

# Relatorio Missão Prática

## Relatorio listas.py



```
listas.py > ...
1 # Criação da lista 'lista_mesclada'
2 lista_mesclada = [1, 2, 3, "Olá, Python", True, 12.6]
3
4 # Imprimindo o conteúdo da lista
5 print("Conteúdo inicial da lista_mesclada:", lista_mesclada)
6
7 # Adicionando o elemento ["Lista aninhada"] usando o método 'append'
8 lista_mesclada.append(["Lista aninhada"])
9
10 # Imprimindo o conteúdo da lista após o 'append'
11 print("Conteúdo após append:", lista_mesclada)
12
13 # Inserindo o elemento 5 na posição 4 usando o método 'insert'
14 lista_mesclada.insert(4, 5)
15
16 # Imprimindo o conteúdo da lista após o 'insert'
```

PROBLEMAS SAÍDA CONSOLE DE DEPUAÇÃO **TERMINAL** PORTAS Python + - [ ] [ ] ... ^ X

```
PS C:\Users\event\OneDrive\Área de Trabalho\missao_pratica_n3_estrutura_de_dados> & C:/Users/event/AppData/Local/Microsof
t/WindowsApps/python3.11.exe "c:/Users/event/OneDrive/Área de Trabalho/missao_pratica_n3_estrutura_de_dados/listas.py"
Conteúdo inicial da lista_mesclada: [1, 2, 3, 'Olá, Python', True, 12.6]
Conteúdo após append: [1, 2, 3, 'Olá, Python', True, 12.6, ['Lista aninhada']]
Conteúdo após insert: [1, 2, 3, 'Olá, Python', 5, True, 12.6, ['Lista aninhada']]
Tamanho da lista_mesclada: 8
Conteúdo após remoção: [1, 3, 'Olá, Python', 5, True, 12.6, ['Lista aninhada']]
Conteúdo da nova_lista_mesclada: [1, 3, 'Olá, Python', 5, True]
PS C:\Users\event\OneDrive\Área de Trabalho\missao_pratica_n3_estrutura_de_dados>
```

## Relatorio tuplas.py

```
tuplas.py > ...
5 print("Conteúdo da primeira_tupla:", primeira_tupla)
6
7 # Imprimindo o índice do elemento 4 usando o método 'index'
8 indice_4 = primeira_tupla.index(4)
9 print("Índice do elemento 4 na tupla:", indice_4)
10
11 # Verificando se a tupla contém o elemento 3
12 contém_3 = 3 in primeira_tupla
13 print("A tupla contém o elemento 3:", contém_3)
14
15 # Verificando se a tupla contém o elemento 33
16 contém_33 = 33 in primeira_tupla
17 print("A tupla contém o elemento 33:", contém_33)
18
```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL PORTAS Python + - [ ] [ ] ... ^

PS C:\Users\event\OneDrive\Área de Trabalho\missao\_pratica\_n3\_estrutura\_de\_dados> & C:/Users/event/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/event/OneDrive/Área de Trabalho/missao\_pratica\_n3\_estrutura\_de\_dados/tuplas.py"

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

Índice do elemento 4 na tupla: 3

A tupla contém o elemento 3: True

A tupla contém o elemento 33: False

PS C:\Users\event\OneDrive\Área de Trabalho\missao\_pratica\_n3\_estrutura\_de\_dados>

## Relatório sets.py

```
sets.py > ...
31 # Imprimindo o resultado da união entre os dois sets
32 uniao = set_inicial.union(novo_set)
33 print("União entre set_inicial e novo_set:", uniao)
34
35 # Imprimindo o resultado da interseção entre os dois sets
36 interseccao = set_inicial.intersection(novo_set)
37 print("Interseção entre set_inicial e novo_set:", interseccao)
38
39 # Imprimindo o resultado da diferença entre os dois sets
40 diferenca = set_inicial.difference(novo_set)
41 print("Diferença entre set_inicial e novo_set:", diferenca)
42
43 # Imprimindo o resultado da diferença simétrica entre os dois sets
44 diferenca_simetrica = set_inicial.symmetric_difference(novo_set)
45 print("Diferença simétrica entre set_inicial e novo_set:", diferenca_simetrica)
46
```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL PORTAS Python + - [ ] [ ] ... ^

PS C:\Users\event\OneDrive\Área de Trabalho\missao\_pratica\_n3\_estrutura\_de\_dados> & C:/Users/event/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/event/OneDrive/Área de Trabalho/missao\_pratica\_n3\_estrutura\_de\_dados/sets.py"

Conteúdo inicial do set\_inicial: {11, 12, 13, 14}

Conteúdo após add: {11, 12, 13, 14, 15}

Conteúdo após update: {1, 2, 3, 4, 5, 11, 12, 13, 14, 15}

Conteúdo após discard: {1, 2, 3, 4, 5, 11, 12, 14, 15}

Conteúdo do novo\_set: {1, 2, 20, 21, 23}

União entre set\_inicial e novo\_set: {1, 2, 3, 4, 5, 11, 12, 14, 15, 20, 21, 23}

Interseção entre set\_inicial e novo\_set: {1, 2}

Diferença entre set\_inicial e novo\_set: {3, 4, 5, 11, 12, 14, 15}

Diferença simétrica entre set\_inicial e novo\_set: {3, 4, 5, 11, 12, 14, 15, 20, 21, 23}

PS C:\Users\event\OneDrive\Área de Trabalho\missao\_pratica\_n3\_estrutura\_de\_dados>

## Relatório dicionarios.py

```
missao_pratica_n3estrutura_de_dados > dicionarios.py > ...
1  # Criação do dicionário 'meu_dicionario'
2  meu_dicionario = {
3      1: "Python",
4      2: "Java",
5      3: "PHP"
6  }
7
8  # Imprimindo o conteúdo do dicionário
9  print("Conteúdo do meu_dicionario:", meu_dicionario)
10
11 # Imprimindo o tipo de dados do dicionário
12 print("Tipo de dados do meu_dicionario:", type(meu_dicionario))
13
14 # Imprimindo o valor da chave 1 usando o método 'get'
15 # Note que 'get' precisa da chave para buscar o valor correto
16 print("Valor da chave 1:", meu_dicionario.get(1))
```

PROBLEMAS SAÍDA CONSOLE DE DEPUÇÃO **TERMINAL** PORTAS Python + - □ □ ... ^ x

PS C:\Users\event\OneDrive\Área de Trabalho\missao\_pratica\_n3estrutura\_de\_dados > & C:\Users\event\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/Users/event/OneDrive/Área de Trabalho/missao\_pratica\_n3estrutura\_de\_dados/dicionarios.py"

Conteúdo do meu\_dicionario: {1: 'Python', 2: 'Java', 3: 'PHP'}

Tipo de dados do meu\_dicionario: <class 'dict'>

Valor da chave 1: Python

Tamanho do meu\_dicionario: 3

Chave 1 - Nome: limão

Chave 1 - Tipo: ácida

Chave 2 - Nome: laranja

Chave 2 - Tipo: ácida

Valores de todas as chaves 'nome' e 'tipo' em dicionario\_frutas:

```
missao_pratica_n3estrutura_de_dados > dicionarios.py > ...
1  # Criação do dicionário 'meu_dicionario'
2  meu_dicionario = {
3      1: "Python",
4      2: "Java",
5      3: "PHP"
6  }
7
8  # Imprimindo o conteúdo do dicionário
9  print("Conteúdo do meu_dicionario:", meu_dicionario)
10
11 # Imprimindo o tipo de dados do dicionário
12 print("Tipo de dados do meu_dicionario:", type(meu_dicionario))
13
14 # Imprimindo o valor da chave 1 usando o método 'get'
15 # Note que 'get' precisa da chave para buscar o valor correto
16 print("Valor da chave 1:", meu_dicionario.get(1))
```

PROBLEMAS SAÍDA CONSOLE DE DEPUÇÃO **TERMINAL** PORTAS Python + - □ □ ... ^

Chave 1 - Nome: limão

Chave 1 - Tipo: ácida

Chave 2 - Nome: laranja

Chave 2 - Tipo: ácida

Valores de todas as chaves 'nome' e 'tipo' em dicionario\_frutas:

Chave 1 - Nome: limão, Tipo: ácida

Chave 2 - Nome: laranja, Tipo: ácida

Chave 3 - Nome: manga, Tipo: semiácida

Chave 4 - Nome: maçã, Tipo: semiácida

Chave 5 - Nome: banana, Tipo: doce

Chave 6 - Nome: mamão, Tipo: doce

PS C:\Users\event\OneDrive\Área de Trabalho\missao\_pratica\_n3estrutura\_de\_dados > |

## Relatório dicionarios2.py

```
missao_pratica_n3_estrutura_de_dados > dicionarios2.py > ...
32 print("Dicionário original após clear:", dicionario)
33 print("Dicionário cópia após clear:", dicionario_copia)
34
35 # Definindo um novo dicionário usando o método 'fromkeys'
36 novos_dicionario = dict.fromkeys(['a', 'b', 'c'], 'valor_default')
37
38 # Imprimindo o conteúdo do novo dicionário usando o método 'items'
39 print("Conteúdo do novos_dicionario usando 'items':", novos_dicionario.items())
40
41 # Imprimindo apenas as chaves do novo dicionário usando o método 'keys'
42 print("Chaves do novos_dicionario usando 'keys':", novos_dicionario.keys())
43
44 # Imprimindo apenas os valores do novo dicionário usando o método 'values'
45 print("Valores do novos_dicionario usando 'values':", novos_dicionario.values())
46
```

PROBLEMAS SAÍDA CONSOLE DE DEPURÇÃO TERMINAL PORTAS

PS C:\Users\event\OneDrive\Área de Trabalho\missao\_pratica\_n3\_estrutura\_de\_dados> & C:\Users\event\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/Users/event/OneDrive/Área de Trabalho/missao\_pratica\_n3\_estrutura\_de\_dados/dicionarios2.py"

Dicionário atualizado: {1: {'nome': 'Maria', 'idade': 26, 'nacionalidade': 'brasileira'}, 2: {'nome': 'João', 'idade': 30, 'nacionalidade': 'portuguesa'}, 3: {'nome': 'Ana', 'idade': 22, 'nacionalidade': 'argentina'}}  
Dicionário após pop (elemento removido): {2: {'nome': 'João', 'idade': 30, 'nacionalidade': 'portuguesa'}, 3: {'nome': 'Ana', 'idade': 22, 'nacionalidade': 'argentina'}}  
Elemento removido: {'nome': 'Maria', 'idade': 26, 'nacionalidade': 'brasileira'}  
Dicionário após popitem (último elemento removido): {2: {'nome': 'João', 'idade': 30, 'nacionalidade': 'portuguesa'}}  
Elemento removido: (3, {'nome': 'Ana', 'idade': 22, 'nacionalidade': 'argentina'})  
Dicionário original após clear: {}  
Dicionário cópia após clear: {}

```
missao_pratica_n3_estrutura_de_dados.py - dicionarios.py
32 print("Dicionário original após clear:", dicionario)
33 print("Dicionário cópia após clear:", dicionario_copia)
34
35 # Definindo um novo dicionário usando o método 'fromkeys'
36 novos_dicionario = dict.fromkeys(['a', 'b', 'c'], 'valor_default')
37
38 # Imprimindo o conteúdo do novo dicionário usando o método 'items'
39 print("Conteúdo do novos_dicionario usando 'items':", novos_dicionario.items())
40
41 # Imprimindo apenas as chaves do novo dicionário usando o método 'keys'
42 print("Chaves do novos_dicionario usando 'keys':", novos_dicionario.keys())
43
44 # Imprimindo apenas os valores do novo dicionário usando o método 'values'
45 print("Valores do novos_dicionario usando 'values':", novos_dicionario.values())
46
```

PROBLEMAS    SAÍDA    CONSOLE DE DEPURAÇÃO    TERMINAL    PORTAS

Dicionário após pop (elemento removido): {2: {'nome': 'João', 'idade': 30, 'nacionalidade': 'portuguesa'}, 3: {'nome': 'Ana', 'idade': 22, 'nacionalidade': 'argentina'}}  
Elemento removido: {'nome': 'Maria', 'idade': 26, 'nacionalidade': 'brasileira'}  
Dicionário após popitem (último elemento removido): {2: {'nome': 'João', 'idade': 30, 'nacionalidade': 'portuguesa'}}  
Elemento removido: (3, {'nome': 'Ana', 'idade': 22, 'nacionalidade': 'argentina'})  
Dicionário original após clear: {}  
Dicionário cópia após clear: {}  
Conteúdo do novos\_dicionario usando 'items': dict\_items([('a', 'valor\_default'), ('b', 'valor\_default'), ('c', 'valor\_default')])  
Chaves do novos\_dicionario usando 'keys': dict\_keys(['a', 'b', 'c'])  
Valores do novos\_dicionario usando 'values': dict\_values(['valor\_default', 'valor\_default', 'valor\_default'])  
PS C:\Users\evert\OneDrive\Área de Trabalho\missao\_pratica\_n3\_estrutura\_de\_dados>

## Relatório Estruturando Dados

## operacoes.py

```
missao_pratica_n3_estrutura_de_dados > operacoes.py > calcular_media
1 def calcular_media(notas):
2     """Calcula a média das notas fornecidas."""
3     if not notas:
4         return 0
5     return sum(notas) / len(notas)
6
7 def verificar_reprovacao(media):
8     """Verifica se o aluno foi reprovado com base na média."""
9     return media < 6
10
11 def relatorio_reprovacao(dados_alunos, matriculas_reprovados):
12     """Gera um relatório dos alunos reprovados e suas médias."""
13     relatorio = []
14     for matricula in matriculas_reprovados:
15         aluno = dados_alunos.get(matricula, {})
16         nome = aluno.get('nome', 'Desconhecido')
```

PROBLEMAS SAÍDA CONSOLE DE DEBURAÇÃO TERMINAL PORTAS Python + - □ □ □ ... ^ ×

PS C:\Users\event\OneDrive\Área de Trabalho\missao\_pratica\_n3\_estrutura\_de\_dados> & C:/Users/event/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/event/OneDrive/Área de Trabalho/missao\_pratica\_n3\_estrutura\_de\_dados/operacoes.py"

PS C:\Users\event\OneDrive\Área de Trabalho\missao\_pratica\_n3\_estrutura\_de\_dados>

## main.py

```
missao_pratica_n3_estrutura_de_dados > main.py > ...
1 from operacoes import calcular_media, verificar_reprovacao, relatorio_reprovacao, relatorio_completo
2
3 def inicializar_cadastro():
4     """Inicializa o cadastro de alunos com nomes e matrículas."""
5     nomes = ('Maria', 'Ana', 'João', 'Agatha', 'Joaquin', 'Felix') # Tupla com nomes
6     matriculas = (26, 101, 13, 37, 72, 5) # Tupla com matrículas
7     cadastro = dict(zip(matriculas, nomes)) # Dicionário associando matrículas a nomes
8     return cadastro
9
10 def inicializar_notas():
11     """Inicializa as notas por bimestre associadas a cada matrícula."""
12     notas_primeiro_bimestre = {
13         26: 7,
14         101: 5,
15         13: 8,
16         37: 4,
```

PROBLEMAS SAÍDA CONSOLE DE DEBURAÇÃO TERMINAL PORTAS Python + - □ □ □ ... ^ ×

t/WindowsApps/python3.11.exe "c:/Users/event/OneDrive/Área de Trabalho/missao\_pratica\_n3\_estrutura\_de\_dados/operacoes.py"

PS C:\Users\event\OneDrive\Área de Trabalho\missao\_pratica\_n3\_estrutura\_de\_dados> & C:/Users/event/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/event/OneDrive/Área de Trabalho/missao\_pratica\_n3\_estrutura\_de\_dados/main.py"

Relatório dos Alunos Reprovados:  
Aluno Reprovado: Agatha - Matrícula: 37 - Média Final: 5.00  
Aluno Reprovado: Ana - Matrícula: 101 - Média Final: 5.50

Relatório Completo:  
Aluno: Maria - Matrícula: 26 - 1º Bimestre: 7 - 2º Bimestre: 6 - 3º Bimestre: 5 - 4º Bimestre: 8 - Média Final: 6.50 - Status: Aprovado  
Aluno: Ana - Matrícula: 101 - 1º Bimestre: 5 - 2º Bimestre: 4 - 3º Bimestre: 6 - 4º Bimestre: 7 - Média Final: 5.50 - Status: Reprovado

Ln 1, Col 1 Espaços: 4 UTF-8 CRLF Python 3.11.9 64-bit (Microsoft Store)

```
missao_pratica
10 def (variable) notas_primeiro_bimestre: dict[int, int]
12     notas_primeiro_bimestre = {
13         26: 7,
14         101: 5,
15         13: 8,
16         37: 4,
17         72: 7,
18         5: 9
19     }
20
21     notas_segundo_bimestre = {
22         26: 6,
23         101: 4,
24         13: 7,
25         37: 5,
26         72: 7,
27         5: 8
28     }
29
30     notas_terceiro_bimestre = {
31         26: 8,
32         101: 6,
33         13: 9,
34         37: 6,
35         72: 8,
36         5: 7
37     }
38
39     notas_quarto_bimestre = {
40         26: 9,
41         101: 7,
42         13: 6,
43         37: 7,
44         72: 9,
45         5: 6
46     }
47
48     return {
49         'primeiro': notas_primeiro_bimestre,
50         'segundo': notas_segundo_bimestre,
51         'terceiro': notas_terceiro_bimestre,
52         'quarto': notas_quarto_bimestre
53     }
54
55 def atualizar_dados_alunos(cadastro, notas):
56     """Atualiza os dados dos alunos com suas notas e calcula a média final."""
57     dados_alunos = {}
58
59     for matricula, nome in cadastro.items():
60         notas_aluno = [
61             notas['primeiro'].get(matricula, 0),
62             notas['segundo'].get(matricula, 0),
63             notas['terceiro'].get(matricula, 0),
64             notas['quarto'].get(matricula, 0)
65         ]
66
67         media_final = sum(notas_aluno) / len(notas_aluno)
68
69         dados_alunos[matricula] = {
70             'nome': nome,
71             'media_final': media_final
72         }
73
74     return dados_alunos
75
76 def main():
77     cadastro = {
78         26: 'Maria',
79         101: 'Ana',
80         13: 'João',
81         37: 'Agatha',
82         72: 'Joaquim',
83         5: 'Felix'
84     }
85
86     notas = inicializar_notas()
87
88     dados_alunos = atualizar_dados_alunos(cadastro, notas)
89
90     print_relatorio(dados_alunos)
91
92 if __name__ == '__main__':
93     main()
```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO **TERMINAL** PORTAS Python + - [ ] [ ] ... ^ x

Relatório Completo:  
Aluno: Maria - Matrícula: 26 - 1º Bimestre: 7 - 2º Bimestre: 6 - 3º Bimestre: 8 - 4º Bimestre: 9 - Média Final: 6.50 - Status: Aprovado  
Aluno: Ana - Matrícula: 101 - 1º Bimestre: 5 - 2º Bimestre: 4 - 3º Bimestre: 6 - 4º Bimestre: 7 - Média Final: 5.50 - Status: Reprovado  
Aluno: João - Matrícula: 13 - 1º Bimestre: 8 - 2º Bimestre: 7 - 3º Bimestre: 9 - 4º Bimestre: 6 - Média Final: 7.50 - Status: Aprovado  
Aluno: Agatha - Matrícula: 37 - 1º Bimestre: 4 - 2º Bimestre: 5 - 3º Bimestre: 6 - 4º Bimestre: 7 - Média Final: 5.50 - Status: Reprovado  
Aluno: Joaquim - Matrícula: 72 - 1º Bimestre: 7 - 2º Bimestre: 8 - 3º Bimestre: 9 - 4º Bimestre: 6 - Média Final: 7.00 - Status: Aprovado

```
missao_pratica_n3_estrutura_de_dados > main.py > ...
10 def inicializar_notas():
11     return {
12         'primeiro': (variable) notas_segundo_bimestre: dict[int, int]
13         'segundo': notas_segundo_bimestre,
14         'terceiro': notas_terceiro_bimestre,
15         'quarto': notas_quarto_bimestre
16     }
17
18 def atualizar_dados_alunos(cadastro, notas):
19     """Atualiza os dados dos alunos com suas notas e calcula a média final."""
20     dados_alunos = {}
21
22     for matricula, nome in cadastro.items():
23         notas_aluno = [
24             notas['primeiro'].get(matricula, 0),
25             notas['segundo'].get(matricula, 0),
26             notas['terceiro'].get(matricula, 0),
27             notas['quarto'].get(matricula, 0)
28         ]
29
30         media_final = sum(notas_aluno) / len(notas_aluno)
31
32         dados_alunos[matricula] = {
33             'nome': nome,
34             'media_final': media_final
35         }
36
37     return dados_alunos
38
39 def main():
40     cadastro = {
41         37: 'Agatha',
42         72: 'Joaquim',
43         5: 'Felix',
44         999: 'Novo Aluno'
45     }
46
47     notas = inicializar_notas()
48
49     dados_alunos = atualizar_dados_alunos(cadastro, notas)
50
51     print_relatorio(dados_alunos)
52
53 if __name__ == '__main__':
54     main()
```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO **TERMINAL** PORTAS Python + - [ ] [ ] ... ^ x

Aluno: Agatha - Matrícula: 37 - 1º Bimestre: 4 - 2º Bimestre: 5 - 3º Bimestre: 6 - 4º Bimestre: 7 - Média Final: 5.50 - Status: Reprovado  
Aluno: Joaquim - Matrícula: 72 - 1º Bimestre: 7 - 2º Bimestre: 8 - 3º Bimestre: 9 - 4º Bimestre: 6 - Média Final: 7.00 - Status: Aprovado  
Aluno: Felix - Matrícula: 5 - 1º Bimestre: 9 - 2º Bimestre: 8 - 3º Bimestre: 10 - 4º Bimestre: 9 - Média Final: 9.00 - Status: Aprovado  
Aluno: Novo Aluno - Matrícula: 999 - 1º Bimestre: 8 - 2º Bimestre: 7 - 3º Bimestre: 6 - 4º Bimestre: 7 - Média Final: 7.00 - Status: Aprovado  
PS C:\Users\event\OneDrive\Área de Trabalho\missao\_pratica\_n3\_estrutura\_de\_dados> pyinstaller --onefile -w

