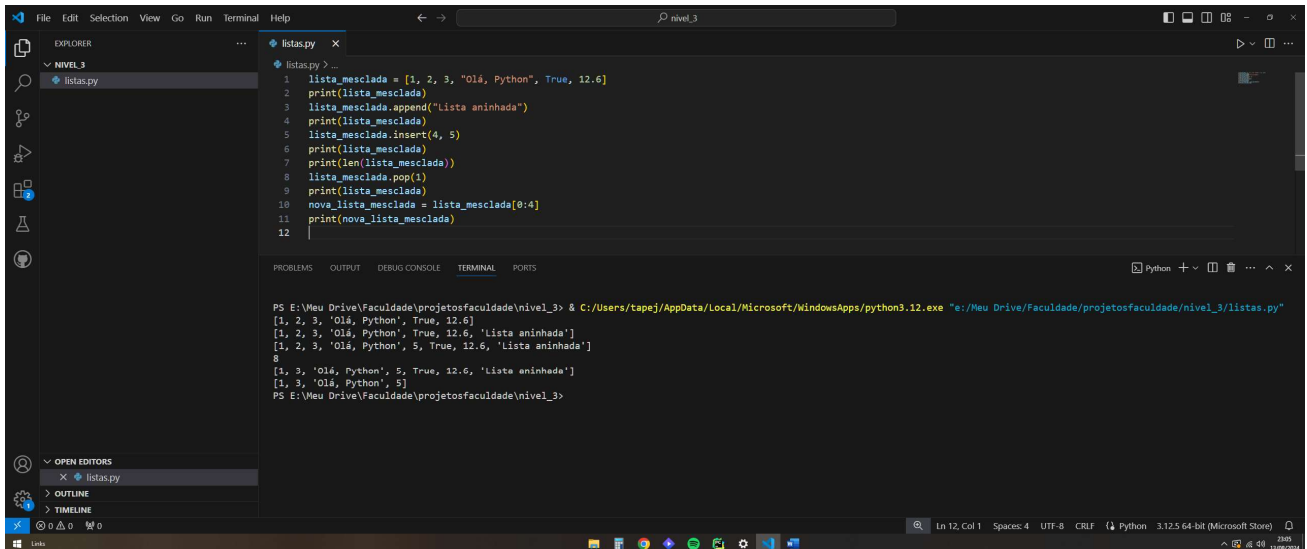


Missão Prática - Nível 3 – Mundo 1

Microatividade 1:

Manipulação da estrutura de dados lista em Python



The screenshot shows the Visual Studio Code interface with a file explorer on the left showing a project named 'NIVEL 3' containing a file 'listas.py'. The main editor displays the code for 'listas.py' with the following content:

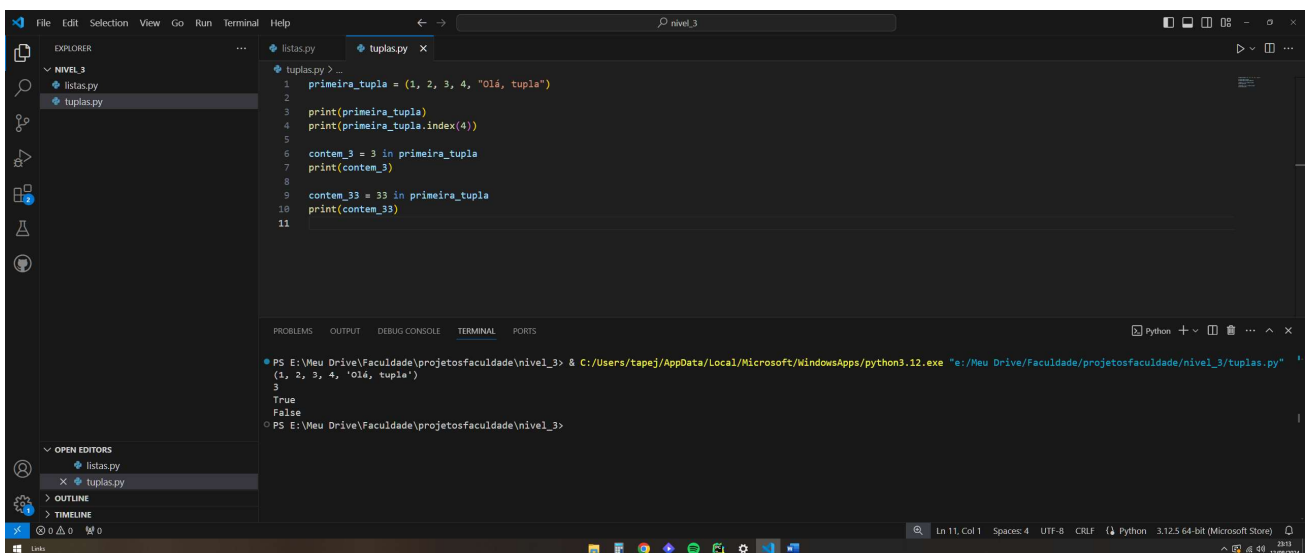
```
1 lista_mesclada = [1, 2, 3, "Olá, Python", True, 12.6]
2 print(lista_mesclada)
3 lista_mesclada.append("Lista aninhada")
4 print(lista_mesclada)
5 lista_mesclada.insert(4, 5)
6 print(lista_mesclada)
7 print(len(lista_mesclada))
8 lista_mesclada.pop(1)
9 print(lista_mesclada)
10 nova_lista_mesclada = lista_mesclada[0:4]
11 print(nova_lista_mesclada)
12
```

The bottom panel shows the terminal output for the command: `PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel_3> & C:/Users/tapej/AppData/Local/Microsoft/WindowsApps/python3.12.exe "e:/Meu Drive/Faculdade/projetosfaculdade/nivel_3/listas.py"`

```
[1, 2, 3, 'Olá, Python', True, 12.6]
[1, 2, 3, 'Olá, Python', True, 12.6, 'Lista aninhada']
[1, 2, 3, 'Olá, Python', 5, True, 12.6, 'Lista aninhada']
8
[1, 3, 'Olá, Python', 5, True, 12.6, 'Lista aninhada']
[1, 3, 'Olá, Python', 5]
PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel_3>
```

Microatividade 2:

Manipulação da estrutura de dados tupla em Python



The screenshot shows the Visual Studio Code interface with a file explorer on the left showing a project named 'NIVEL 3' containing a file 'tuplas.py'. The main editor displays the code for 'tuplas.py' with the following content:

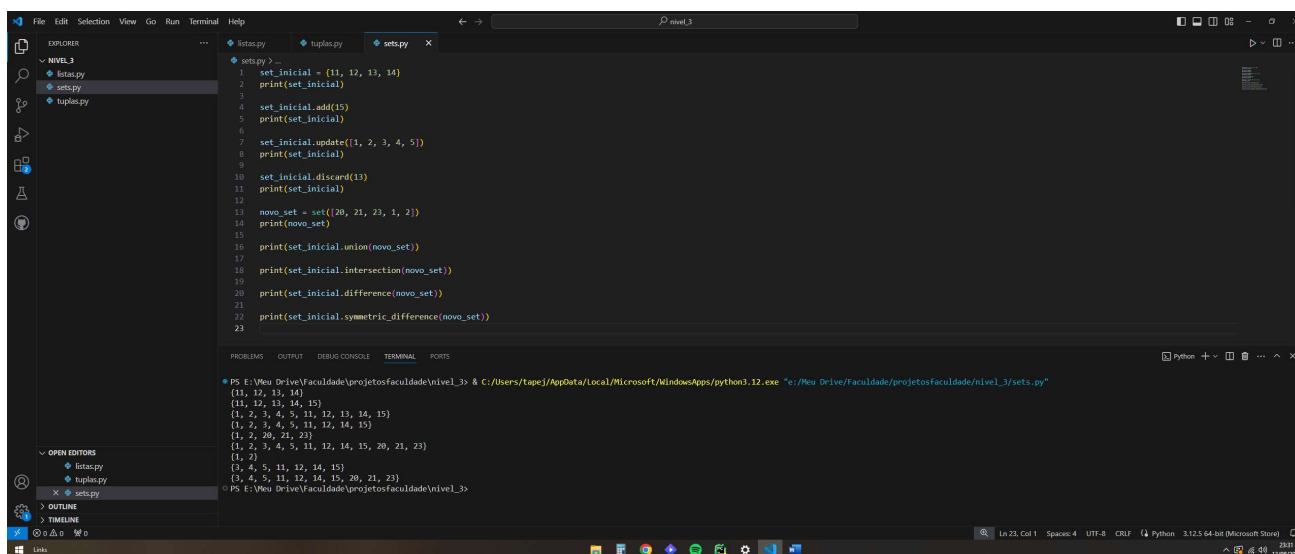
```
1 primeira_tupla = (1, 2, 3, 4, "Olá, tupla")
2
3 print(primeira_tupla)
4 print(primeira_tupla.index(4))
5
6 contem_3 = 3 in primeira_tupla
7 print(contem_3)
8
9 contem_33 = 33 in primeira_tupla
10 print(contem_33)
11
```

The bottom panel shows the terminal output for the command: `PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel_3> & C:/Users/tapej/AppData/Local/Microsoft/WindowsApps/python3.12.exe "e:/Meu Drive/Faculdade/projetosfaculdade/nivel_3/tuplas.py"`

```
(1, 2, 3, 4, 'Olá, tupla')
3
True
False
PS E:\Meu Drive\Faculdade\projetosfaculdade\nivel_3>
```

Microatividade 3:

Manipulação da estrutura de dados set em Python



The screenshot shows a Python IDE with a file explorer on the left containing 'nivel_3', 'listasy.py', 'setspy.py', and 'tuplas.py'. The 'setspy.py' file is open in the editor, displaying the following code:

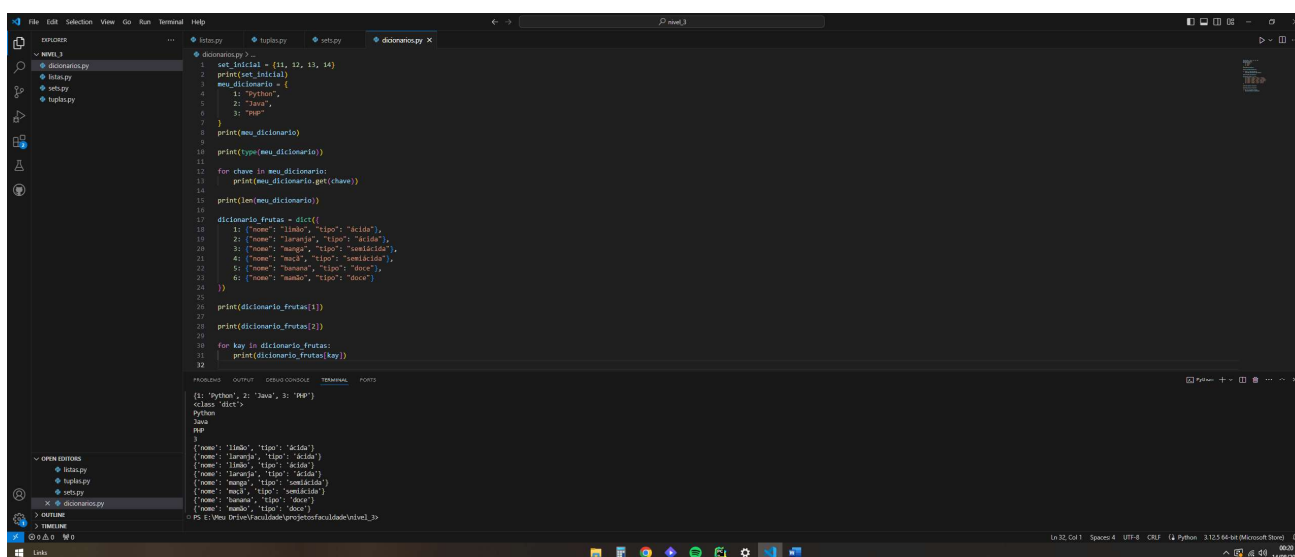
```
1 set_inicial = {11, 12, 13, 14}
2 print(set_inicial)
3
4 set_inicial.add(15)
5 print(set_inicial)
6
7 set_inicial.update([1, 2, 3, 4, 5])
8 print(set_inicial)
9
10 set_inicial.discard(13)
11 print(set_inicial)
12
13 novo_set = set([20, 21, 23, 1, 2])
14 print(novo_set)
15
16 print(set_inicial.union(novo_set))
17
18 print(set_inicial.intersection(novo_set))
19
20 print(set_inicial.difference(novo_set))
21
22 print(set_inicial.symmetric_difference(novo_set))
23
```

The terminal at the bottom shows the output of the script:

```
PS E:\Veu Drive\Faculdade\projetos\faculdade\nivel_3> & C:/Users/tapej/AppData/Local/Microsoft/WindowsApps/python3.12.exe "e:/Veu Drive/Faculdade/projetos/faculdade/nivel_3/setspy.py"
{11, 12, 13, 14}
{11, 12, 13, 14, 15}
{1, 2, 3, 4, 5, 11, 12, 13, 14, 15}
{1, 2, 3, 4, 5, 11, 12, 14, 15}
{1, 2, 20, 21, 23}
{1, 2, 3, 4, 5, 11, 12, 14, 15, 20, 21, 23}
{1, 2}
{3, 4, 5, 11, 12, 14, 15}
{3, 4, 5, 11, 12, 14, 15, 20, 21, 23}
PS E:\Veu Drive\Faculdade\projetos\faculdade\nivel_3>
```

Microatividade 4:

Criação da estrutura de dados dicionário em Python



The screenshot shows a Python IDE with a file explorer on the left containing 'nivel_3', 'listasy.py', 'setspy.py', and 'dicionariopy.py'. The 'dicionariopy.py' file is open in the editor, displaying the following code:

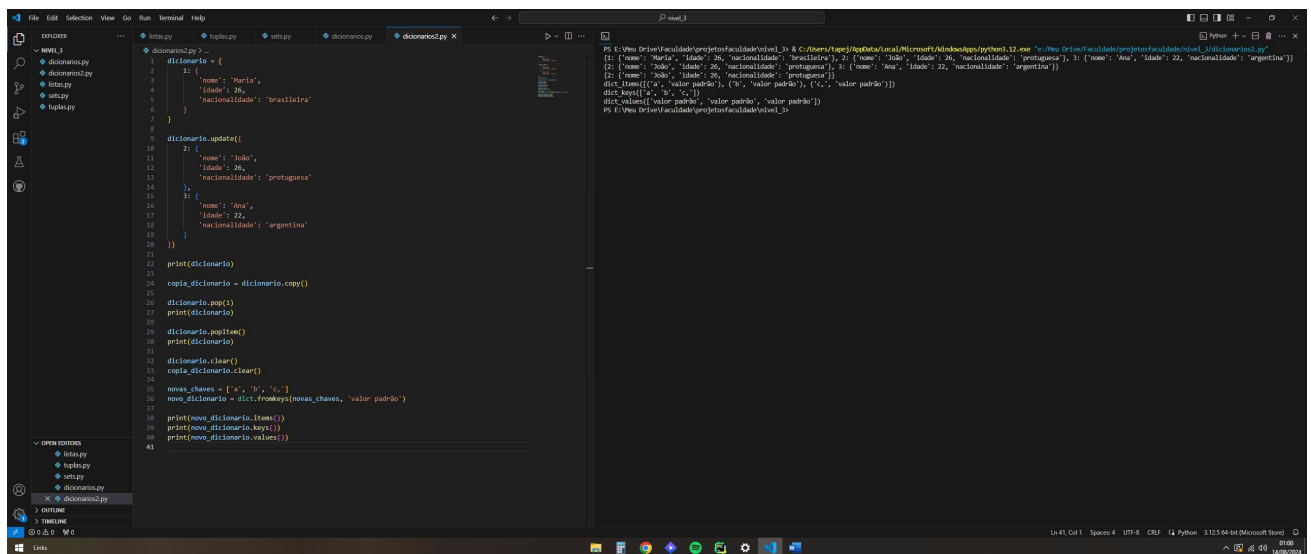
```
1 set_inicial = {11, 12, 13, 14}
2 print(set_inicial)
3 meu_dicionario = {
4     1: "Python",
5     2: "Java",
6     3: "PHP"
7 }
8 print(meu_dicionario)
9
10 print(type(meu_dicionario))
11
12 for chave in meu_dicionario:
13     print(meu_dicionario.get(chave))
14
15 print(len(meu_dicionario))
16
17 dicionario_frutas = dict([
18     1: ("nome": "lilabo", "tipo": "ácida"),
19     2: ("nome": "laranja", "tipo": "ácida"),
20     3: ("nome": "manga", "tipo": "semiacida"),
21     4: ("nome": "maçã", "tipo": "semiacida"),
22     5: ("nome": "banana", "tipo": "doce"),
23     6: ("nome": "mamo", "tipo": "doce")
24 ])
25
26 print(dicionario_frutas[1])
27
28 print(dicionario_frutas[2])
29
30 for key in dicionario_frutas:
31     print(dicionario_frutas[key])
32
```

The terminal at the bottom shows the output of the script:

```
PS E:\Veu Drive\Faculdade\projetos\faculdade\nivel_3>
{1: 'Python', 2: 'Java', 3: 'PHP'}
<class 'dict'>
Python
Java
PHP
1
{'nome': 'lilabo', 'tipo': 'ácida'}
{'nome': 'laranja', 'tipo': 'ácida'}
{'nome': 'laranja', 'tipo': 'ácida'}
{'nome': 'manga', 'tipo': 'semiacida'}
{'nome': 'maçã', 'tipo': 'semiacida'}
{'nome': 'banana', 'tipo': 'doce'}
{'nome': 'mamo', 'tipo': 'doce'}
PS E:\Veu Drive\Faculdade\projetos\faculdade\nivel_3>
```

Microatividade 5:

Atualização de dados em um dicionário



The image shows a screenshot of a Visual Studio Code editor with a Python file named `dicionario2.py` open. The code defines a dictionary `dicionario` with three entries: Maria (26, Brazilian), João (26, Portuguese), and Ana (22, Argentine). It then performs several operations: prints the dictionary, creates a copy, adds a new entry for João, updates Maria's age to 27, removes Ana, and finally prints the dictionary again. The output in the terminal shows the dictionary after these changes: `{'nome': 'Maria', 'idade': 27, 'nacionalidade': 'brasileira'}, {'nome': 'João', 'idade': 26, 'nacionalidade': 'portuguesa'}`.

```
1 dicionario = {
2     1: {
3         'nome': 'Maria',
4         'idade': 26,
5         'nacionalidade': 'brasileira'
6     }
7 }
8
9 dicionario.update({
10     2: {
11         'nome': 'João',
12         'idade': 26,
13         'nacionalidade': 'portuguesa'
14     },
15     3: {
16         'nome': 'Ana',
17         'idade': 22,
18         'nacionalidade': 'argentina'
19     }
20 })
21
22 print(dicionario)
23
24 copia_dicionario = dicionario.copy()
25
26 dicionario.pop(1)
27 print(dicionario)
28
29 dicionario.popitem()
30 print(dicionario)
31
32 dicionario.clear()
33 copia_dicionario.clear()
34
35 novas_chaves = ['a', 'b', 'c']
36 novo_dicionario = dict.fromkeys(novas_chaves, 'valor padrão')
37
38 print(novo_dicionario.items())
39 print(novo_dicionario.keys())
40 print(novo_dicionario.values())
41
```

Missão Prática:

Estruturando os Dados

```
main.py
1 from operacoes import calcular_media, verificar_reprovacao, identificar_reprovados
2
3 # Estruturas de dados necessárias para armazenar os dados dos alunos e as notas de cada bimestre.
4 dados_alunos = {
5     101: {'nome': 'Maria', 'notas': [8, 7, 5, 9]},
6     102: {'nome': 'Ana', 'notas': [9, 9, 8, 8]},
7     103: {'nome': 'João', 'notas': [6, 6, 5, 5]},
8     104: {'nome': 'Agatha', 'notas': [8, 6, 7, 5, 9]},
9     105: {'nome': 'Joaquim', 'notas': [6, 5, 5, 5, 7]},
10    106: {'nome': 'Felix', 'notas': [10, 8, 8, 8]}
11 }
12
13 matriculas_reprovados = []
14
15 # Calcular a média de cada aluno e verificar se foi reprovado
16 for matricula, info_aluno in dados_alunos.items():
17     media = calcular_media(info_aluno['notas'])
18     info_aluno['media'] = media
19     if verificar_reprovacao(media):
20         matriculas_reprovados.append(matricula)
21
22 mensagens_reprovados = identificar_reprovados(dados_alunos, matriculas_reprovados)
23 for mensagem in mensagens_reprovados:
24     print(mensagem)
25
operacoes.py
1
2
3 def calcular_media(notas):
4     """
5     :param notas: Array contendo as notas dos 4 bimestres de cada aluno
6     :return: A média das notas
7     """
8     return sum(notas) / len(notas)
9
10 def verificar_reprovacao(media):
11     """
12     :param media: Média das notas do aluno.
13     :return: True se o aluno foi reprovado (média inferior a 6), False caso aluno aprovado.
14     """
15     return media < 6
16
17 def identificar_reprovados(dados_alunos, matriculas_reprovados):
18     """
19     :param dados_alunos: Dicionário com os dados dos alunos (matricula, nome e notas).
20     :param matriculas_reprovados: Lista com os números das matrículas dos alunos reprovados.
21     :return: Lista de strings com as mensagens dos alunos reprovados.
22     """
23     mensagens = []
24     for matricula in matriculas_reprovados:
25         aluno = dados_alunos[matricula]
26         mensagem = f'Aluno Reprovado: {aluno["nome"]} - Matrícula: {matricula} - Média Final: {aluno["media"]:.2f}'
27         mensagens.append(mensagem)
28     return mensagens
29
```