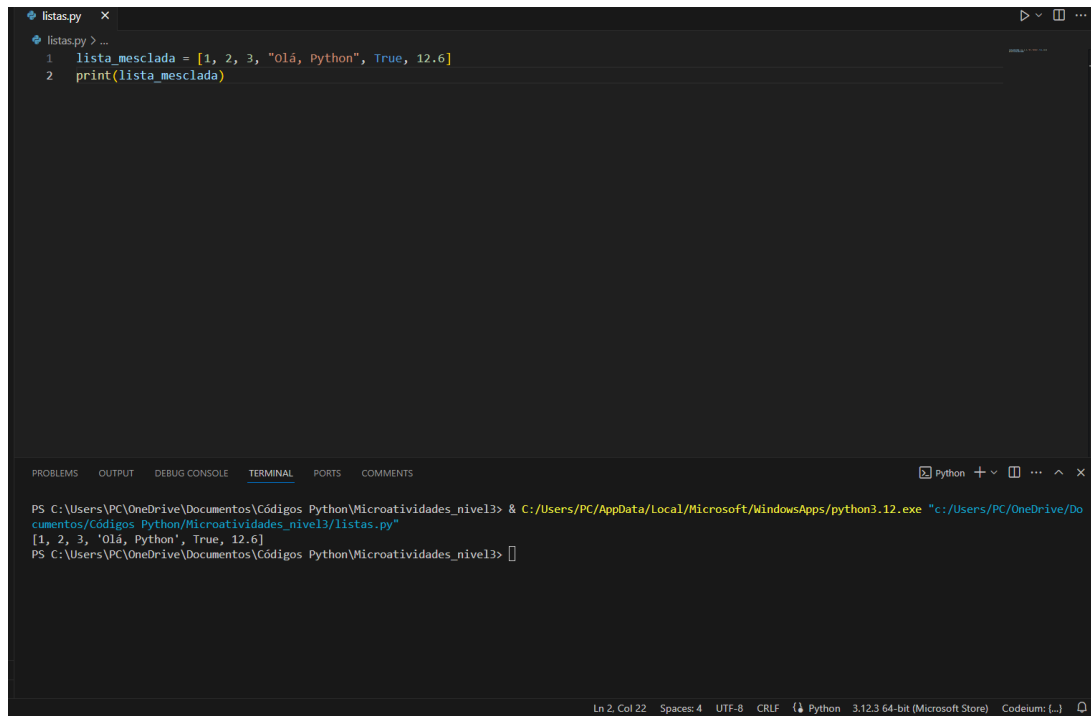


DESENVOLVIMENTO FULL STACK

NÍVEL 3: ESTRUTURANDO OS DADOS

Microatividade-1: Descrever a manipulação da estrutura de dados lista em Python

M 1.1 #Conteúdo da lista



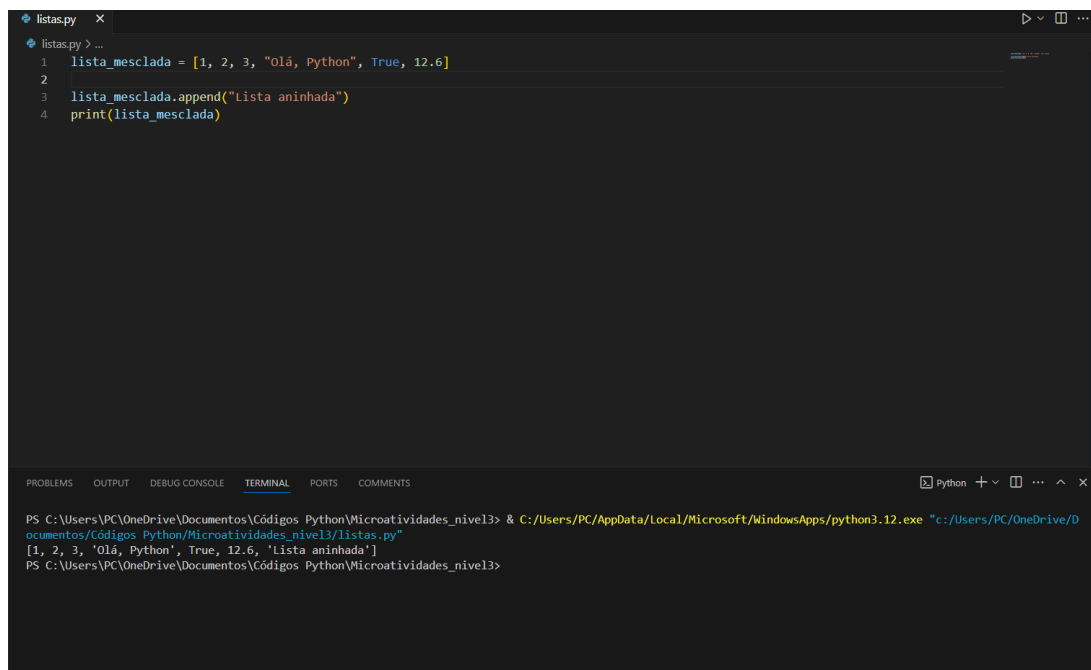
```
listas.py x
listas.py > ...
1 lista_mesclada = [1, 2, 3, "Olá, Python", True, 12.6]
2 print(lista_mesclada)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Python + - ... ^ x

PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/listas.py"
[1, 2, 3, 'Olá, Python', True, 12.6]
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>

Ln 2, Col 22 Spaces: 4 UTF-8 CRLF Python 3.12.3 64-bit (Microsoft Store) Codeium: [...]
```

M 1.2 #Método “append” adicionando o elemento [“Listinha aninhada”] a lista.

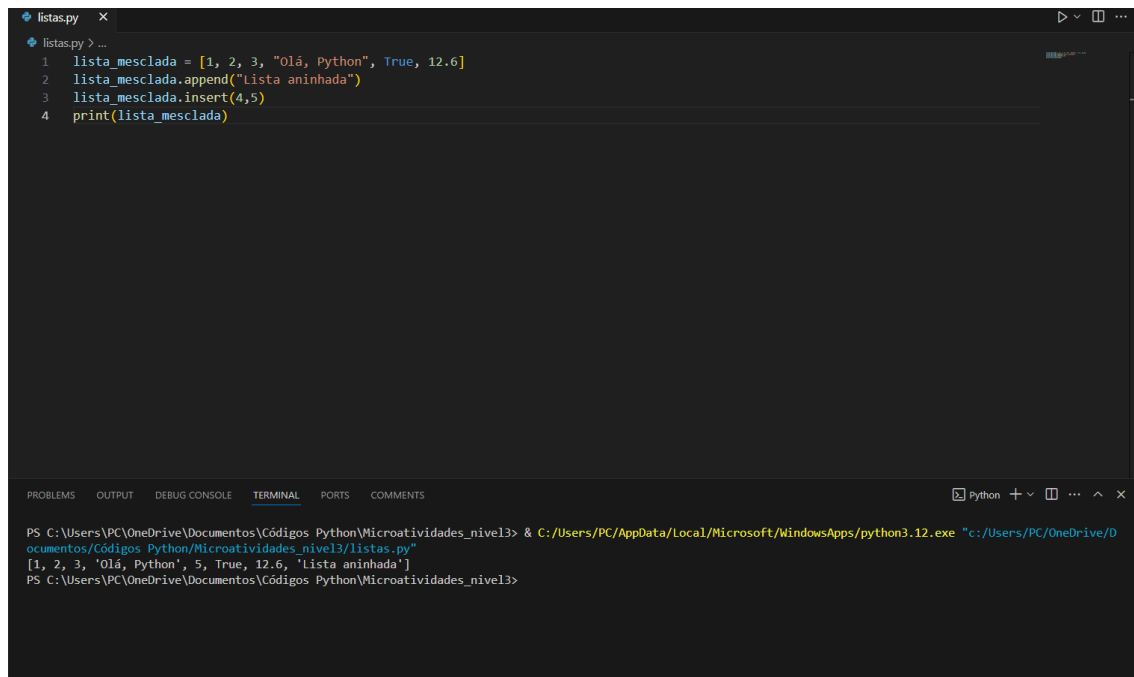


```
listas.py x
listas.py > ...
1 lista_mesclada = [1, 2, 3, "Olá, Python", True, 12.6]
2
3 lista_mesclada.append("Lista aninhada")
4 print(lista_mesclada)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Python + - ... ^ x

PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/listas.py"
[1, 2, 3, 'Olá, Python', True, 12.6, 'Lista aninhada']
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 1.3 #Com o método “insert” inserir o elemento 5 na posição 4 da lista.

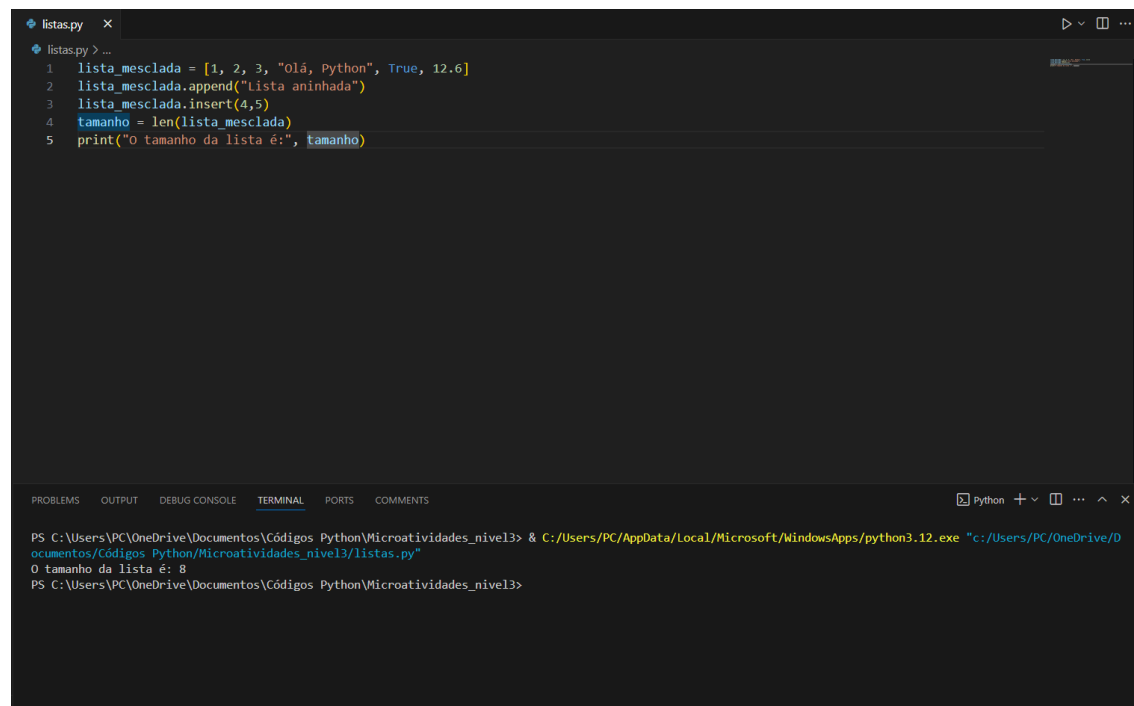


```
listas.py x
listas.py > ...
1 lista_mesclada = [1, 2, 3, "Olá, Python", True, 12.6]
2 lista_mesclada.append("Lista aninhada")
3 lista_mesclada.insert(4,5)
4 print(lista_mesclada)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS Python + - ... ^ x

```
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/listas.py"
[1, 2, 3, 'Olá, Python', 5, True, 12.6, 'Lista aninhada']
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 1.4 #Imprimir o tamanho atual da lista.

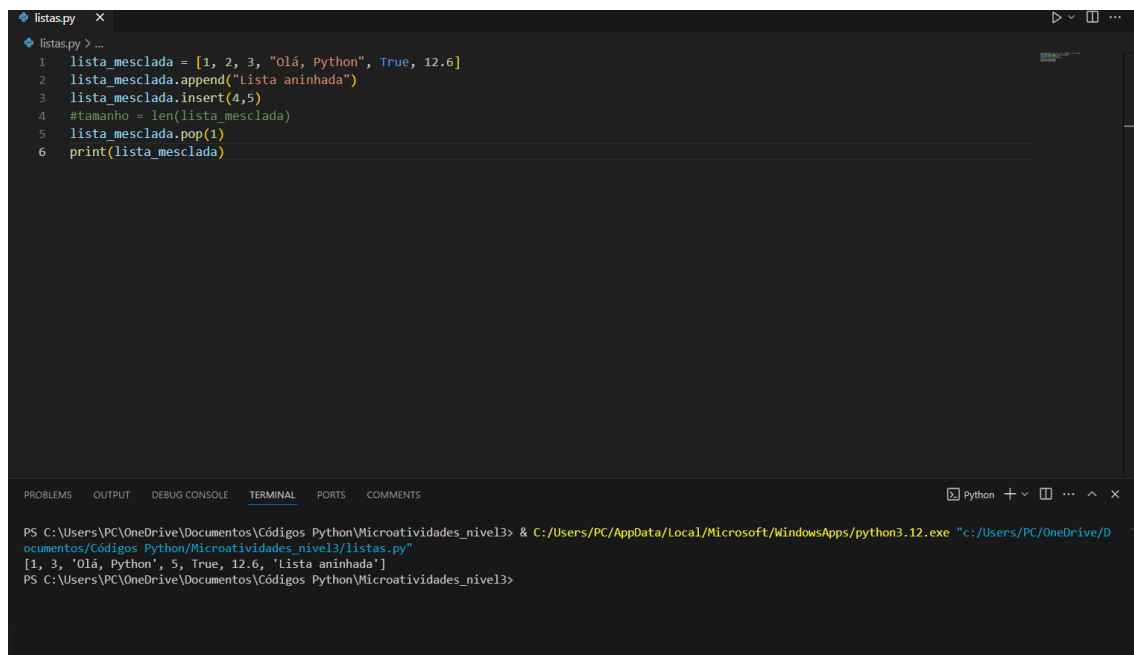


```
listas.py x
listas.py > ...
1 lista_mesclada = [1, 2, 3, "Olá, Python", True, 12.6]
2 lista_mesclada.append("Lista aninhada")
3 lista_mesclada.insert(4,5)
4 tamanho = len(lista_mesclada)
5 print("O tamanho da lista é:", tamanho)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS Python + - ... ^ x

```
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/listas.py"
O tamanho da lista é: 8
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 1.5 #Remover o item que está na posição 1 na lista.



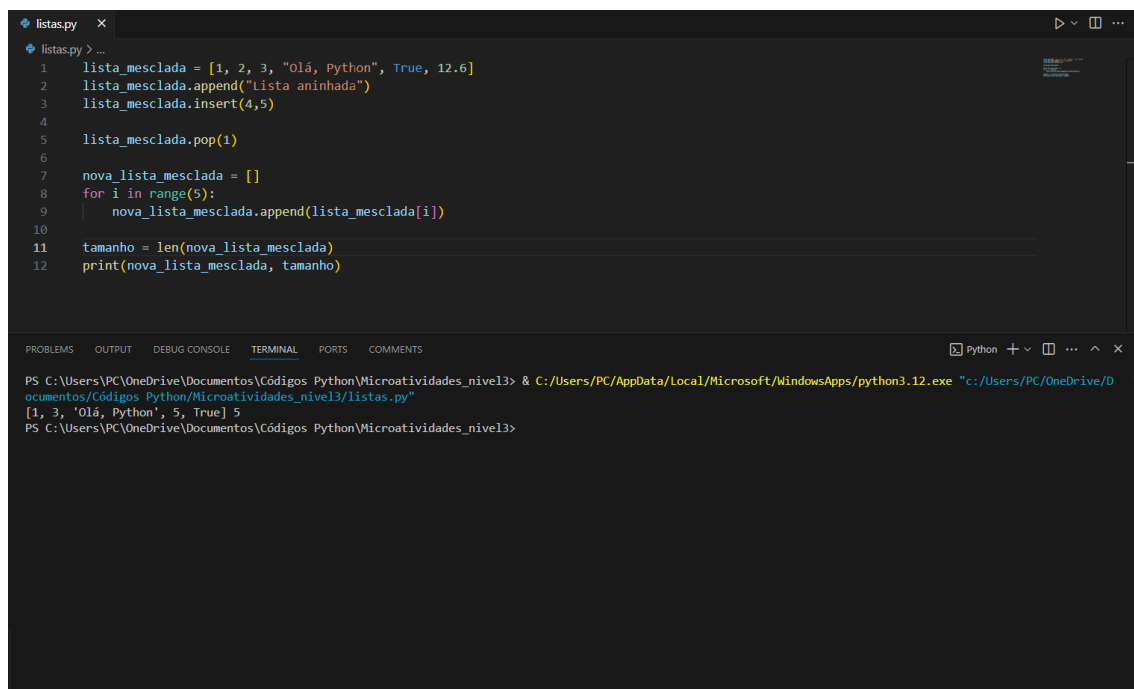
The screenshot shows a VS Code editor window with a file named `listas.py`. The code in the editor is as follows:

```
1 lista_mesclada = [1, 2, 3, "Olá, Python", True, 12.6]
2 lista_mesclada.append("Lista aninhada")
3 lista_mesclada.insert(4,5)
4 #tamanho = len(lista_mesclada)
5 lista_mesclada.pop(1)
6 print(lista_mesclada)
```

Below the editor, the terminal window shows the command to run the script and its output:

```
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/listas.py"
[1, 3, 'Olá, Python', 5, True, 12.6, 'Lista aninhada']
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 1.6 #Criar uma nova lista e adicionar a ela os itens até a posição 4 da lista anterior.



The screenshot shows a VS Code editor window with a file named `listas.py`. The code in the editor is as follows:

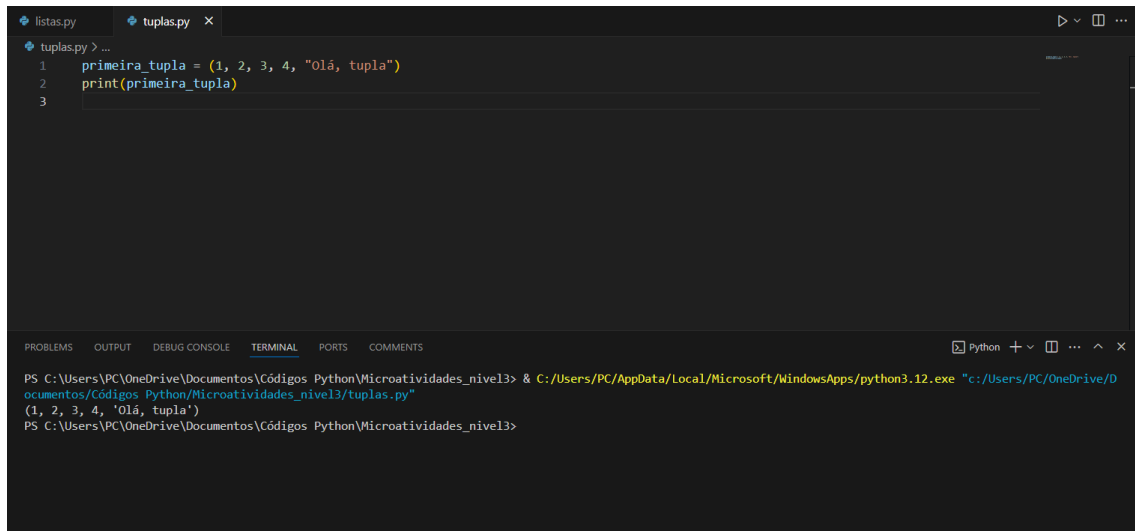
```
1 lista_mesclada = [1, 2, 3, "Olá, Python", True, 12.6]
2 lista_mesclada.append("Lista aninhada")
3 lista_mesclada.insert(4,5)
4
5 lista_mesclada.pop(1)
6
7 nova_lista_mesclada = []
8 for i in range(5):
9     nova_lista_mesclada.append(lista_mesclada[i])
10
11 tamanho = len(nova_lista_mesclada)
12 print(nova_lista_mesclada, tamanho)
```

Below the editor, the terminal window shows the command to run the script and its output:

```
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/listas.py"
[1, 3, 'Olá, Python', 5, True] 5
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

Microatividade-2: Descrever a manipulação da estrutura de dados tupla em Python.

M 2.1 #Crie uma estrutura de dados do tipo tupla atribuindo os valores.



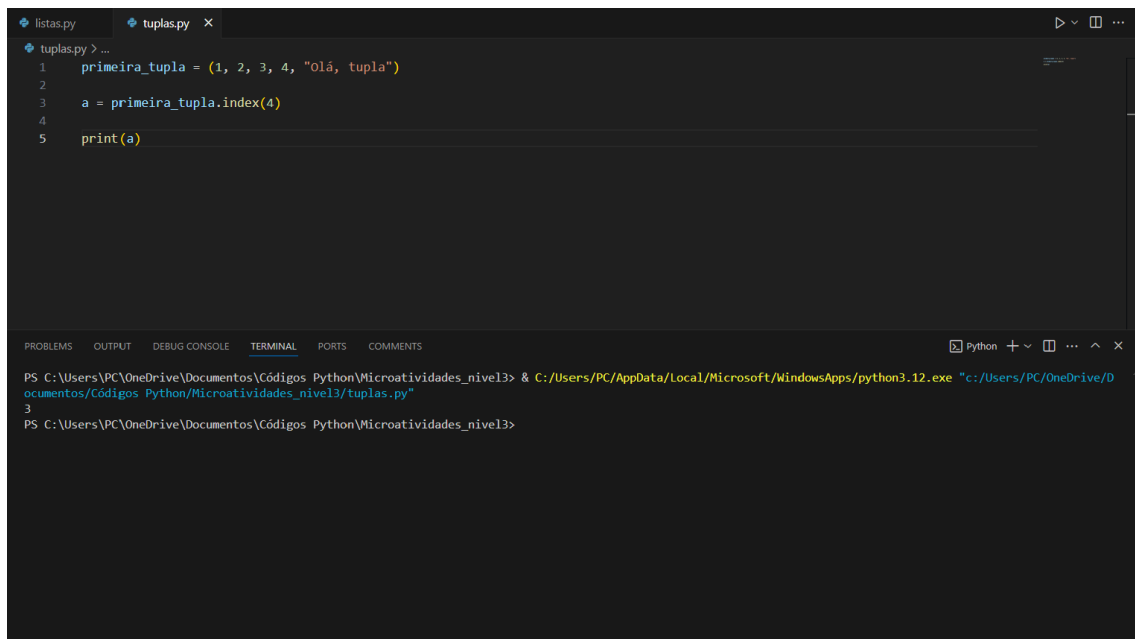
The screenshot shows a Python IDE with two tabs: 'listas.py' and 'tuplas.py'. The 'tuplas.py' tab is active, displaying the following code:

```
1 primeira_tupla = (1, 2, 3, 4, "Olá, tupla")
2 print(primeira_tupla)
3
```

The bottom panel shows the terminal output:

```
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/tuplas.py"
(1, 2, 3, 4, 'Olá, tupla')
```

M 2.2 #Com o método “index”, adicione o elemento 4 da tupla.



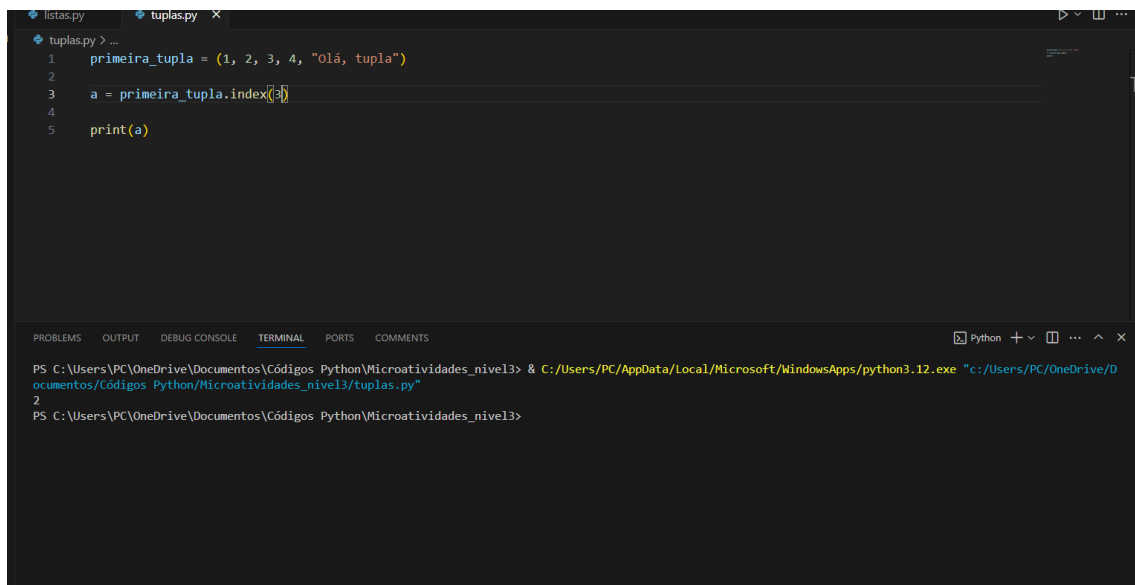
The screenshot shows the same Python IDE with the 'tuplas.py' tab active. The code now includes an additional line to find the index of the value 4:

```
1 primeira_tupla = (1, 2, 3, 4, "Olá, tupla")
2
3 a = primeira_tupla.index(4)
4
5 print(a)
```

The terminal output shows the result of the index method:

```
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/tuplas.py"
3
```

M 2.3 #Verificação se a tupla tem o elemento 3.



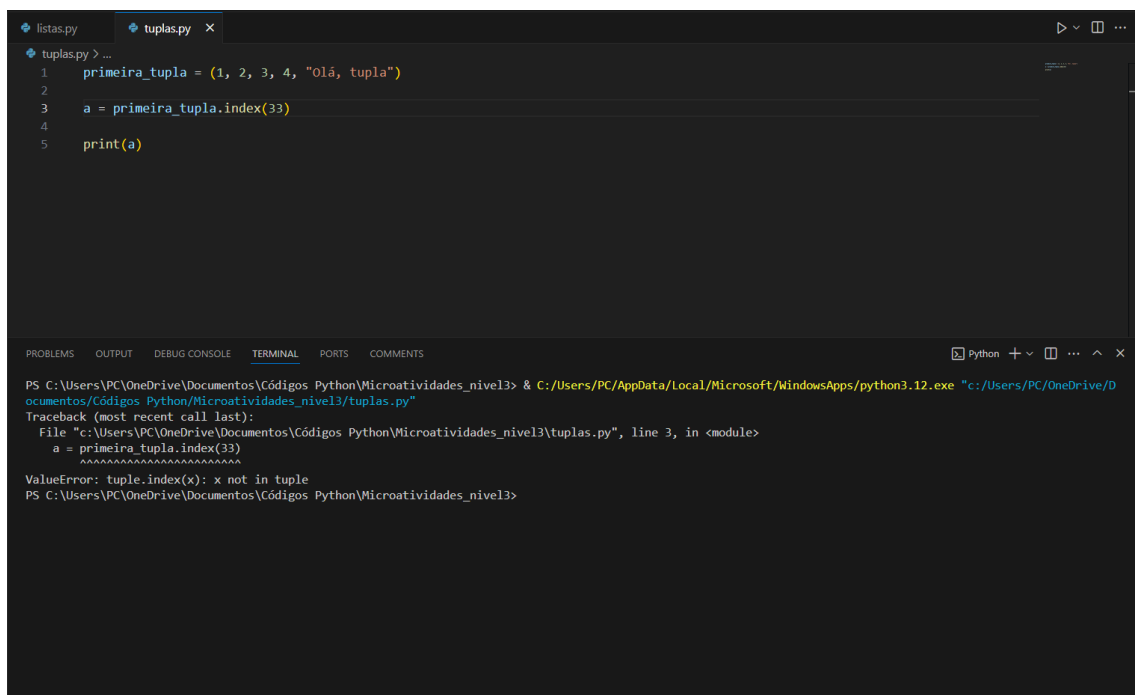
The screenshot shows a Python IDE with a file named `tuplas.py`. The code in the editor is as follows:

```
1 primeira_tupla = (1, 2, 3, 4, "Olá, tupla")
2
3 a = primeira_tupla.index(3)
4
5 print(a)
```

The terminal output shows the command to run the script and the resulting output:

```
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/tuplas.py"
2
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 2.4 # Verificação se a tupla tem o elemento 33.



The screenshot shows the same Python IDE with the `tuplas.py` file. The code in the editor is as follows:

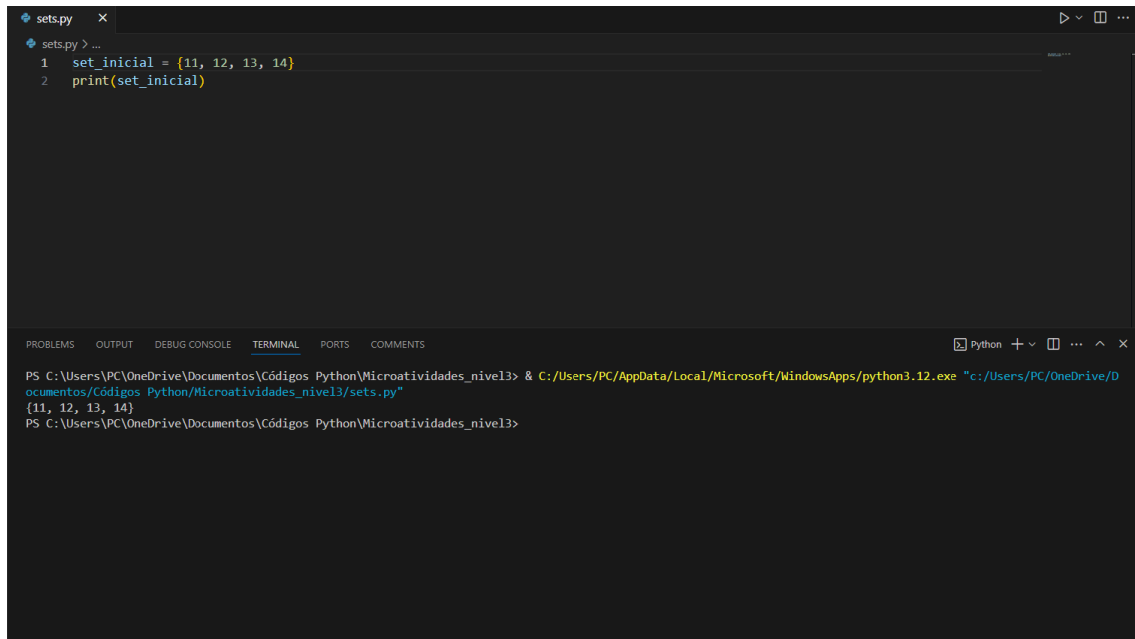
```
1 primeira_tupla = (1, 2, 3, 4, "Olá, tupla")
2
3 a = primeira_tupla.index(33)
4
5 print(a)
```

The terminal output shows the command to run the script and the resulting error:

```
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/tuplas.py"
Traceback (most recent call last):
  File "c:/Users/PC/OneDrive\Documentos\Códigos Python\Microatividades_nivel3\tuplas.py", line 3, in <module>
    a = primeira_tupla.index(33)
      ~~~~~^^^^^^^^^^^^^^^^^^^^
ValueError: tuple.index(x): x not in tuple
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

Microatividade-3: Descrever a manipulação da estrutura de dados set em Python.

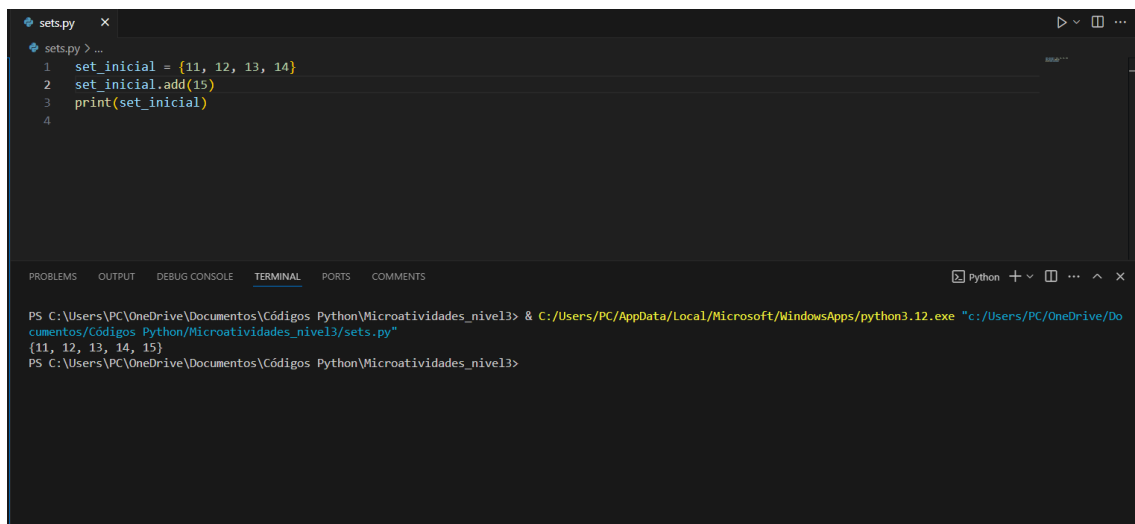
M 3.1 #Criação da estrutura de dados do tipo “set” e atribuindo valores a ela.



The screenshot shows a code editor with a file named 'sets.py'. The code defines a set 'set_inicial' with the values {11, 12, 13, 14} and prints it. The terminal output shows the execution of the script, resulting in the set {11, 12, 13, 14} being displayed.

```
sets.py > ...  
1 set_inicial = {11, 12, 13, 14}  
2 print(set_inicial)  
  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS  
Python + - ... ^ x  
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/sets.py"  
{11, 12, 13, 14}  
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 3.2 #Com o método “add”, adicione o elemento 15 ao set.



The screenshot shows the same code editor as before, but now the code adds the value 15 to the set 'set_inicial' using the 'add' method and prints the updated set. The terminal output shows the execution of the script, resulting in the set {11, 12, 13, 14, 15} being displayed.

```
sets.py > ...  
1 set_inicial = {11, 12, 13, 14}  
2 set_inicial.add(15)  
3 print(set_inicial)  
4  
  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS  
Python + - ... ^ x  
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/sets.py"  
{11, 12, 13, 14, 15}  
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 3.3 #Com o método “update”, atualize o set.

```
sets.py > ...
1 set_inicial = {11, 12, 13, 14}
2 set_inicial.add(15)
3 set_inicial.update([1,2,3,4,5])
4 print(set_inicial)
5
```

```
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/sets.py"
{1, 2, 3, 4, 5, 11, 12, 13, 14, 15}
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 3.4 #Com o método “discard”, o elemento 13 do set foi removido.

```
sets.py > ...
1 set_inicial = {11, 12, 13, 14}
2 set_inicial.add(15)
3 set_inicial.update([1,2,3,4,5])
4 set_inicial.discard(13)
5 print(set_inicial)
6
```

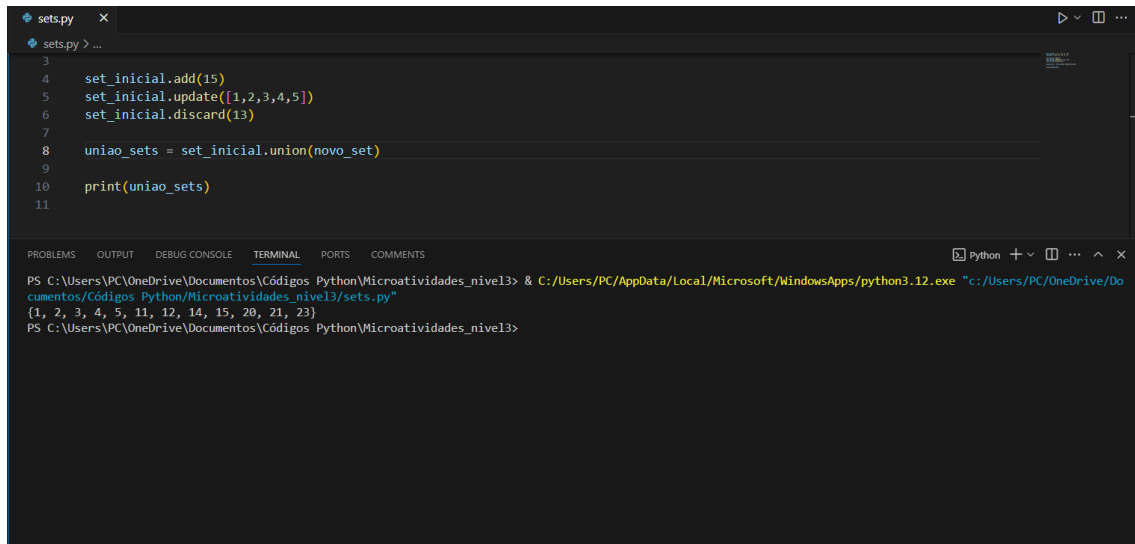
```
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/sets.py"
{1, 2, 3, 4, 5, 11, 12, 13, 14, 15}
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/sets.py"
{1, 2, 3, 4, 5, 11, 12, 14, 15}
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 3.5 # Criação de um novo set e atribuindo ao mesmo os elementos.

```
sets.py > ...
1 set_inicial = {11, 12, 13, 14}
2 set_inicial.add(15)
3 set_inicial.update([1,2,3,4,5])
4 set_inicial.discard(13)
5 novo_set = {20, 21, 23, 1, 2}
6
7 print(novo_set)
8
```

```
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/sets.py"
{1, 2, 3, 4, 5, 11, 12, 13, 14, 15}
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/sets.py"
{1, 2, 3, 4, 5, 11, 12, 14, 15}
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/sets.py"
{1, 2, 20, 21, 23}
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

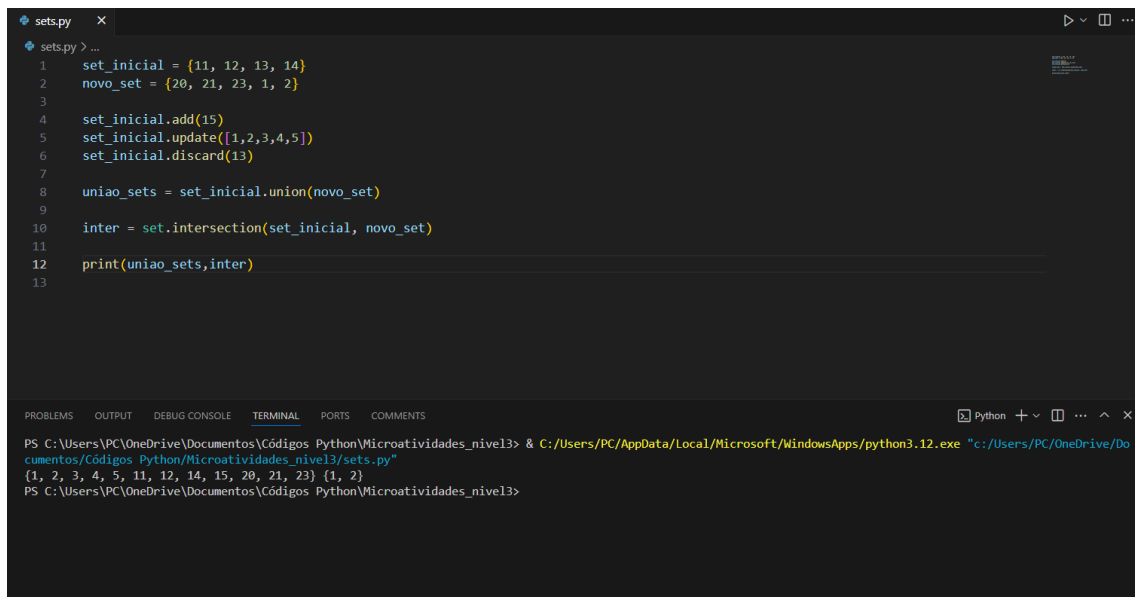
M 3.6 #União entre os dois sets criados



```
sets.py X
sets.py > ...
3
4     set_inicial.add(15)
5     set_inicial.update([1,2,3,4,5])
6     set_inicial.discard(13)
7
8     uniao_sets = set_inicial.union(novo_set)
9
10    print(uniao_sets)
11

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Python + - [] ... ^ x
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Do
cumentos/Códigos Python/Microatividades_nivel3/sets.py"
{1, 2, 3, 4, 5, 11, 12, 14, 15, 20, 21, 23}
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 3.7 #Resultado da interação entre os dois sets.



```
sets.py X
sets.py > ...
1     set_inicial = {11, 12, 13, 14}
2     novo_set = {20, 21, 23, 1, 2}
3
4     set_inicial.add(15)
5     set_inicial.update([1,2,3,4,5])
6     set_inicial.discard(13)
7
8     uniao_sets = set_inicial.union(novo_set)
9
10    inter = set.intersection(set_inicial, novo_set)
11
12    print(uniao_sets,inter)
13

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Python + - [] ... ^ x
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Do
cumentos/Códigos Python/Microatividades_nivel3/sets.py"
{1, 2, 3, 4, 5, 11, 12, 14, 15, 20, 21, 23} {1, 2}
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```


M 3.8 #Resultado da diferença entre os dois sets.

```
sets.py X
sets.py > ...
1 set_inicial = {11, 12, 13, 14}
2 novo_set = {20, 21, 23, 1, 2}
3
4 set_inicial.add(15)
5 set_inicial.update([1,2,3,4,5])
6 set_inicial.discard(13)
7
8 uniao_sets = set_inicial.union(novo_set)
9
10 inter = set.intersection(set_inicial, novo_set)
11 diferenca = set.difference(set_inicial, novo_set)
12
13 print(uniao_sets,diferenca)
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Python + - ... ^ x

PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/sets.py"
{1, 2, 3, 4, 5, 11, 12, 14, 15, 20, 21, 23} {1, 2}
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/sets.py"
{1, 2, 3, 4, 5, 11, 12, 14, 15, 20, 21, 23} {3, 4, 5, 11, 12, 14, 15}
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>

M 3.9 #Resultado da diferença simétrica entre os dois sets.

```
sets.py X
sets.py (variable) novo_set: set[int]
1
2 novo_set = {20, 21, 23, 1, 2}
3
4 set_inicial.add(15)
5 set_inicial.update([1,2,3,4,5])
6 set_inicial.discard(13)
7
8 uniao_sets = set_inicial.union(novo_set)
9
10 inter = set.intersection(set_inicial, novo_set)
11 diferenca = set.difference(set_inicial, novo_set)
12 dif_simetrica = set.symmetric_difference(set_inicial, novo_set)
13
14 print(uniao_sets,dif_simetrica)
15
```

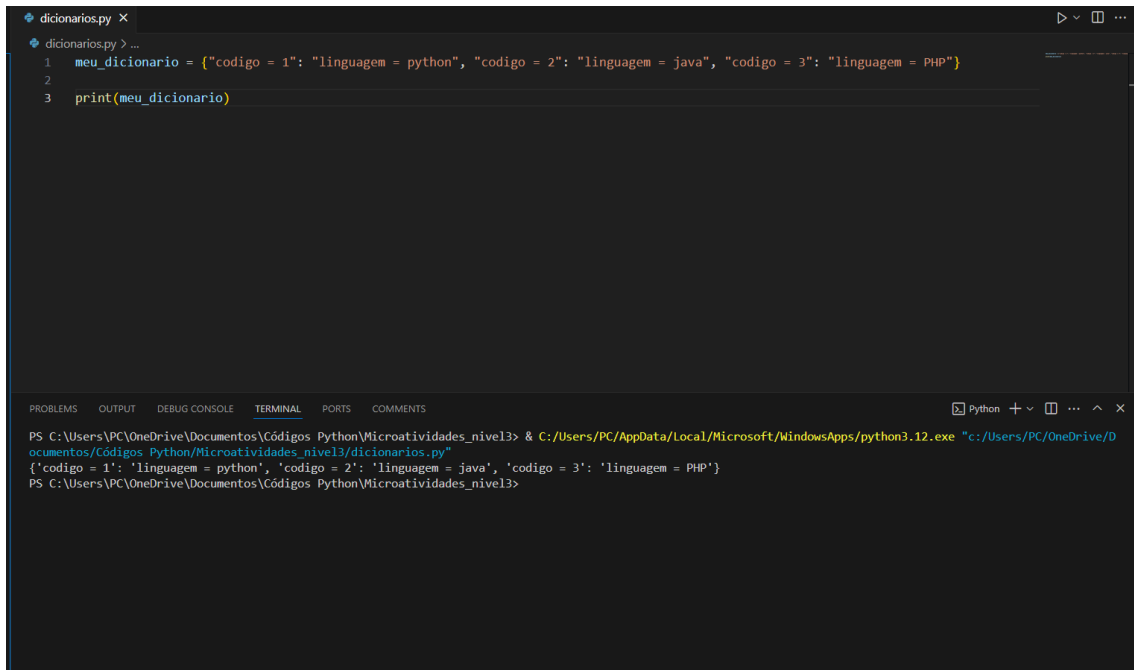
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Python + - ... ^ x

PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/sets.py"
{1, 2, 3, 4, 5, 11, 12, 14, 15, 20, 21, 23} {3, 4, 5, 11, 12, 14, 15, 20, 21, 23}
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>

Microatividade 4: Descrever a criação da estrutura de dados dicionário em Python.

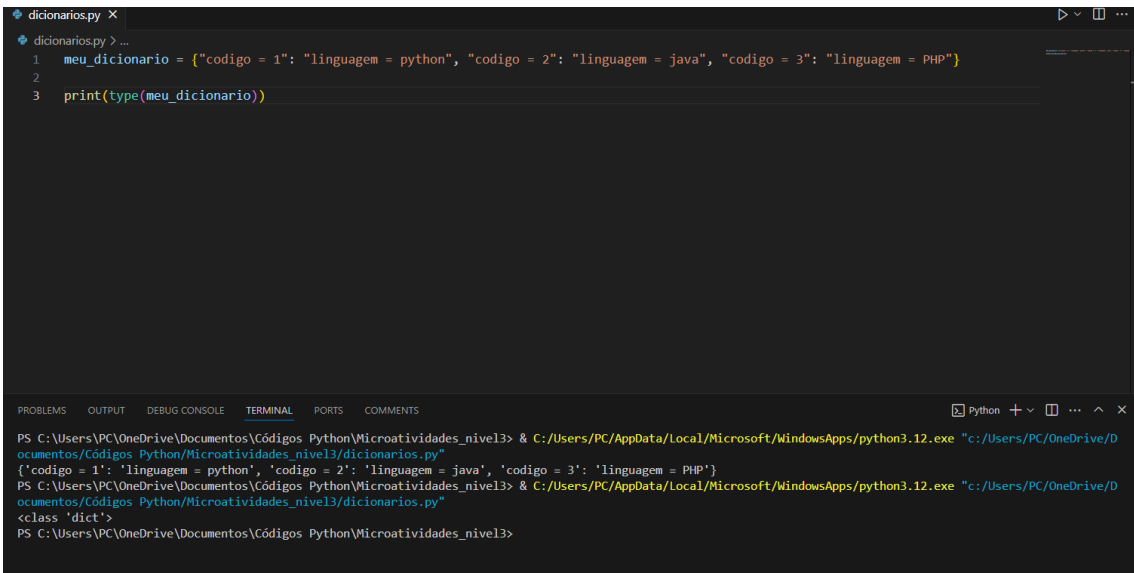
M 4.1 #Criando uma estrutura de dados tipo dicionário e atribuindo pares de chave/valor.



```
dicionarios.py X
dicionarios.py > ...
1 meu_dicionario = {"codigo = 1": "linguagem = python", "codigo = 2": "linguagem = java", "codigo = 3": "linguagem = PHP"}
2
3 print(meu_dicionario)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Python + - [] ... ^ x
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/D
ocumentos/Códigos Python/Microatividades_nivel3/dicionarios.py"
{'codigo = 1': 'linguagem = python', 'codigo = 2': 'linguagem = java', 'codigo = 3': 'linguagem = PHP'}
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

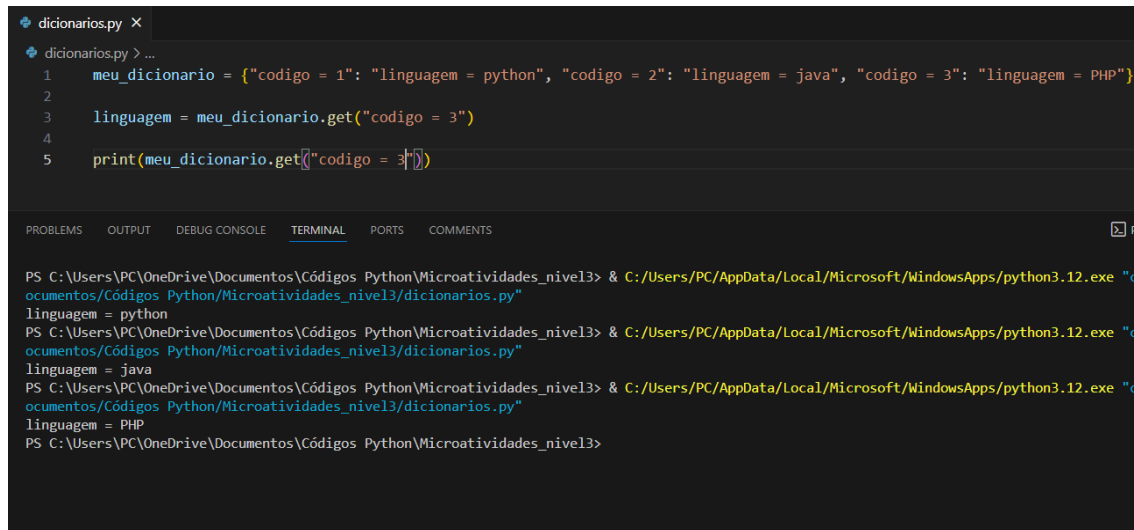
M 4.2 #Utilizando o método “type”.



```
dicionarios.py X
dicionarios.py > ...
1 meu_dicionario = {"codigo = 1": "linguagem = python", "codigo = 2": "linguagem = java", "codigo = 3": "linguagem = PHP"}
2
3 print(type(meu_dicionario))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Python + - [] ... ^ x
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/D
ocumentos/Códigos Python/Microatividades_nivel3/dicionarios.py"
{'codigo = 1': 'linguagem = python', 'codigo = 2': 'linguagem = java', 'codigo = 3': 'linguagem = PHP'}
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/D
ocumentos/Códigos Python/Microatividades_nivel3/dicionarios.py"
<class 'dict'>
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 4.3 #Utilizando o método get para obter o vaalor da linguagem.



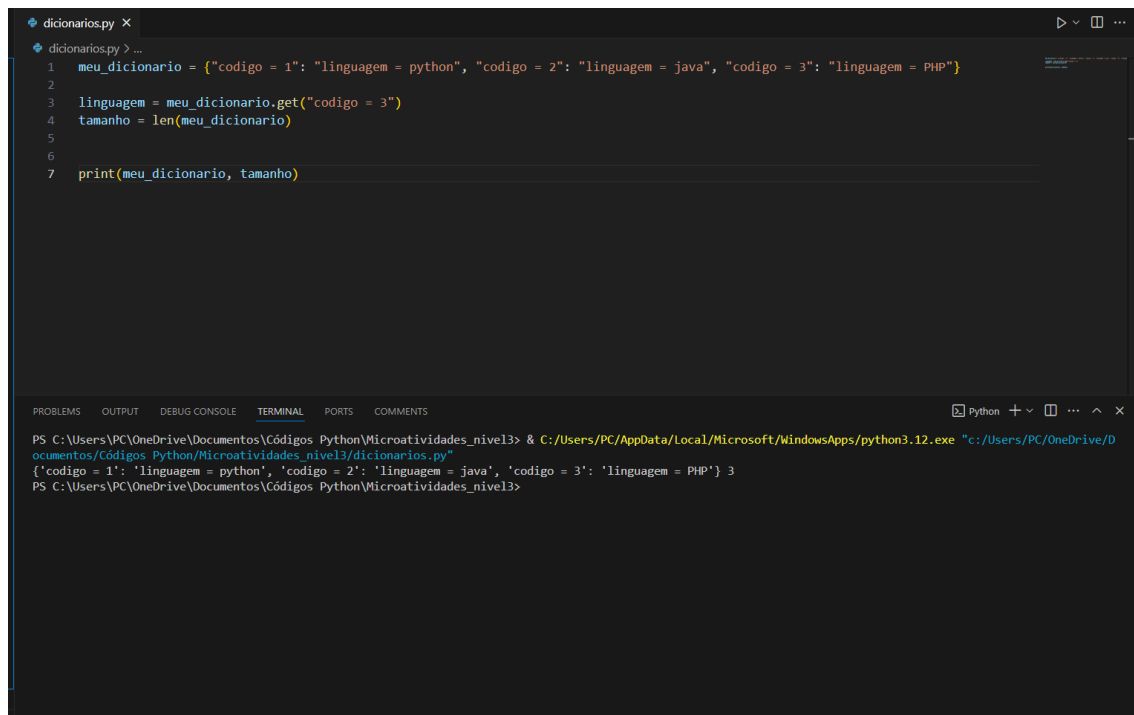
The screenshot shows a code editor with a file named `dicionarios.py`. The code defines a dictionary `meu_dicionario` with three entries: `codigo = 1` for `python`, `codigo = 2` for `java`, and `codigo = 3` for `PHP`. It then uses the `get` method to retrieve the value for `codigo = 3` and prints it.

```
1 meu_dicionario = {"codigo = 1": "linguagem = python", "codigo = 2": "linguagem = java", "codigo = 3": "linguagem = PHP"}
2
3 linguagem = meu_dicionario.get("codigo = 3")
4
5 print(meu_dicionario.get("codigo = 3"))
```

The terminal output shows the execution of the script, confirming that the value for `codigo = 3` is `PHP`.

```
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3\dicionarios.py"
linguagem = python
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3\dicionarios.py"
linguagem = java
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3\dicionarios.py"
linguagem = PHP
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 4.4 #Tamanho do dicionário.



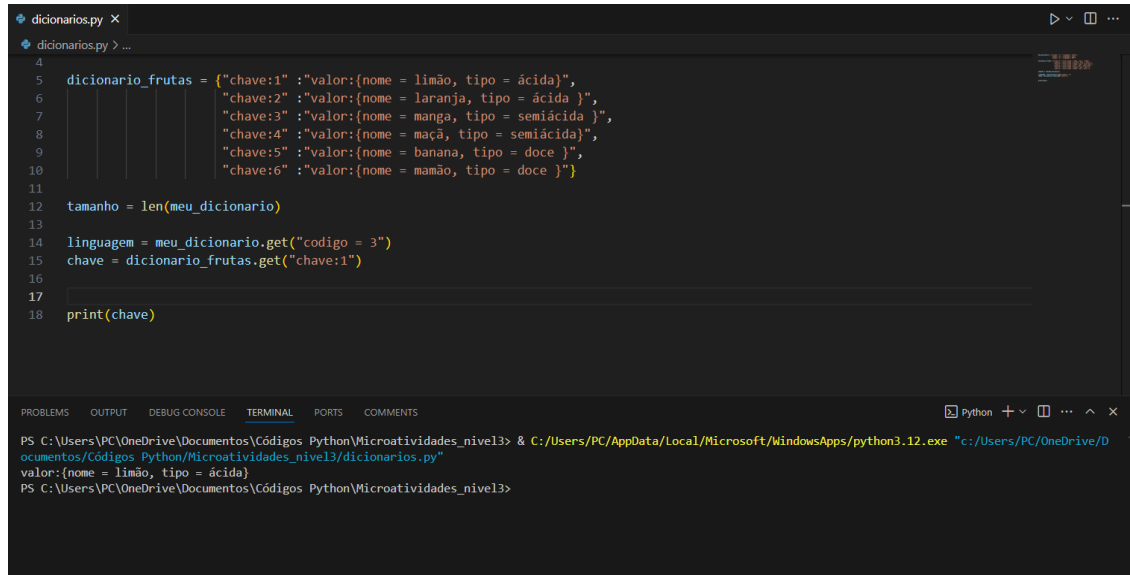
The screenshot shows a code editor with a file named `dicionarios.py`. The code defines a dictionary `meu_dicionario` with three entries: `codigo = 1` for `python`, `codigo = 2` for `java`, and `codigo = 3` for `PHP`. It then uses the `len` function to get the size of the dictionary and prints it.

```
1 meu_dicionario = {"codigo = 1": "linguagem = python", "codigo = 2": "linguagem = java", "codigo = 3": "linguagem = PHP"}
2
3 linguagem = meu_dicionario.get("codigo = 3")
4 tamanho = len(meau_dicionario)
5
6
7 print(meu_dicionario, tamanho)
```

The terminal output shows the execution of the script, displaying the dictionary and its size (3).

```
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3\dicionarios.py"
{'codigo = 1': 'linguagem = python', 'codigo = 2': 'linguagem = java', 'codigo = 3': 'linguagem = PHP'} 3
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 4.5 #Criando um novo dicionário.

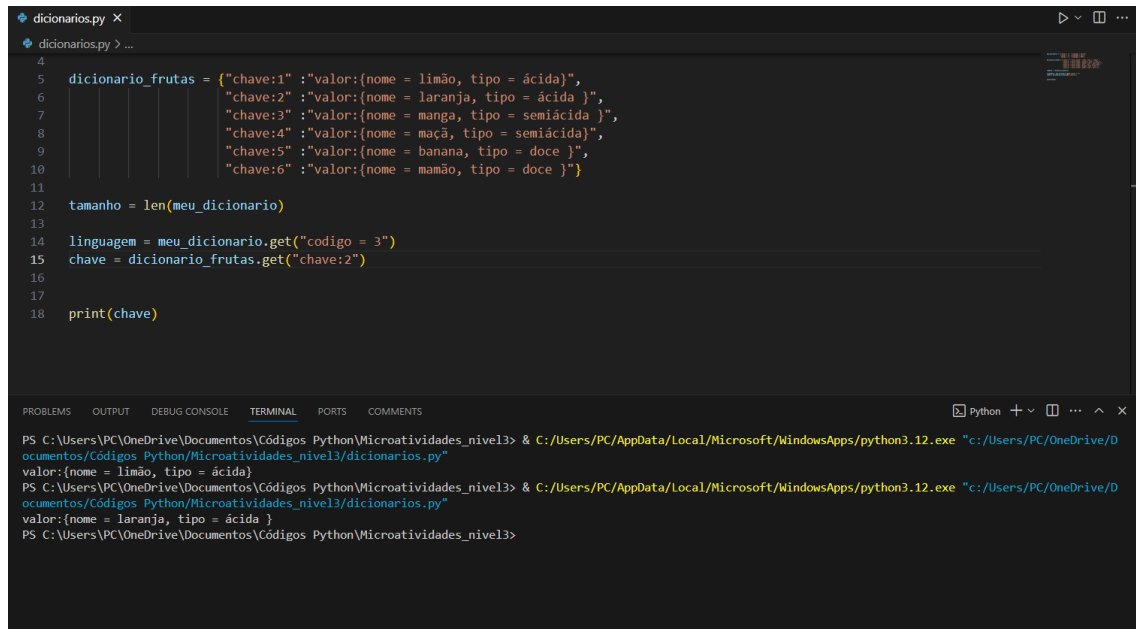


```
dicionarios.py X
dicionarios.py > ...
4
5  dicionario_frutas = {"chave:1" : "valor:{nome = limão, tipo = ácida}",
6                        "chave:2" : "valor:{nome = laranja, tipo = ácida }",
7                        "chave:3" : "valor:{nome = manga, tipo = semiácida }",
8                        "chave:4" : "valor:{nome = maçã, tipo = semiácida}",
9                        "chave:5" : "valor:{nome = banana, tipo = doce }",
10                       "chave:6" : "valor:{nome = mamão, tipo = doce }"}
11
12  tamanho = len(meu_dicionario)
13
14  linguagem = meu_dicionario.get("codigo = 3")
15  chave = dicionario_frutas.get("chave:1")
16
17
18  print(chave)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
Python + - □ ... ^ x

PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/D
ocumentos/Códigos Python/Microatividades_nivel3/dicionarios.py"
valor:{nome = limão, tipo = ácida}
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 4.6 #Valor da chave:1 e 2 “nome” e “tipo”.



```
dicionarios.py X
dicionarios.py > ...
4
5  dicionario_frutas = {"chave:1" : "valor:{nome = limão, tipo = ácida}",
6                        "chave:2" : "valor:{nome = laranja, tipo = ácida }",
7                        "chave:3" : "valor:{nome = manga, tipo = semiácida }",
8                        "chave:4" : "valor:{nome = maçã, tipo = semiácida}",
9                        "chave:5" : "valor:{nome = banana, tipo = doce }",
10                       "chave:6" : "valor:{nome = mamão, tipo = doce }"}
11
12  tamanho = len(meu_dicionario)
13
14  linguagem = meu_dicionario.get("codigo = 3")
15  chave = dicionario_frutas.get("chave:2")
16
17
18  print(chave)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
Python + - □ ... ^ x

PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/D
ocumentos/Códigos Python/Microatividades_nivel3/dicionarios.py"
valor:{nome = limão, tipo = ácida}
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/D
ocumentos/Códigos Python/Microatividades_nivel3/dicionarios.py"
valor:{nome = laranja, tipo = ácida }
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 4.7 #Utilizando o laço for, imprimir todos os valores de chaves “nome” e “tipo”.

```
diccionarios.py x
diccionarios.py > ...
3         "codigo = 3": "linguagem = PHP"}
4
5     dicionario_frutas = {"chave:1" : "valor:{nome = limão, tipo = ácida}",
6                          "chave:2" : "valor:{nome = laranja, tipo = ácida }",
7                          "chave:3" : "valor:{nome = manga, tipo = semiácida }",
8                          "chave:4" : "valor:{nome = maçã, tipo = semiácida}",
9                          "chave:5" : "valor:{nome = banana, tipo = doce }",
10                         "chave:6" : "valor:{nome = mamão, tipo = doce }"}
11
12     tamanho = len(meu_dicionario)
13
14     linguagem = meu_dicionario.get("codigo = 3")
15     chave = dicionario_frutas.get("chave:2")
16     valores = dicionario_frutas.values()
17
18     for valores in dicionario_frutas.values():
19         print(valores)
20
21
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
Python + - □ ... ^ x

PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/diccionarios.py"
valor:{nome = limão, tipo = ácida}
valor:{nome = laranja, tipo = ácida }
valor:{nome = manga, tipo = semiácida }
valor:{nome = maçã, tipo = semiácida}
valor:{nome = banana, tipo = doce }
valor:{nome = mamão, tipo = doce }
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

Microatividade 5: Descrever a atualização de dados em um dicionário

M 5.1

```
diccionarios2.py •
diccionarios2.py > ...
1     dicionario_padrao = {"nome": "Maria", "idade": 26, "nacionalidade": "brasileira"}
2     novos_elementos = {"cidade": "recife", "estado": "pernambuco-PE"}
3
4     dicionario_padrao.update(novos_elementos)
5
6     print(dicionario_padrao)
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
Python + - □ ... ^ x

PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/diccionarios2.py"
{'nome': 'Maria', 'idade': 26, 'nacionalidade': 'brasileira', 'cidade': 'recife', 'estado': 'pernambuco-PE'}
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 5.2

```
dicionarios2.py X
1  dicionario_padrao = {"nome": "Maria", "idade": 26, "nacionalidade": "brasileira"}
2  novos_elementos = {"cidade": "recife", "estado": "pernambuco-PE"}
3
4  dicionario_padrao.update(novos_elementos)
5  novo_dicionario = dicionario_padrao.copy()
6  remover_dic_padrao = dicionario_padrao[1].pop("nacionalidade")
7
8
9  print(remover_dic_padrao)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/dicionarios2.py"
brasileira
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 5.3

```
dicionarios2.py X
1  dicionario_padrao = {"nome": "Maria", "idade": 26, "nacionalidade": "brasileira"}
2  novos_elementos = {"cidade": "recife", "estado": "pernambuco-PE"}
3
4  dicionario_padrao.update(novos_elementos)
5  novo_dicionario = dicionario_padrao.copy()
6  remover_dic_padrao = dicionario_padrao[1].pop("nacionalidade")
7  remove_item_dic_padrao = dicionario_padrao[1].popitem()
8
9
10 print(dicionario_padrao)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/dicionarios2.py"
{'nome': 'Maria', 'cidade': 'recife', 'estado': 'pernambuco-PE'}
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```

M 5.4

```
dicionarios2.py X
1 dicionario_padrao = {"nome": "Maria", "idade": 26, "nacionalidade": "brasileira"}
2 novos_elementos = {"cidade": "recife", "estado": "pernambuco-PE"}
3
4 dicionario_padrao.update(novos_elementos)
5 novo_dicionario = dicionario_padrao.copy()
6 remover_dic_padrao = dicionario_padrao.pop("nacionalidade")
7 remove_item_dic_padrao = dicionario_padrao.popitem()
8 limpando_dicionario_padrao = dicionario_padrao.clear()
9 limpando_novo_dicionario = novo_dicionario.clear()
10
11 print(dicionario_padrao, novo_dicionario)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Python + ...

PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>

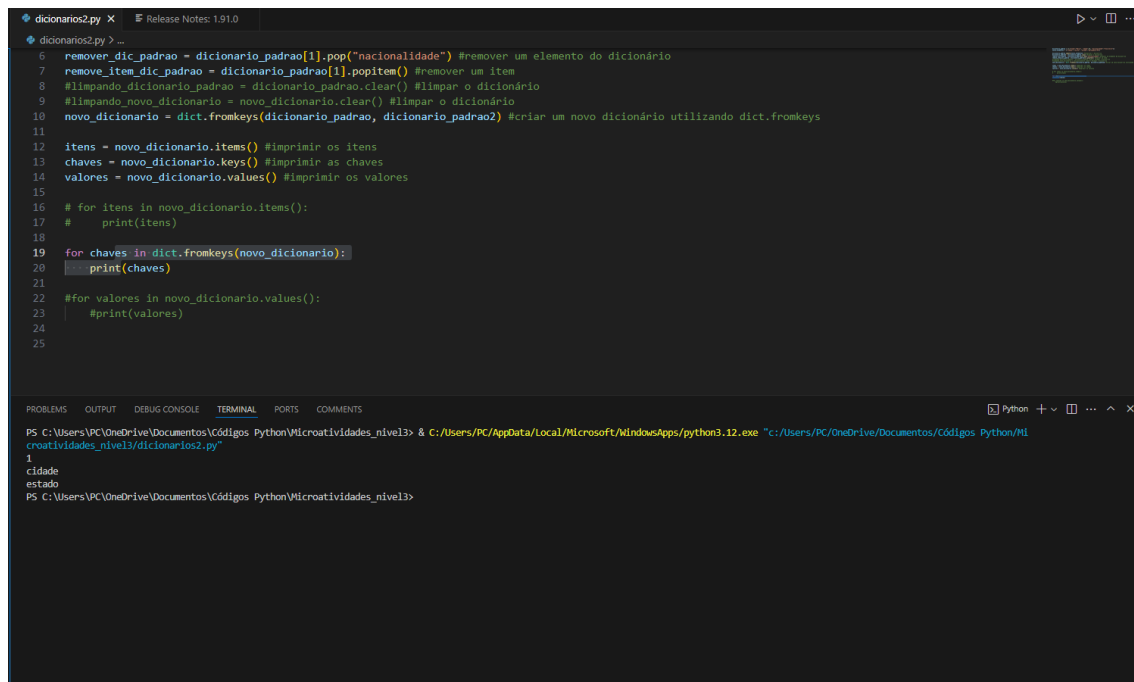
M 5.5

```
dicionarios2.py X Release Notes: 1.91.0
dicionarios2.py > ...
6 remover_dic_padrao = dicionario_padrao[1].pop("nacionalidade") #remover um elemento do dicionário
7 remove_item_dic_padrao = dicionario_padrao[1].popitem() #remover um item
8 limpando_dicionario_padrao = dicionario_padrao.clear() #limpar o dicionário
9 limpando_novo_dicionario = novo_dicionario.clear() #limpar o dicionário
10 novo_dicionario = dict.fromkeys(dicionario_padrao, dicionario_padrao2) #criar um novo dicionário utilizando dict.fromkeys
11
12 itens = novo_dicionario.items() #imprimir os itens
13 chaves = novo_dicionario.keys() #imprimir as chaves
14 valores = novo_dicionario.values() #imprimir os valores
15
16 for itens in novo_dicionario.items():
17     print(itens)
18
19 #for chaves in dict.fromkeys(novo_dicionario):
20     #print(chaves)
21
22 #for valores in novo_dicionario.values():
23     #print(valores)
24
25
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS Python + - ... ^ v X

```
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/diccionarios2.py"
(1, {1: {'nome': 'Maria'}, 'cidade': 'recife', 'estado': 'pernambuco-PE'})
{'cidade': {1: {'nome': 'Maria'}, 'cidade': 'recife', 'estado': 'pernambuco-PE'}}
{'estado': {1: {'nome': 'Maria'}, 'cidade': 'recife', 'estado': 'pernambuco-PE'}}
```

M 5.6



```
diccionario2.py x Release Notes: 1.91.0
diccionario2.py > --
6 remove_dic_padrao = diccionario_padrao[1].pop("nacionalidade") #remover um elemento do dicionário
7 remove_item_dic_padrao = diccionario_padrao[1].popitem() #remover um item
8 limpando_diccionario_padrao = diccionario_padrao.clear() #limpar o dicionário
9 limpando_novo_diccionario = novo_diccionario.clear() #limpar o dicionário
10 novo_diccionario = dict.fromkeys(diccionario_padrao, diccionario_padrao2) #criar um novo dicionário utilizando dict.fromkeys
11
12 itens = novo_diccionario.items() #imprimir os itens
13 chaves = novo_diccionario.keys() #imprimir as chaves
14 valores = novo_diccionario.values() #imprimir os valores
15
16 # for itens in novo_diccionario.items():
17 #     print(itens)
18
19 for chaves in dict.fromkeys(novo_diccionario):
20     print(chaves)
21
22 #for valores in novo_diccionario.values():
23 |     #print(valores)
24
25

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Python + - - - ^ x
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/PC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/diccionario2.py"
1
cidade
estado
PS C:\Users\PC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3>
```


Missão Prática | Estruturando os Dados

operações.py

```
1  # Missão prática
2  def calcular_media(notas):
3      return sum(notas) / len(notas)
4
5
6  # Crie um método, que receberá o valor da média e retornará um
7  # booleano, para verificar se o aluno foi reprovado. Será considerado
8  # como reprovado um aluno com média inferior a 6;
9
10 def verificar_reprovado(media):
11     if media < 6:
12         return True
13     else:
14         return False
15
16 # Crie um método, que receberá dois parâmetros - os dados dos
17 # alunos e os números das matrículas dos alunos reprovados - e
18 # retornará a seguinte saída:
19 # 'Aluno Reprovado: NOME DO ALUNO - Matrícula: MATRÍCULA
20 # DO ALUNO - Média Final: MÉDIA FINAL DO ALUNO'
21 def dados_dos_alunos(alunos, alunos_reprovados):
22     for aluno in alunos:
23         if aluno['media'] in alunos_reprovados:
24             print(f"Aluno Reprovado: {aluno['nome']} - Matrícula: {aluno['matricula']} - Média Final: {aluno['media']}")
25
```

main.py

```
1 from operacoes import calcular_media, verificar_reprovado, dados_dos_alunos
2
3 #Crie as estruturas de dados necessárias para armazenar os dados
4 #dos alunos e as notas
5 dados_alunos = [
6     {"nome": "Maria", "matricula": "26", "notas": [8, 7, 5, 9]},
7     {"nome": "Ana", "matricula": "101", "notas": [9, 9, 8, 9]},
8     {"nome": "João", "matricula": "13", "notas": [6, 5, 5, 5]},
9     {"nome": "Ágatha", "matricula": "37", "notas": [8, 6, 7.5, 9]},
10    {"nome": "Joaquim", "matricula": "72", "notas": [6, 5.5, 5, 7]},
11    {"nome": "Félix", "matricula": "5", "notas": [10, 8, 8, 8]}
12 ]
13
14 #calcular a média de cada aluno
15 for aluno in dados_alunos:
16     media = calcular_media(aluno['notas'])
17     aluno['media'] = media
18     print(f"Aluno: {aluno['nome']} - Matrícula: {aluno['matricula']} - Média Final: {aluno['media']}")
19
20 #verificar se o aluno foi reprovado;
21 for aluno in dados_alunos:
22     reprovado = verificar_reprovado(aluno['media'])
23     print(verificar_reprovado(aluno['media']))
24
25 #identificar qual aluno foi reprovado;
26 for aluno in dados_alunos:
27     if verificar_reprovado(aluno['media']):
28         print(dados_dos_alunos(dados_alunos, [aluno['media']]))
```

Resposta no Terminal

```
PS C:\Users\FC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> & C:/Users/FC/AppData/Local/Microsoft/WindowsApps/python3.12.exe "C:/Users/FC/OneDrive/Documentos/Códigos Python/Microatividades_nivel3/main.py"
Aluno: Maria - Matrícula: 26 - Média Final: 7.25
Aluno: Ana - Matrícula: 101 - Média Final: 8.75
Aluno: João - Matrícula: 13 - Média Final: 5.25
Aluno: Ágatha - Matrícula: 37 - Média Final: 7.625
Aluno: Joaquim - Matrícula: 72 - Média Final: 5.875
Aluno: Félix - Matrícula: 5 - Média Final: 8.5
False
False
True
False
True
False
Aluno Reprovado: João - Matrícula: 13 - Média Final: 5.25
None
Aluno Reprovado: Joaquim - Matrícula: 72 - Média Final: 5.875
None
PS C:\Users\FC\OneDrive\Documentos\Códigos Python\Microatividades_nivel3> 
```