



UNIVERSIDADE
VILA VELHA
ESPIRITO SANTO

Laboratório de programação

Lista de Exercício - LISTAS HOMOGÊNEAS E HETEROGÊNEAS

NOME: Nicolás De Paula Muciaccia - CC2N

MEU LINK COLAB: [Link](#)

✓ Exercício 1:

Escrever um programa que faça a Modelagem dos Dados, conforme o formulário a seguir, preenchendo os dados de cadastro de 100 funcionários da LISTA: EMPRESA. Após isso, faça:

FORMULÁRIO

Nome:

Cargo:

Matrícula: (Sortear um número entre [1000, 10000])

Plano de Saúde:

Salário(R\$):

- Exiba um relatório gerencial com a taxa de adesão (em porcentagem) ao Plano de Saúde.
- Exiba um relatório gerencial com os nomes dos funcionários que possuem Plano de Saúde.
- Exiba um relatório gerencial com os cargos dos funcionários que estão acima da média salarial.
- Exiba um relatório administrativo com o Total da Folha de Pagamento: Bruto e Líquido desta empresa descontando R\$ 212.54 dos funcionários que tem Plano de Saúde.

```
1 funcionario = []
2 empresa = []

1
2 print('MENU [0, 5]:')
3 print('INSIRA "0" PARA ENCERRAR O PROGRAMA:')
4 print('INSIRA "1" PARA INSERIR FUNCIONÁRIO:')
5 print('INSIRA "2" PARA VER TAXA DE ADESÃO AO PLANO DE SAÚDE:')
6 print('INSIRA "3" PARA VER OS FUNCIONÁRIO QUE TÊM PLANO DE SAÚDE:')
7 print('INSIRA "4" PARA VER OS FUNCIONÁRIOS COM SALÁRIO MAIOR QUE A MÉDIA SALARIAL:')
8 print('INSIRA "5" PARA VER TOTAL DA FOLHA DE PAGAMENTO BRUTO E LÍQUIDO(DESCONTO DE 212,54R$ PARA OS QUE TEM PLANO DE SAÚDE)')
9 while len(empresa) < 100:
10     try:
11         opcao = int(input('INSIRA SUA OPÇÃO: '))
12         if(opcao == 0):
13             print('PROGRAMA ENCERRADO!')
14             break
15         elif(opcao < 1 or opcao > 5):
16             print('ERRO, TENTE NOVAMENTE!')
17         elif(opcao == 1):
18             print(f'INSIRA OS DADOS DO {len(empresa) + 1}º FUNCIONÁRIO')
19             nome = input('NOME: ')
20             cargo = input('CARGO: ')
21             matricula = np.random.randint(1000, 9999)
22             # matricula = int(input('MATRÍCULA: '))
23             while True:
24                 try:
25                     plano = int(input('PLANO DE SAÚDE(1 p/ SIM e 0 p/ NÃO): '))
26                     if(plano < 0 or plano > 1):
27                         print('TENTE NOVAMENTE!')
28                     else:
29                         break
30                 except:
31                     print(f'ERRO: INSIRA NOVAMENTE')
32             while True:
33                 trv:
```

```

34     salario = float(input('SALÁRIO(R$): '))
35     break
36 except:
37     print(f'ERRO: INSIRA NOVAMENTE')
38 funcionario = [nome, cargo, matricula, plano, salario]
39 empresa.append(funcionario)
40 elif(opcao == 2):
41     try:
42         listaPlano = [funcionario[3] for funcionario in empresa]
43         print(f'TAXA DE ADESAO DO PLANO(%) É: {(sum(listaPlano) / len(listaPlano)) * 100: .2f}%)')
44     except:
45         print('VALORES NÃO ENCONTRADOS, NÃO HÁ FUNCIONÁRIOS CADASTRADOS!')
46 elif(opcao == 3):
47     listaFuncionarioPlano = [funcionario for funcionario in empresa if (funcionario[3] == 1)]
48     print(f'SÃO ESSES OS {len(listaFuncionarioPlano)} FUNCIONÁRIOS QUE POSSUEM PLANO DE SAÚDE:')
49     for indice, dados in enumerate(listaFuncionarioPlano):
50         print(f'{indice + 1}º: {dados}')
51 elif(opcao == 4):
52     try:
53         listaSalario = [funcionario[4] for funcionario in empresa]
54         listaSalarioMedia = [funcionario for funcionario in empresa if (funcionario[4] > (sum(listaSalario) / len(listaSalario)))]
55         print(f'UM TOTAL DE {len(listaSalarioMedia)} FUNCIONÁRIOS C/ SALÁRIO ACIMA DE {(sum(listaSalario) / len(listaSalario)): .2f}')
56         for indice, dados in enumerate(listaSalarioMedia):
57             print(f'{indice + 1}º: {dados}')
58     except:
59         print('VALORES NÃO ENCONTRADOS, NÃO HÁ FUNCIONÁRIOS CADASTRADOS!')
60 else:
61     totalBruto = [funcionario[4] for funcionario in empresa]
62     totalLiquido = [funcionario[4] for funcionario in empresa if (funcionario[3] == 1)]
63     print(f'FOLHA PAGAMENTO BRUTA : {sum(totalBruto): .2f}')
64     print(f'FOLHA PAGAMENTO LÍQUIDA: {sum(totalBruto) - (len(totalLiquido) * 212.54): .2f}')
65 except Exception as erro:
66     print(f'ERRO: {erro}')
67 print()

```

✓ Exercício 2:

Escrever um programa que faça a Modelagem dos Dados, conforme o formulário a seguir, preenchendo os dados de cadastro de vários atletas (Faça um MENU para o usuário) da LISTA: TRIATLON. Após isso, faça:

Nome:

Patrocinador:

Data Nascimento!

Dia: Mês: Ano:

Tempo(segundos)!

Natação: Corrida: Ciclismo:

- Exiba um relatório com os nomes dos triatletas que fizeram os melhores tempos nas etapas de natação, corrida e ciclismo.
- Exiba um relatório com o nome e patrocinador do melhor atleta entre todos.
- Exiba um relatório com o nome e o tempo dos atletas que fizeram um tempo abaixo da média.

```

1  atleta = []
2  nascimento = []
3  tempo = []
4  triatlon = []

1 print('MENU [0, 3]:')
2 print('INSIRA "0" PARA ENCERRAR O PROGRAMA:')
3 print('INSIRA "1" PARA INSERIR ATLETA:')
4 print('INSIRA "2" PARA VER OS ATLETAS COM MELHORES TEMPOS EM (natação, corrida e ciclismo):')
5 print('INSIRA "3" PARA VER O PATROCINADOR E O NOME DO MELHOR ATLETA:')
6 print('INSIRA "4" PARA VER OS ATLETAS(nome, tempo(segundos)) QUE FICARAM ABAIXO DA MÉDIA:')
7 while True:
8     try:
9         opcao = int(input('INSIRA SUA OPÇÃO [0, 4]: '))
10        if(opcao == 0):
11            print('PROGRAMA ENCERRADO!!')
12            break
13        elif(opcao < 1 or opcao > 4):
14            print('ERRO: VALORES INVÁLIDOS!!')
15        elif(opcao == 1):
16            print(f'INSIRA OS DADOS DO {len(triatlon) + 1}º ATLETA: ')
17            nome = input('NOME: ')

```

```

18 patrocinador = input('PATROCINADOR: ')
19 while True:
20     try:
21         dia = int(input('DIA DE NASCIMENTO: '))
22         if(dia < 0):
23             print('ERRO: INSIRA NOVAMENTE!')
24         else:
25             break
26     except:
27         print('ERRO: INSIRA NOVAMENTE!')
28 while True:
29     try:
30         mes = int(input('MÊS DE NASCIMENTO: '))
31         if(mes < 0):
32             print('ERRO: INSIRA NOVAMENTE!')
33         else:
34             break
35     except:
36         print('ERRO: INSIRA NOVAMENTE!')
37 while True:
38     try:
39         ano = int(input('ANO DE NASCIMENTO: '))
40         if(ano < 0):
41             print('ERRO: INSIRA NOVAMENTE!')
42         else:
43             break
44     except:
45         print('ERRO: INSIRA NOVAMENTE!')
46 while True:
47     try:
48         natacao = float(input('TEMPO NATAÇÃO(Segundos) : '))
49         if(natacao < 0):
50             print('ERRO: INSIRA NOVAMENTE!')
51         else:
52             break
53     except:
54         print('ERRO: INSIRA NOVAMENTE!')
55 while True:
56     try:
57         corrida = float(input('TEMPO CORRIDA(Segundos) : '))
58         if(corrida < 0):
59             print('ERRO: INSIRA NOVAMENTE!')
60         else:
61             break
62     except:
63         print('ERRO: INSIRA NOVAMENTE!')
64 while True:
65     try:
66         ciclismo = float(input('TEMPO CICLISMO(Segundos): '))
67         if(ciclismo < 0):
68             print('ERRO: INSIRA NOVAMENTE!')
69         else:
70             break
71     except:
72         print('ERRO: TENTE NOVAMENTE!')
73 nascimento = [dia, mes, ano]
74 tempo = [natacao, corrida, ciclismo]
75 atleta = [nome, patrocinador, nascimento, tempo]
76 triatlon.append(atleta)
77 elif(opcao == 2):
78     try:
79         listaTempos = [atleta[3] for atleta in triatlon]
80         listaNatacao = [tempo[0] for tempo in listaTempos]
81         indexNatacao = listaNatacao.index(min(listaNatacao))
82         print(f'MELHOR TEMPO DE NATAÇÃO FOI O {triatlon[indexNatacao][0]} DE {min(listaNatacao)} SEGUNDOS')
83         listaCorrida = [tempo[1] for tempo in listaTempos]
84         indexCorrida = listaCorrida.index(min(listaCorrida))
85         print(f'MELHOR TEMPO DE CORRIDA FOI O {triatlon[indexCorrida][0]} DE {min(listaCorrida)} SEGUNDOS')
86         listaCiclismo = [tempo[2] for tempo in listaTempos]
87         indexCiclismo = listaCiclismo.index(min(listaCiclismo))
88         print(f'MELHOR TEMPO DE CICLISMO FOI O {triatlon[indexCiclismo][0]} DE {min(listaCiclismo)} SEGUNDOS')
89     except:
90         print('NENHUM ATLETA ENCONTRADO, A LISTA ESTÁ VAZIA!')
91 elif(opcao == 3):
92     try:
93         listaTempos = [atleta[3] for atleta in triatlon]
94         listaTempoTotal = [(tempo[0] + tempo[1] + tempo[2]) for tempo in listaTempos]
95         indexCampeao = listaTempoTotal.index(min(listaTempoTotal))
96         print(f'O MELHOR ATLETA É {triatlon[indexCampeao][0]} E SEU PATROCINADOR É {triatlon[indexCampeao][1]}')
97     except:
98         print('NENHUM ATLETA ENCONTRADO, A LISTA ESTÁ VAZIA!')
99 else:

```

```
100     try:
101         listaTempos = [sum(atleta[3]) for atleta in triatlon]
102         listaFinal = [(atleta[0], sum(atleta[3])) for atleta in triatlon if(sum(atleta[3]) > (sum(listaTempos) / len(triatlon)))]
103         print(f'SEGUE A LISTA DOS ATLETAS QUE FICARAM ABAIXO DA MÉDIA DE {sum(listaTempos) / len(triatlon): .2f} SEGUNDOS: ')
104         for indice, atletas in enumerate(listaFinal):
105             print(f'{indice + 1}º: {atletas[0]} E {atletas[1]} SEGUNDOS')
106     except:
107         print('NENHUM ATLETA ABAIXO DA MÉDIA FOI ENCONTRADO, A LISTA ESTÁ VAZIA!')
108 except Exception as erro:
```