'driver' in locals() and driver:
31         driver.close()
32     print("Conexão fechada.")
```

8. Instalação da extensão Python no VSCode



9. Abertura da pasta do projeto no VSCode



10. Criação e Teste do Arquivo Python

```
(venv) PS C:\Desenvolvimento\trabalho_final_BD> .\venv\Scripts\Activate
(venv) PS C:\Desenvolvimento\trabalho_final_BD> python --version
Python 3.13.9
(venv) PS C:\Desenvolvimento\trabalho_final_BD> pip --version
pip 25.2 from C:\Desenvolvimento\trabalho_final_BD\venv\Lib\site-packages\pip (python 3.13)
(venv) PS C:\Desenvolvimento\trabalho_final_BD> pip list
Package Version
-----
neo4j    6.0.3
pip      25.2
pytz     2025.2
(venv) PS C:\Desenvolvimento\trabalho_final_BD>
```

11. Métodos da classe **Driver** da biblioteca Neo4j em Python

Na biblioteca oficial do Neo4j para Python (**neo4j**), o objeto **Driver** é responsável por gerenciar a conexão e fornecer sessões para executar consultas. Os principais métodos da classe **Driver** são:

1. **session()**

Cria uma sessão para executar transações ou consultas.

```
from neo4j import GraphDatabase
```

```
driver = GraphDatabase.driver("bolt://localhost:7687", auth=("neo4j",
"senha"))
```

```
with driver.session() as session:
```

```
result = session.run("MATCH (n) RETURN n LIMIT 5")
for record in result:
    print(record)
```

2. `close()`

Encerra a conexão com o banco de dados e libera os recursos.

```
driver.close()
```

3. `verify_connectivity()`

Testa se a conexão com o banco está ativa e funcionando.

```
try:
    driver.verify_connectivity()
    print("Conexão bem-sucedida!")
except Exception as e:
    print(f"Falha na conexão: {e}")
```

4. `execute_query()`

Executa uma consulta sem precisar abrir a sessão manualmente.

```
result = driver.execute_query("MATCH (p:Pessoa) RETURN p.nome AS nome")
for record in result.records:
    print(record["nome"])
```

5. `session(database="nome_do_banco")`

Permite escolher explicitamente em qual banco executar as consultas.

```
with driver.session(database="neo4j") as session:
    result = session.run("MATCH (n) RETURN count(n)")
    print(result.single()[0])
```

12. Pergunte também como o texto gerado pela IA pode ser referenciado na documentação do projeto e adicione a referência ao texto gerado.

Quando utilizamos conteúdos gerados por Inteligência Artificial, como explicações técnicas, exemplos de código ou definições conceituais, é importante registrar essa contribuição na

documentação do projeto, preservando a transparência acadêmica. A IA pode ser citada como fonte de consulta, assim como artigos, livros e sites. No caso do ChatGPT (OpenAI), como o conteúdo é gerado dinamicamente e não possui uma página específica, a recomendação é fazer uma referência simples, mencionando a ferramenta, o tema consultado e a data de acesso.

Além disso, deve-se indicar no corpo do texto ou em nota de rodapé quando determinada explicação foi obtida com auxílio da IA. Essa prática demonstra que o material foi construído com apoio tecnológico, mas analisado e interpretado pelo estudante.

A seguir, um exemplo de como essa referência pode ser incorporada ao projeto:

→ *Parte das explicações sobre os métodos da classe **Driver** da biblioteca Neo4j em Python foram elaboradas com o auxílio da ferramenta ChatGPT, da OpenAI, utilizada como apoio técnico para consulta e esclarecimento de conceitos.*

E nas referências do documento, pode ser registrada da seguinte forma:

OPENAI. ChatGPT. Respostas fornecidas sobre os métodos da classe Driver do Neo4j em Python. Acesso em: 20 nov. 2025.

Essa forma de registro atende ao requisito de documentar o uso da IA como fonte de consulta, sem necessidade de incluir um link específico, já que o conteúdo é acessado por meio de uma plataforma interativa.


```
gabri@AcerFerri MINGW64 ~  
$ pip install psycpg2-binary redis  
Defaulting to user installation because normal site-packages is not writeable  
Collecting psycpg2-binary  
  Downloading psycpg2_binary-2.9.11-cp313-cp313-win_amd64.whl.metadata (5.1 kB)  
Collecting redis  
  Downloading redis-7.0.1-py3-none-any.whl.metadata (12 kB)  
Downloading psycpg2_binary-2.9.11-cp313-cp313-win_amd64.whl (2.7 MB)  
  2.7/2.7 MB 3.8 MB/s eta 0:00:00  
Downloading redis-7.0.1-py3-none-any.whl (339 kB)  
Installing collected packages: redis, psycpg2-binary  
Successfully installed psycpg2-binary-2.9.11 redis-7.0.1  
  
[notice] A new release of pip is available: 25.1.1 -> 25.3  
[notice] To update, run: python.exe -m pip install --upgrade pip  
  
gabri@AcerFerri MINGW64 ~  
$
```