

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

```
File Edit Selection View Go Run Terminal Help ⏪ ⏴ TRABALHO - FULL STACK 3
EXPLORER listas.py
listas.py
1 lista_mesclada = [1,2,3,"Olá Python",True,12.6]
2 print(lista_mesclada)
3 lista_mesclada.append("Lista aninhada")
4 print(lista_mesclada)
5 lista_mesclada.insert(3,5)
6 print(lista_mesclada)
7 print(len(lista_mesclada))
8 lista_mesclada.remove(1)
9 print(lista_mesclada)
10 nova_lista_mesclada = lista_mesclada
11 nova_lista_mesclada.pop(0)
12 nova_lista_mesclada.pop(5)
13 nova_lista_mesclada.pop(4)
14 print(nova_lista_mesclada)

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
listas.py*
[1, 2, 3, 'Olá Python', True, 12.6]
[1, 2, 3, 'Olá Python', True, 12.6, ['Lista aninhada']]
[1, 2, 3, 5, 'Olá Python', True, 12.6, ['Lista aninhada']]
[2, 3, 5, 'Olá Python']
PS C:\Users\lucas\OneDrive\Área de Trabalho\TRABALHO - FULL STACK 3>
Ln 11, Col 26 Spaces: 4 UTF-8 CRLF Python 3.12.4 64-bit
```

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

```
File Edit Selection View Go Run Terminal Help ⏪ ⏴ TRABALHO - FULL STACK 3
EXPLORER listas.py tuplas.py
tuplas.py
1 primeira_tupla = 1, 2, 3, 4, "Olá, tupla"
2 print(primeira_tupla)
3 indice_4 = primeira_tupla.index('Olá, tupla')
4 print(indice_4)
5 print(3 in primeira_tupla)
6 print([3] in primeira_tupla)

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
PS C:\Users\lucas\OneDrive\Área de Trabalho\TRABALHO - FULL STACK 3> & C:/Users/lucas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/lucas/OneDrive/Área de Trabalho/TRABALHO - FULL STACK 3/tuplas.py"
(1, 2, 3, 4, 'Olá, tupla')
4
True
False
PS C:\Users\lucas\OneDrive\Área de Trabalho\TRABALHO - FULL STACK 3>
Ln 6, Col 9 Spaces: 4 UTF-8 CRLF Python 3.12.4 64-bit
```

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

```

File Edit Selection View Go Run Terminal Help ← → TRABALHO - FULL STACK 3
EXPLORER ... sets.py > ...
TRABALHO - FULL STACK 3
listas.py tuples.py sets.py
1 set_inicial = {11, 12, 13, 14}
2 print("set_inicial é", set_inicial)
3 set_inicial.add(15)
4 print("set_inicial.add é", set_inicial)
5 x = {1,2,3,4,5}
6 set_inicial.update(x)
7 print("set_inicial.update é", set_inicial)
8 set_inicial.discard(13)
9 print("set_inicial.discard é", set_inicial)
10 novo_set = {20,21,23,1,2}
11 print("novo_set é", novo_set)
12 uniao = set_inicial.union(novo_set)
13 print("novo_set.union é", uniao)
14 intersecao = set_inicial.intersection(novo_set)
15 print("novo_set.intersecao é", intersecao)
16 diferenca = set_inicial.difference(novo_set)
17 print("novo_set.diferenca é", diferenca)
18 diferenca_simetrica = set_inicial.symmetric_difference(novo_set)
19 print("novo_set.symmetric_difference é", diferenca_simetrica)

```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

```

PS C:\Users\lucas\OneDrive\Área de Trabalho\TRABALHO - FULL STACK 3> & C:/Users/lucas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/lucas/OneDrive/Área de Trabalho/sets.py"
set_inicial é {11, 12, 13, 14}
set_inicial.add é {11, 12, 13, 14, 15}
set_inicial.update é {1, 2, 3, 4, 5, 11, 12, 13, 14, 15}
set_inicial.discard é {1, 2, 3, 4, 5, 11, 12, 14, 15}
novo_set é {1, 2, 20, 21, 23}
novo_set.union é {1, 2, 3, 4, 5, 11, 12, 14, 15, 20, 21, 23}
novo_set.intersecao é {1, 2}
novo_set.diferenca é {3, 4, 5, 11, 12, 14, 15}
novo_set.symmetric_difference é {3, 4, 5, 11, 12, 14, 15, 20, 21, 23}

```

Ln 19, Col 63 Spaces: 4 CR/LF: 1 Python 3.12.4 64-bit

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

```

File Edit Selection View Go Run Terminal Help ← → TRABALHO - FULL STACK 3
EXPLORER ... dicionarios.py > ...
TRABALHO - FULL STACK 3
listas.py tuples.py sets.py dicionarios.py
1 meu_dicionario = {
2     'tecnologia_1': {'codigo': 1, 'linguagem': 'Python'},
3     'tecnologia_2': {'codigo': 2, 'linguagem': 'Java'},
4     'tecnologia_3': {'codigo': 3, 'linguagem': 'PHP'}
5 }
6 print(meu_dicionario)
7 print(type(meu_dicionario))
8 for tecnologia in meu_dicionario.values():
9     print(tecnologia['linguagem'])
10 print(len(meu_dicionario))
11
12
13 dicionario_frutas = {
14     'chave_1': {'nome': 'limao', 'tipo': 'ácida'},
15     'chave_2': {'nome': 'laranja', 'tipo': 'ácida'},
16     'chave_3': {'nome': 'manga', 'tipo': 'semiacida'},
17     'chave_4': {'nome': 'maça', 'tipo': 'semiacida'},
18     'chave_5': {'nome': 'banana', 'tipo': 'doce'},
19     'chave_6': {'nome': 'mamão', 'tipo': 'doce'}
20 }
21
22 print(dicionario_frutas['chave_1']['nome'])
23 print(type(dicionario_frutas['chave_1']))
24 print(dicionario_frutas['chave_2']['nome'])
25 print(type(dicionario_frutas['chave_2']))
26
27 for frutas in dicionario_frutas.values():
28     print(frutas['nome'], frutas['tipo'])

```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

```

PS C:\Users\lucas\OneDrive\Área de Trabalho\TRABALHO - FULL STACK 3> & C:/Users/lucas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/lucas/OneDrive/Área de Trabalho/dicionarios.py"

```

```

File Edit Selection View Go Run Terminal Help ← → TRABALHO - FULL STACK 3
EXPLORER ... PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
TRABALHO - FULL STACK 3
dicionarios.py
listas.py tuples.py sets.py
PS C:\Users\lucas\OneDrive\Área de Trabalho\TRABALHO - FULL STACK 3> & C:/Users/lucas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/lucas/OneDrive/Área de Trabalho/dicionarios.py"
{'tecnologia_1': {'codigo': 1, 'linguagem': 'Python'}, 'tecnologia_2': {'codigo': 2, 'linguagem': 'Java'}, 'tecnologia_3': {'codigo': 3, 'linguagem': 'PHP'}}
python
java
php
g
limao
<class 'dict'>
laranja
<class 'dict'>
limao ácida
laranja ácida
manga semiacida
maça semiacida
banana doce
mamão doce

```

## Microatividade 5: Descrever a atualização de dados em um dicionário

The screenshot shows a Python script named `dicionarios2.py` in the VS Code editor. The code demonstrates various operations on dictionaries, including updating, copying, popping items, clearing, and creating new dictionaries from keys and values.

```

1 dicionario = {1 : {'nome' : 'Maria' , 'idade' : 26 , 'nacionalidade' : 'brasileira'}}
2 dicionario.update({'religiao' : 'catolica' , 'cor' : 'parda' , 'filhos' : 2})
3 print(dicionario)
4 copia = dicionario.copy()
5 print(copia)
6 dicionario.pop(1)
7 print(dicionario)
8 print(copia)
9 filhos = dicionario.popitem()
10 print(dicionario)
11 dicionario.clear()
12 copia.clear()
13 print(dicionario)
14 print(copia)
15 x = ('nome', 'idade', 'nacionalidade')
16 y = ('A preencher')
17 novo_dicionario = dict.fromkeys(x,y)
18 print(novo_dicionario)
19 print(novo_dicionario.items())
20 print(novo_dicionario.keys())
21 print(novo_dicionario.values())
22

```

The terminal output shows the execution of the script, displaying the resulting dictionary structure and its various states after performing different operations.

## Missão Prática |

### Estruturando os

#### Dados

The screenshot shows a Python script named `main.py` in the VS Code editor. The script imports functions from `operacoes` and `calculos` modules to calculate the average marks of students and their final results.

```

1 from operacoes import calculo_media
2 from operacoes import resultado_final
3 from operacoes import reprovado
4
5 alunos = [26 : 'maria' , 101 : 'ana' , 13 : 'João' , 37 : 'Agatha' , 72: 'Joaquim' , 5 : 'Félix']
6 maria = [8,7,5,9]
7 ana = [9,9,8,9]
8 joao = [6,5,5,5]
9 agatha = [8,6,7,5,9]
10 joaquim = [6,5,5,5,7]
11 felix = [10,8,8,8]
12
13 maria_
14 maria_ = calculo_media(maria)
15 maria_resul = resultado_final(maria_)
16 #ana
17 ana_ = calculo_media(ana)
18 ana_resul = resultado_final(ana_)
19 #joao
20 joao_ = calculo_media(joao)
21 joao_resul = resultado_final(joao_)
22 #agatha
23 agatha_ = calculo_media(agatha)
24 agatha_resul = resultado_final(agatha_)
25 #joaquim
26 joaquim_ = calculo_media(joaquim)
27 joaquim_resul = resultado_final(joaquim_)
28 #felix
29 felix_ = calculo_media(felix)
30 felix_resul = resultado_final(felix_)

```

The terminal output shows the execution of the script, displaying the calculated averages and final results for each student.

This screenshot shows a terminal window displaying the results of the script execution. It lists the names of the students and their calculated averages, indicating which students were approved based on the defined threshold.

```

PS C:\Users\lucas\OneDrive\Área de Trabalho\TRABALHO - FULL STACK 3> & c:/Users/lucas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/lucas/OneDrive/Área de Trabalho/TRABALHO - FULL STACK 3/main.py"
7.25
Aluno aprovado
8.75
Aluno aprovado
5.25
Reprovado
7.625
Aluno aprovado
5.875
Reprovado
8.5
Aluno aprovado
PS C:\Users\lucas\OneDrive\Área de Trabalho\TRABALHO - FULL STACK 3>

```

