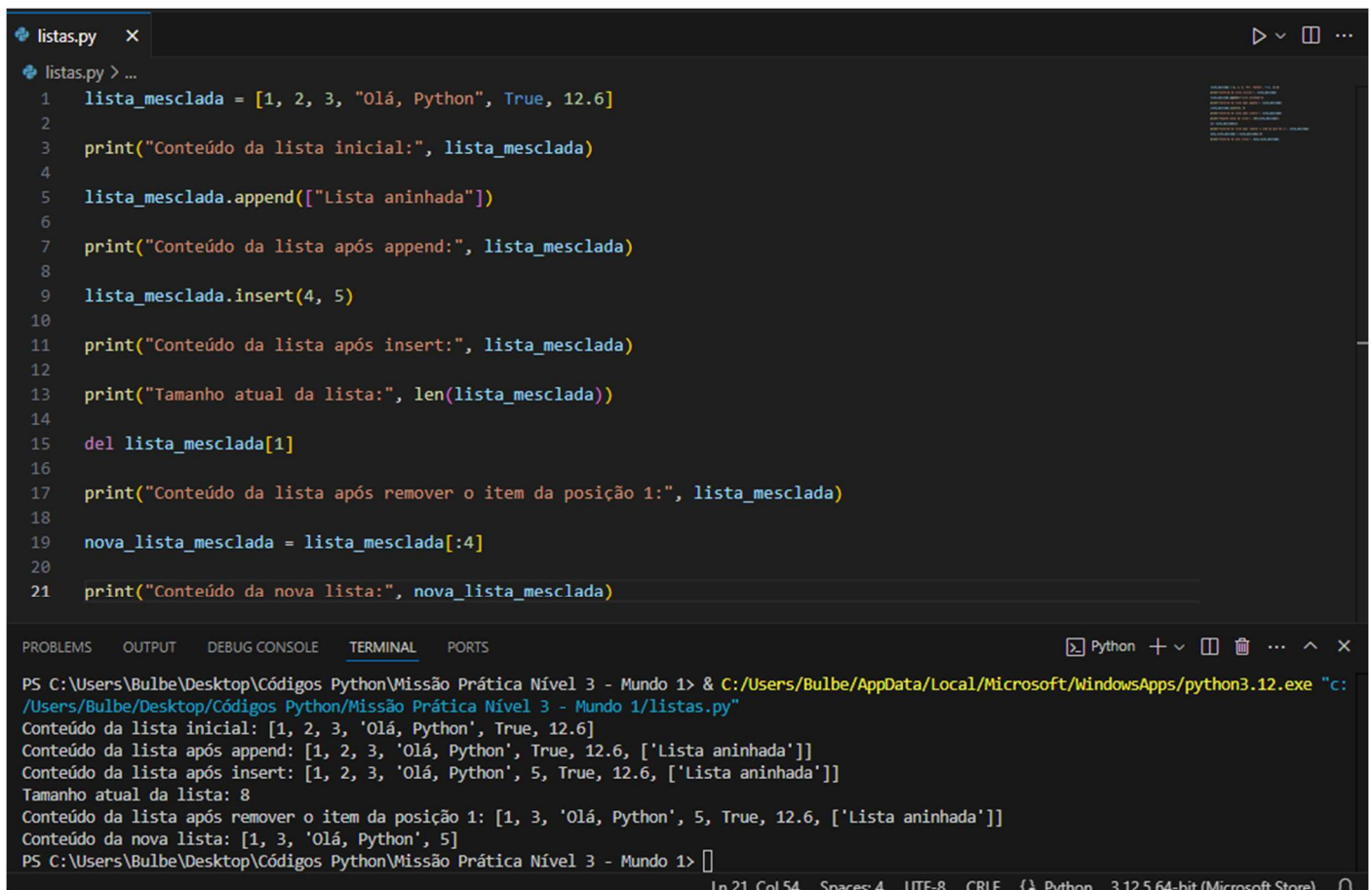


Curso: Desenvolvimento Full Stack

Missão: **Nível 3**

Aluno: Celso de Oliveira Silva Júnior

- Microatividade 1:



```
listas.py x
listas.py > ...
1  lista_mesclada = [1, 2, 3, "Olá, Python", True, 12.6]
2
3  print("Conteúdo da lista inicial:", lista_mesclada)
4
5  lista_mesclada.append(["Lista aninhada"])
6
7  print("Conteúdo da lista após append:", lista_mesclada)
8
9  lista_mesclada.insert(4, 5)
10
11 print("Conteúdo da lista após insert:", lista_mesclada)
12
13 print("Tamanho atual da lista:", len(lista_mesclada))
14
15 del lista_mesclada[1]
16
17 print("Conteúdo da lista após remover o item da posição 1:", lista_mesclada)
18
19 nova_lista_mesclada = lista_mesclada[:4]
20
21 print("Conteúdo da nova lista:", nova_lista_mesclada)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS Python + -

PS C:\Users\Bulbe\Desktop\Códigos Python\Missão Prática Nível 3 - Mundo 1> & C:/Users/Bulbe/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Bulbe/Desktop/Códigos Python/Missão Prática Nível 3 - Mundo 1/listas.py"

Conteúdo da lista inicial: [1, 2, 3, 'Olá, Python', True, 12.6]

Conteúdo da lista após append: [1, 2, 3, 'Olá, Python', True, 12.6, ['Lista aninhada']]

Conteúdo da lista após insert: [1, 2, 3, 'Olá, Python', 5, True, 12.6, ['Lista aninhada']]

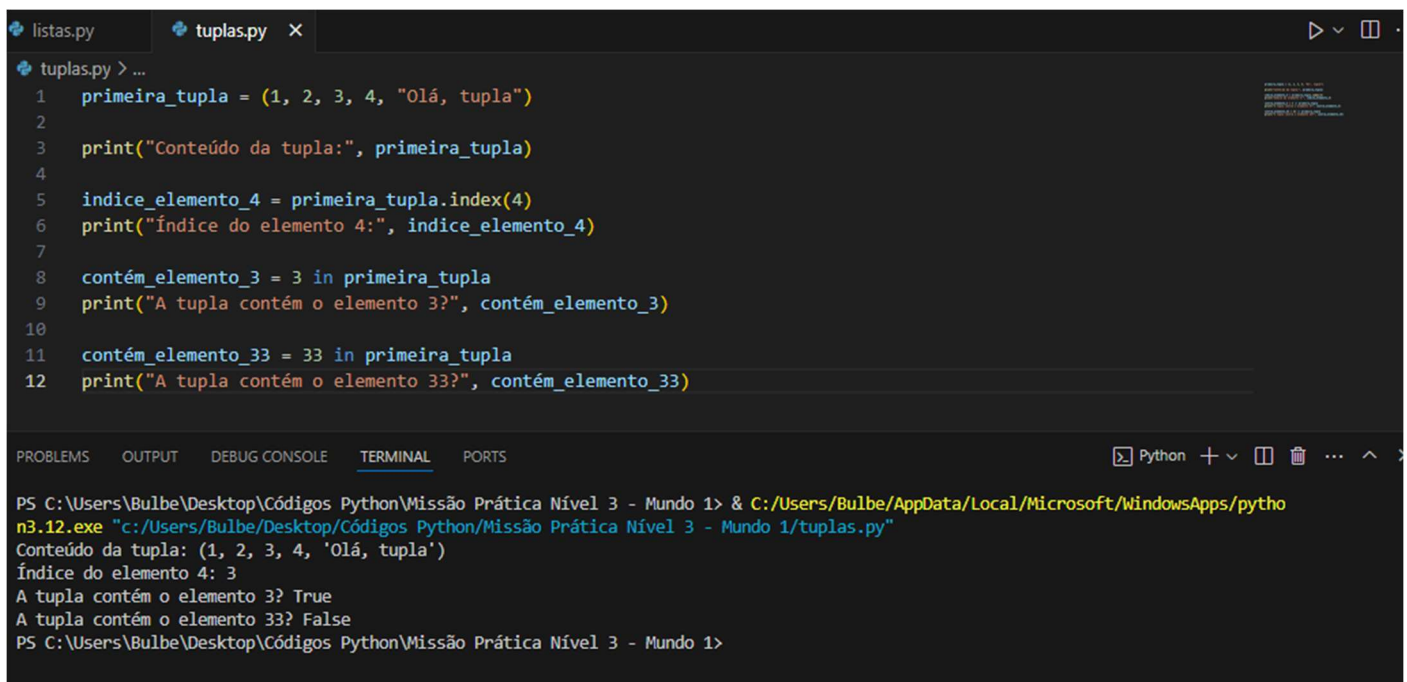
Tamanho atual da lista: 8

Conteúdo da lista após remover o item da posição 1: [1, 3, 'Olá, Python', 5, True, 12.6, ['Lista aninhada']]

Conteúdo da nova lista: [1, 3, 'Olá, Python', 5]

PS C:\Users\Bulbe\Desktop\Códigos Python\Missão Prática Nível 3 - Mundo 1>

- Microatividade 2:



```
listas.py x tuplas.py x
tuplas.py > ...
1  primeira_tupla = (1, 2, 3, 4, "Olá, tupla")
2
3  print("Conteúdo da tupla:", primeira_tupla)
4
5  indice_elemento_4 = primeira_tupla.index(4)
6  print("Índice do elemento 4:", indice_elemento_4)
7
8  contém_elemento_3 = 3 in primeira_tupla
9  print("A tupla contém o elemento 3?", contém_elemento_3)
10
11 contém_elemento_33 = 33 in primeira_tupla
12 print("A tupla contém o elemento 33?", contém_elemento_33)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS Python + -

PS C:\Users\Bulbe\Desktop\Códigos Python\Missão Prática Nível 3 - Mundo 1> & C:/Users/Bulbe/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/Bulbe/Desktop/Códigos Python/Missão Prática Nível 3 - Mundo 1/tuplas.py"

Conteúdo da tupla: (1, 2, 3, 4, 'Olá, tupla')

Índice do elemento 4: 3

A tupla contém o elemento 3? True

A tupla contém o elemento 33? False

PS C:\Users\Bulbe\Desktop\Códigos Python\Missão Prática Nível 3 - Mundo 1>

- Microatividade 3:

```
listas.py  tuplas.py  sets.py  ●

sets.py > ...
1  set_inicial = {11, 12, 13, 14}
2
3  print("Conteúdo do set inicial:", set_inicial)
4
5  set_inicial.add(15)
6
7  print("Conteúdo do set após adicionar 15:", set_inicial)
8
9  set_inicial.update({1, 2, 3, 4, 5})
10
11 print("Conteúdo do set após update:", set_inicial)
12
13 set_inicial.discard(13)
14
15 print("Conteúdo do set após remover 13:", set_inicial)
16
17 novo_set = {20, 21, 23, 1, 2}
18
19 print("Conteúdo do novo set:", novo_set)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Bulbe\Desktop\Códigos Python\Missão Prática Nível 3 - Mundo 1> & C:/Users/Bulbe/AppData/Local/Microsoft/WindowsApps/python
n3.12.exe "c:/Users/Bulbe/Desktop/Códigos Python/Missão Prática Nível 3 - Mundo 1/sets.py"
Conteúdo do set inicial: {11, 12, 13, 14}
Conteúdo do set após adicionar 15: {11, 12, 13, 14, 15}
Conteúdo do set após update: {1, 2, 3, 4, 5, 11, 12, 13, 14, 15}
Conteúdo do set após remover 13: {1, 2, 3, 4, 5, 11, 12, 14, 15}
Conteúdo do novo set: {1, 2, 20, 21, 23}
Resultado da união dos sets: {1, 2, 3, 4, 5, 11, 12, 14, 15, 20, 21, 23}
Resultado da interseção dos sets: {1, 2}
Resultado da diferença dos sets (set_inicial - novo_set): {3, 4, 5, 11, 12, 14, 15}
Resultado da diferença simétrica dos sets: {3, 4, 5, 11, 12, 14, 15, 20, 21, 23}
PS C:\Users\Bulbe\Desktop\Códigos Python\Missão Prática Nível 3 - Mundo 1>
```

```

20
21 uniao_sets = set_inicial.union(novo_set)
22 print("Resultado da união dos sets:", uniao_sets)
23
24 intersecao_sets = set_inicial.intersection(novo_set)
25 print("Resultado da interseção dos sets:", intersecao_sets)
26
27 diferenca_sets = set_inicial.difference(novo_set)
28 print("Resultado da diferença dos sets (set_inicial - novo_set):", diferenca_sets)
29
30 diferenca_simetrica_sets = set_inicial.symmetric_difference(novo_set)
31 print("Resultado da diferença simétrica dos sets:", diferenca_simetrica_sets)

```

- Microatividade 4:

```
listas.py  tuplas.py  sets.py  dicionarios.py  [Python icon] [Run icon] [Debug icon] [Terminal icon] [Ports icon]

dicionarios.py > ...
1  meu_dicionario = {
2      1: "Python",
3      2: "Java",
4      3: "PHP"
5  }
6
7  print("Conteúdo do dicionário:", meu_dicionario)
8
9  print("Tipo de dados de meu_dicionario:", type(meu_dicionario))
10
11 print("Valor da chave 1:", meu_dicionario.get(1))
12
13 print("Tamanho do dicionário:", len(meu_dicionario))
14
15 dicionario_frutas = {
16     1: {"nome": "limão", "tipo": "ácida"},
17     2: {"nome": "laranja", "tipo": "ácida"},
18     3: {"nome": "manga", "tipo": "semiácida"},
19     4: {"nome": "maçã", "tipo": "semiácida"},
20 }

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
Python + - [Python icon] [Run icon] [Debug icon] [Terminal icon] [Ports icon]

Conteúdo do dicionário: {1: 'Python', 2: 'Java', 3: 'PHP'}
Tipo de dados de meu_dicionario: <class 'dict'>
Valor da chave 1: Python
Tamanho do dicionário: 3
Chave 1 - Nome: limão , Tipo: ácido
Chave 2 - Nome: laranja , Tipo: ácido
Chave 1 - Nome: limão, Tipo: ácido
Chave 2 - Nome: laranja, Tipo: ácido
Chave 3 - Nome: manga, Tipo: semiácida
Chave 4 - Nome: maçã, Tipo: semiácida
Chave 5 - Nome: banana, Tipo: doce
Chave 6 - Nome: mamão, Tipo: doce
```

```
listas.py  tuplas.py  sets.py  dicionarios.py  [Python icon] [Run icon] [Debug icon] [Terminal icon] [Ports icon]

dicionarios.py > ...
14
15 dicionario_frutas = {
16     1: {"nome": "limão", "tipo": "ácida"},
17     2: {"nome": "laranja", "tipo": "ácida"},
18     3: {"nome": "manga", "tipo": "semiácida"},
19     4: {"nome": "maçã", "tipo": "semiácida"},
20     5: {"nome": "banana", "tipo": "doce"},
21     6: {"nome": "mamão", "tipo": "doce"}
22 }
23 print("Chave 1 - Nome:", dicionario_frutas[1]["nome"], ", Tipo:", dicionario_frutas[1]["tipo"])
24
25 print("Chave 2 - Nome:", dicionario_frutas[2]["nome"], ", Tipo:", dicionario_frutas[2]["tipo"])
26
27 for chave, valor in dicionario_frutas.items():
28     print(f"Chave {chave} - Nome: {valor['nome']}, Tipo: {valor['tipo']}")
29
```


- Microatividade 5:

```
listas.py  tuplas.py  sets.py  dicionarios.py  dicionarios2.py
dicionarios2.py > ...
1  meu_dicionario = {
2      1: {'nome': 'Maria', 'idade': 26, 'nacionalidade': 'brasileira'}
3  }
4
5  meu_dicionario.update({
6      2: {'nome': 'João', 'idade': 30, 'nacionalidade': 'português'},
7      3: {'nome': 'Ana', 'idade': 22, 'nacionalidade': 'espanhola'}
8  })
9
10 print("Dicionário atualizado:", meu_dicionario)
11
12 copia_dicionario = meu_dicionario.copy()
13
14 removido_pop = meu_dicionario.pop(2)
15 print("Elemento removido com pop:", removido_pop)
16 print("Conteúdo do dicionário após pop:", meu_dicionario)
17
18 removido_popitem = meu_dicionario.popitem()
19 print("Elemento removido com popitem:", removido_popitem)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
Python + v  [icon] [icon] ... ^ x

n3.12.exe "c:/Users/Bulbe/Desktop/Códigos Python/Missão Prática Nível 3 - Mundo 1/dicionarios2.py"
Dicionário atualizado: {1: {'nome': 'Maria', 'idade': 26, 'nacionalidade': 'brasileira'}, 2: {'nome': 'João', 'idade': 30, 'nacionali
dade': 'português'}, 3: {'nome': 'Ana', 'idade': 22, 'nacionalidade': 'espanhola'}}
Elemento removido com pop: {'nome': 'João', 'idade': 30, 'nacionalidade': 'português'}
Conteúdo do dicionário após pop: {1: {'nome': 'Maria', 'idade': 26, 'nacionalidade': 'brasileira'}, 3: {'nome': 'Ana', 'idade': 22, '
nacionalidade': 'espanhola'}}
Elemento removido com popitem: (3, {'nome': 'Ana', 'idade': 22, 'nacionalidade': 'espanhola'})
Conteúdo do dicionário após popitem: {1: {'nome': 'Maria', 'idade': 26, 'nacionalidade': 'brasileira'}}
Conteúdo do novo dicionário (items): dict_items([(4, 'valor padrão'), (5, 'valor padrão'), (6, 'valor padrão')])
Chaves do novo dicionário (keys): dict_keys([4, 5, 6])
Valores do novo dicionário (values): dict_values(['valor padrão', 'valor padrão', 'valor padrão'])
PS C:\Users\Bulbe\Desktop\Códigos Python\Missão Prática Nível 3 - Mundo 1>
```

```
listas.py  tuplas.py  sets.py  dicionarios.py  dicionarios2.py
dicionarios2.py > ...
17
18 removido_popitem = meu_dicionario.popitem()
19 print("Elemento removido com popitem:", removido_popitem)
20 print("Conteúdo do dicionário após popitem:", meu_dicionario)
21
22 meu_dicionario.clear()
23 copia_dicionario.clear()
24
25 novas_chaves = [4, 5, 6]
26 valor_padrao = "valor padrão"
27 novo_dicionario = dict.fromkeys(novas_chaves, valor_padrao)
28
29 print("Conteúdo do novo dicionário (items):", novo_dicionario.items())
30
31 print("Chaves do novo dicionário (keys):", novo_dicionario.keys())
32
33 print("Valores do novo dicionário (values):", novo_dicionario.values())
```

- Missão Prática:

```
operacoes.py > ...
1 def calcular_media(notas):
2     """
3     Calcula a média das notas fornecidas.
4
5     :param notas: Lista com as notas dos 4 bimestres.
6     :return: Média das notas.
7     """
8     return sum(notas) / len(notas)
9
10 def verificar_reprovacao(media):
11     """
12     Verifica se a média é inferior a 6 (reprovado).
13
14     :param media: Média das notas.
15     :return: True se a média for inferior a 6, False caso contrário.
16     """
17     return media < 6
18
19 def alunos_reprovados(dados_alunos, matriculas_reprovados):
20     """
21     Retorna informações sobre os alunos reprovados.
22
23     :param dados_alunos: Dicionário com informações dos alunos.
24     :param matriculas_reprovados: Lista com as matrículas dos alunos reprovados.
25     :return: Lista de strings com informações sobre os alunos reprovados.
26     """
27     reprovados = []
28     for matricula in matriculas_reprovados:
29         aluno = dados_alunos.get(matricula)
30         if aluno:
31             reprovados.append(
32                 f"Aluno Reprovado: {aluno['nome']} | Matrícula: {matricula} | Média Final: {aluno['media']:.2f}"
33             )
34     return reprovados
35
36 dados_alunos = {
37     26: {'nome': 'Maria', 'notas': [8, 7, 5, 9]},
38     101: {'nome': 'Ana', 'notas': [9, 9, 8, 9]},
39     13: {'nome': 'João', 'notas': [6, 5, 5, 5]},
40     37: {'nome': 'Ágatha', 'notas': [8, 6, 7.5, 9]},
41     72: {'nome': 'Joaquim', 'notas': [6, 5.5, 5, 7]},
42     5: {'nome': 'Félix', 'notas': [10, 8, 8, 8]}
43 }
44
45 for matricula, dados in dados_alunos.items():
46     media = calcular_media(dados['notas'])
47     dados['media'] = media
48
49 matriculas_reprovados = [matricula for matricula, dados in dados_alunos.items() if verificar_reprovacao(dados['media'])]
50
51 lista_reprovados = alunos_reprovados(dados_alunos, matriculas_reprovados)
```

```

main.py > ...
1
2 from operacoes import calcular_media, verificar_reprovacao, alunos_reprovados
3
4 dados_alunos = {
5     26: {'nome': 'Maria', 'notas': [8, 7, 5, 9]},
6     101: {'nome': 'Ana', 'notas': [9, 9, 8, 9]},
7     13: {'nome': 'João', 'notas': [6, 5, 5, 5]},
8     37: {'nome': 'Ágatha', 'notas': [8, 6, 7.5, 9]},
9     72: {'nome': 'Joaquim', 'notas': [6, 5.5, 5, 7]},
10    5: {'nome': 'Félix', 'notas': [10, 8, 8, 8]}
11 }
12
13 for matricula, dados in dados_alunos.items():
14     media = calcular_media(dados['notas'])
15     dados['media'] = media
16
17 matriculas_reprovados = [matricula for matricula, dados in dados_alunos.items() if verificar_reprovacao(dados['media'])]
18
19 lista_reprovados = alunos_reprovados(dados_alunos, matriculas_reprovados)
20
21 for info in lista_reprovados:
22     print(info)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python + v [

PS C:\Users\Bulbe\Desktop\Códigos Python\Missão Prática Nível 3 - Mundo 1> & C:/Users/Bulbe/AppData/Local/Microsoft/WindowsApps/python3.12.exe /Desktop/Códigos Python/Missão Prática Nível 3 - Mundo 1/main.py"

Aluno Reprovado: João - Matrícula: 13 - Média Final: 5.25

Aluno Reprovado: Joaquim - Matrícula: 72 - Média Final: 5.88

PS C:\Users\Bulbe\Desktop\Códigos Python\Missão Prática Nível 3 - Mundo 1>