

ATIVIDADE PRÁTICA:

Lógica de Programação e Algoritmos

**ANO**

**2022**

|  |
| --- |
| **QUESTÃO 1 de 4 (25 pontos)**  **CONTEÚDO ATÉ AULA 03** |
| **Enunciado: Imagina-se que você é um dos programadores responsáveis pela construção de app de vendas para uma determinada empresa X que vende em atacado. Para calcular o valor total que o deve-se levar em consideração o custo de embalagem conforme a tabela abaixo**   |  |  | | --- | --- | | Quantidade | Custo de embalagem para frete | | 0 <= quantidade < 11 | **R$ 30.00** | | 11 <= quantidade < 101 | **R$ 60.00** | | 101 <= quantidade < 1001 | **R$ 120.00** | | quantidade >= 1001 | **R$ 240.00** |   **Elabore um programa em Python que:**   1. **Entre com o valor unitário do produto (Lembrar que número decimal é feito com PONTO e não VÍRGULA);** 2. **Entre com a quantidade desse produto;** 3. **O programa deve retornar o valor total sem o custo de embalagem para frete;** 4. **O programa deve retornar o valor total após o custo de embalagem para frete;** 5. **Deve-se utilizar estruturas if, elif e else (EXIGÊNCIA 1 de 1);** 6. **Colocar um exemplo de SAIDA DE CONSOLE de compra de mais de 1000 und.**   **Segue o exemplo de SAIDA DE CONSOLE:**    **Figura: Exemplo de programa com 1001 unidades sendo pedidas.** |

|  |
| --- |
| **Apresentação do Código (FORMATO TEXTO)** |
| print('Buenas! Bem vindo ao lojão Brique do Marcus Vinícius da Silva Debon!') print('Preço bom e frete diferenciado é conosco. Abaixo segue a tabela de preços...') print('Até 10 produtos - R$ 30.00') print('De 11 a 100 produtos - R$ 60.00') print('De 101 a 1000 produtos - R$ 120.00') print('Acima de 1000 produtos - R$ 240.00') print('Peço para que utilize o ponto (.) caso deseje informar valores decimais. Boas compras!') vp = float(input('Informe o valor de custo unitário do produto')) # Recebendo o valor do produto qp = int(input('informe a quantidade desejada do produto')) # Recebendo a quantidade  if qp <= 10:  f = 30 # frete 30 reais elif 11 <= qp <= 100:  f = 60 # frete 60 reais elif 101 <= qp <= 1000:  f = 120 # frete 120 reais else:  f = 240 # frete 240 reais  print('O valor total da compra sem frete é de {:.2f} R$'.format(vp \* qp)) print('O valor final com o custo do frete é de {:.2f}'.format(vp \* qp + f)) |

|  |
| --- |
| **Saída do Console (FORMATO IMAGEM – PRINT DA TELA)** |
| **Figura 1: Saída de console para 1200 produtos.** |

|  |
| --- |
| **QUESTÃO 2 de 4 (25 pontos)**  **CONTEÚDO ATÉ AULA 04** |
| **Enunciado: Você e sua equipe de programadores foram contratados para desenvolver um app de vendas para uma sorveteria. Você ficou com a parte de desenvolver a interface do cliente para retirada do produto.**  **A Sorveteria possui seguinte tabela com o código, descrição, tamanhos e os valores de sorvete:**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Código | Descrição | Tamanho P  (500 ml) | Tamanho M  (1000 ml) | Tamanho G  (2000 ml) | | TR | **Sabores Tradicionais** | **R$ 6,00** | **R$ 10,00** | **R$ 18,00** | | ES | **Sabores Especiais** | **R$ 7,00** | **R$ 12,00** | **R$ 21,00** | | PR | **Sabores Premium** | **R$ 8,00** | **R$ 14,00** | **R$ 24,00** |   **Elabore um programa em Python que:**   1. **Entre com o tamanho do pote de sorvete desejado;** 2. **Entre com o código do sorvete desejado;** 3. **Pergunte se o cliente quer pedir mais alguma coisa (se sim repetir a partir do item 1. Caso contrário ir para próximo passo);** 4. **Encerre a conta do cliente com o valor total;** 5. **Deve-se utilizar estruturas if, elif e else (EXIGÊNCIA 1 de 3);** 6. **Se a pessoa digitar um TAMANHO de sorvete e/ou código diferente dos da tabela printar na tela: ‘TAMANHO ou CÓDIGO inválido(s)’ e voltar para o menu (EXIGÊNCIA 2 de 3);** 7. **Deve-se utilizar while, break, continue (EXIGÊNCIA 3 de 3);**    * **(DICA: utilizar o continue dentro else que verifica a opção inválida)**    * **(DICA: utilizar o break dentro if que verifica a opção sair)** 8. **Colocar um exemplo de SAIDA DE CONSOLE com 3 (três) sorvetes** 9. **Colocar um exemplo de SAIDA DE CONSOLE com erro tamanho** 10. **Colocar um exemplo de SAIDA DE CONSOLE com erro código**   **Segue o exemplo de SAIDA DE CONSOLE:**    **Figura: Exemplo de programa com 3 sorvetes pedidos, 1 erro no pedido do tamanho e 1 erro no código do sorvete.** |

|  |
| --- |
| **Apresentação do Código (FORMATO TEXTO)** |
| print('Sorveteria do Marcus Vinícius da Silva Debon. Bora encher a barriga') print('-' \* 34,'OPÇÕES DE SORVETE','-' \* 34) print('| Código | Sabor do sorvete | Tamanho P 200ml | Tamanho M 350ml | Tamanho G 800ml |') print('| TR | Sabores Tradicionais | R$ 6,00 | R$ 10,00 | R$ 18,00 |') print('| ES | Sabores Especiais | R$ 7,00 | R$ 12,00 | R$ 21,00 |') print('| PR | Sabores Premium | R$ 8,00 | R$ 14,00 | R$ 24,00 |') print('-' \* 87) conta = 0  while True: # Arco de repetição para incluir pedidos até a opção de finalizar  ts = input('Informe o tamanho do sorvete conforme o código acima (p/m/g)') # tamanho do sorvete  cs = input('Informe o código do sorvete desejado (tr/es/pr)') # código do sorvete   if ts == 'p' and cs == 'tr':  conta += 6 # Sorvete pequeno - tradicional  elif ts == 'p' and cs == 'es':  conta += 7 # Sorvete pequeno - especial  elif ts == 'p' and cs == 'pr':  conta += 8 # Sorvete pequeno - premium  elif ts == 'm' and cs == 'tr':  conta += 10 # Sorvete médio - Tradicional  elif ts == 'm' and cs == 'es':  conta += 12 # Sorvete médio - especial  elif ts == 'm' and cs == 'pr':  conta += 14 # Sorvete médio - premium  elif ts == 'g' and cs == 'tr':  conta += 18 # Sorvete grande - Tradicional  elif ts == 'g' and cs == 'es':  conta += 21 # Sorvete grande - especial  elif ts == 'g' and cs == 'pr':  conta += 24 # Sorvete grande - premium  else:  print('Você digitou um tamanho ou código de sorvete inválido. Verifique novamente o seu pedido')  continue  tcfa = input('Deseja pedir mais alguma coisa? (responda s ou n)')  if tcfa != 's':  break # Opção por encerrar o pedido e fechar a conta  else:  continue # Opção por seguir inserindo sorvete para a turma  print('Feito o brique! Valor total da "sorveteada" de {:.2f} R$'.format(conta)) |

|  |
| --- |
| **Saída do Console (FORMATO IMAGEM – PRINT DA TELA)** |
| **Figura 1: Saída de console com pedido de 3 sorvetes.**    **Figura 2: Saída de console com erro no tamanho.**    **Figura 3: Saída de console com erro no código.** |

|  |
| --- |
| **QUESTÃO 3 de 4 (25 pontos)**  **CONTEÚDO ATÉ AULA 05** |
| **Enunciado: Imagina-se que você e sua equipe foram contratados por uma empresa preste serviços de limpeza para desenvolver a solução de software. Você ficou encarregado da parte de interação com o usuário.**  **O valor que a empresa cobra por limpeza é dado pela seguinte equação:**  **Em que cada uma das variáveis que compõe o preço total é quantizada da seguinte maneira:**   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Quadro 1: Metragem versus valor**   |  |  | | --- | --- | | Metragem (m²) | Valor (R$) | | 30 <= metragem < 300 | **60 + 0.3 \* metragem** | | 300 <= metragem < 700 | **120 + 0.5 \* metragem** | | Outros valores | **Não são aceitos** | | **Quadro 2: Tipo versus multiplicador**   |  |  | | --- | --- | | Tipo | Multiplicador | | B – Básica - Indicada para sujeiras semanais ou quinzenais | **1.00** | | C – Completa - Indicada para sujeiras antigas e/ou não rotineiras | **1.30** | | | **Quadro 3: Adicionais versus valor**   |  |  | | --- | --- | | Adicionais | Valor (R$) | | 0- Não desejo mais nada (encerrar) | **0,00** | | 1- Passar 10 peças de roupas - R$ 10.00 | **10,00** | | 2- Limpeza de 1 Forno/Micro-ondas - R$ 12,00 | **12,00** | | 3- Limpeza de 1 Geladeira/Freezer - R$ 20,00 | **20,00** | | |   **Elabore um programa em Python que:**   1. **Pergunte a metragem (em m²);**    * **Se o valor for 30 e 299,deve-se printar: “É necessário contratar 1 pessoa”**    * **Se o valor for 300 e 699, deve-se printar “É necessário contratar 2 pessoas”**    * **Se o valor passar 699, for menor que 30 ou for diferente de número; deve-se repetir a pergunta;** 2. **Pergunte a tipo de limpeza. Se digitar uma opção não válida deve repetir a pergunta** 3. **Pergunte o adicional. Deve-se perguntar ao usuário se desejada mais algum adicional até digitar ele 0** 4. **Encerre o total a ser pago com base na equação desse enunciado;** 5. **Deve-se codificar uma função metragem\_limpeza() (EXIGÊNCIA 1 de 3);**    * **Deve-se perguntar dentro da função a metragem da porção (em m²);**    * **Deve-se ter um if/else ou if/elif ou if/else/elif para verificar se o usuário não digitou uma metragem fora da faixa com que o empresa trabalha;**    * **Deve-se ter try/except para o caso do usuário digitar um valor não numérico;**    * **Deve-se retornar o valor em (RS) conforme a Quadro 1** 6. **Deve-se codificar uma função tipo\_limpeza() (EXIGÊNCIA 2 de 3);**    * **Deve-se perguntar dentro da função a opção desejada;**    * **Deve-se ter um if/elif/else para verificar as opções possíveis ou não;**    * **Deve-se retornar o multiplicador conforme o Quadro 2** 7. **Deve-se codificar uma função adicional\_limpeza() (EXIGÊNCIA 3 de 3);**    * **Deve-se perguntar dentro se deseja ou não mais algum adicional**    * **Deve-se ter um if/elif/else para verificar as opções possíveis ou não;**    * **Deve-se retornar o multiplicador conforme o Quadro 3** 8. **Colocar um exemplo de SAIDA DE CONSOLE com metragem inválida** 9. **Colocar um exemplo de SAIDA DE CONSOLE tipo de limpeza inválido** 10. **Colocar um exemplo de SAIDA DE CONSOLE metragem e opção válidos e mais 3 adicionais e no final deve ter o encerramento da conta com o descritivo de valor de cada varável (ver figura abaixo)**   **Segue o exemplo de SAIDA DE CONSOLE:**    **Figura: Exemplo de programa com metragem inválido e depois válido, tipo de limpeza inválido e depois válido, 3 adicionais e com total descritivo.** |
|  |

|  |
| --- |
| **Apresentação do Código (FORMATO TEXTO)** |
| # Função para cálculo do valor com base na área de limpeza def metragem\_limpeza():  print('\_'\*25,'Menu 1 de 3 - Área para limpeza','\_'\*25)  while True: # Arco contínuo para informar e retornar a área de limpeza, contínuo no caso de erro ao informar valores  try:  ar = int(input('Digite o valor da área total de limpeza entre 30 e 699 (limitação de serviço) em m²: \n' +  '>>'))  if (ar >= 30) and (ar < 300):  print('será necesário contratar uma pessoa para o serviço')  return 60 + ar \* 0.3  elif (ar >= 300) and (ar < 700):  print('será necesário contratar duas pessoas para o serviço')  return 120 + ar \* 0.5  else: # Retorno para o laço no caso de valor inteiro fora do limite determinado.  print('Valor digitado está fora do limite do nosso serviço. Por favor, digitar valor entre 30 e 699 m²')  continue  except ValueError: # Tratamento de erro para valores incorretos.  print('Não digitar números decimais ou letras. Apenas valores inteiros')  # Função para cálculo de multiplicador com base no tipo de limpeza def tipo\_limpeza():  print('\_'\*29,'Menu 2 de 3 - Tipo de limpeza','\_'\*29)  while True: # Laço para informar o tipo de limpeza desejada  tplp = input('Qual o tipo de limpeza para o seu imóvel? Digite uma das letras conforme modelo que segue\n'+  'B - Básica (Indicada para sujeiras semanais ou quinzenais) \n' +  'C - Completa (Indicada para sujeiras antigas e/ou não rotineiras) \n' +  '>> ')  tplp = tplp.upper() # tratamento para o caso de ser digitada letra minúscula  tplp = tplp.strip() # tratamento para o caso de ser digitados espaços  if tplp == 'B':  return 1.00 # Retorno para tipo de limpeza B  elif tplp == 'C':  return 1.30 # Retorno para tipo de limpeza C  else:  print('Digite apenas as letras B ou C conforme o modelo indicado')  continue # Retorno para o início do laço caso seja digitado valor fora do modelo  # Função dos adicionais do serviço de limpeza def adicional\_limpeza():  print('\_'\*16,'Menu 3 de 3 - Adicionais do serviço','\_'\*1)  valadc=0 # Variável para acumular valor total  while True: # Laço para incluir adicionais na limpeza  adc = input('Deseja algum adcional? Digite um dos números abaixo conforme o modelo \n' +  '0 - Não desejo mais nada (encerrar) \n' +  '1 - Passar 10 peças de roupas - R$ 10,00 \n' +  '2 - Limpeza de 1 Forno/Micro-ondas - R$ 12,00 \n' +  '3 - Limpeza de 1 Geladeira/Freezer - R$ 20,00 \n' +  '>> ')  if adc == '1':  valadc+=10  print('incluída a passagem de 10 peças de ropas')  elif adc == '2':  valadc+=12  print('incluída a limpeza de 1 Forno/Micro-ondas')  elif adc == '3':  valadc+=20  print('incluída a limpeza de 1 Geladeira/Freezer')  elif adc == '0':  print('OK. Estamos encerando o seu pedido')  break  else:  print('Digite apenas os números de 1 a 3 correspondente a cada opção \n' +  'ou 0 para encerrar o seu pedido')  return valadc # Função retornando o valor total dos adicionais contratados.    # Programa Principal print('--- Bem-vindo ao APP da Fast Clean, empresa de serviços de limpeza do empresário Marcus Vinícius da Silva Debon! ---') print('--- Contrate via internet, várias opções de pagamento e receba os funcionários na sua casa no horário desejado. ----') print('\*'\*116) area = metragem\_limpeza() tipo = tipo\_limpeza() adicionais = adicional\_limpeza() total = area \* tipo + adicionais print('\*'\*62) print('Resumindo o seu serviço temos: \n' +  'Valor da limpeza conforme a área e o tipo indicados de R$ {} \n' .format(area\*tipo) +  'Valor total dos adicionais selecionados de R$ {} \n' .format(adicionais) +  'Valor total do serviço de limpeza ficou por R$',total,'reais') |

|  |
| --- |
| **Saída do Console (FORMATO IMAGEM – PRINT DA TELA)** |
| **Figura 1: Saída de console demostrando os erros de metragem informados.**    **Figura 2: Saída de console demostrando os erros de tipo de limpeza informados.**    **Figura 3: Saída de console demostrando a execução completa do programa com a inclusão de 3 adicionais e detalhamento dos valores ao final.** |

|  |
| --- |
| **QUESTÃO 4 de 4 (25 pontos)**  **CONTEÚDO ATÉ AULA 06** |
| **Enunciado: Imagina-se que você está desenvolvendo um software de controle de funcionários para uma empresa de RH. Este software deve ter o seguinte menu e opções:**   1. **Cadastrar Funcionário** 2. **Consultar Funcionários(s)**    1. **Consultar Todas as Funcionários**    2. **Consultar Funcionário por Id**    3. **Consultar Funcionário(s) por Setor**    4. **Retornar** 3. **Remover Funcionário** 4. **Sair**   **Elabore um programa em Python que:**   1. **Deve-se codificar uma função cadastrar\_funcionario(id) (EXIGÊNCIA 1);**    * **Essa função recebe como parâmetro um id exclusivo para cada funcionário cadastrado (DICA: utilize um contador como parâmetro)**    * **Dentro da função perguntar o nome do funcionário;**    * **Dentro da função perguntar o setor do funcionário;**    * **Dentro da função perguntar o salário do funcionário**    * **Cada funcionário cadastrado deve ter os seus dados armazenados num DICIONÁRIO (DICA: Conferir material escrito da p. 22 até p24 da AULA 06)** 2. **Deve-se codificar uma função consultar\_funcionários()(EXIGÊNCIA 2);**    * **Dentro da função ter um menu com as seguintes opções:**      + **Consultar Todos os Funcionários**      + **Consultar Funcionário por Id**      + **Consultar Funcionário(s) por Setor**      + **Retornar** 3. **Deve-se codificar uma função remover\_funcionario() (EXIGÊNCIA 3);**    * **Dentro da função perguntar qual o código do funcionário que se deseja remover do cadastro (da lista de dicionário)** 4. **Colocar um exemplo de SAIDA DO CONSOLE com o cadastro de 3 (ou mais) funcionários . Sendo que 2 delas do mesmo setor – ver figura 1** 5. **Colocar um exemplo de SAIDA DO CONSOLE com a consulta a todos os funcionários cadastrados – ver figura 2** 6. **Colocar um exemplo de SAIDA DO CONSOLE com uma consulta por id – ver figura 3** 7. **Colocar um exemplo de SAIDA DO CONSOLE com uma consulta por setor – ver figura 4** 8. **Colocar um exemplo de SAIDA DO CONSOLE ao remover um funcionário cadastrado e mostrando depois todos os funcionários – ver figura 5**   **Segue os exemplos de SAIDA DE CONSOLE:**    **Figura 1: Exemplo de Cadastro de três funcionário, sendo que 2 deles são do mesmo setor.**    **Figura 2: Exemplo de Consulta Todos os Funcionários e pelo Id.**    **Figura 3: Exemplo de Consultar Funcionários pelo Setor.**    **Figura 4: Exemplo de Remover Funcionário do cadastro e depois Consultar Todos os Produtos. Veja que a produto de código 2 foi removido e não aparece mais no sistema.** |

|  |
| --- |
| **Apresentação do Código (FORMATO TEXTO)** |
| # Variáveis globais lista\_funcionario = [] # Lista que vai conter os dicionáiros de cada funcionário id\_funcionario = 0 # Variável para controle/contagem das IDs dos funcionários  # Função cadastrar funcionários def cadastrar\_funcionario(id): # Parâmetro selecionado - id  print('-'\*34)  print('Menu para cadastro de funcionários')  print('ID do Funcionário: {}'.format(id))  nome = input('Digite o nome do funcionário') # Atribuição do nome  setor = input('Digite o setor do funcionário') # Atribuição do setor  salario = input('Digite o salário do funcionário') # Atribuição do salário  dicionario\_funcionario = {'ID' : id, # Formatação do dicionário do funcionário  'nome' : nome,  'setor' : setor,  'salário' : salario}  lista\_funcionario.append(dicionario\_funcionario.copy())  print('-' \* 22)  print('Funcionário cadastrado') # Função consultar funcionários def consultar\_funcionario():  print('-' \* 34)  print('Menu para consulta de funcionários')  while True:  funcaoconsulta = input('Selecione uma das opções conforme segue: \n'+  '1. Consultar Todos os Funcionários \n' +  '2. Consultar Funcionário por ID \n' +  '3. Consultar Funcionário(s) por Setor \n' +  '4. Retornar \n' +  '>>')  if funcaoconsulta == '1':  print('-' \* 37)  print('Opção Consultar Todos os Funcionários')  for funcionario in lista\_funcionario:  print('-'\*20)  for key, value in funcionario.items(): # Varre todos os conjuntos chave e valor do dicionario funcionarios  print('{}: {}' .format(key,value))  print('-' \* 20)  elif funcaoconsulta == '2':  print('-' \* 34)  print('Opção Consultar Funcionário por ID')  try:  val\_consulta = int(input('Digite o ID do funcionário para consulta: '))  for funcionario in lista\_funcionario:  if funcionario['ID'] == val\_consulta: # Valor do campo ID no diciónario é igual ao ID de consulta?  print('-' \* 20)  for key, value in funcionario.items(): # Varre todos os conjuntos chave e valor do dicionario funcionarios  print('{}: {}'.format(key, value)) # Imprime caso a ID digitada for igual a consultada no laço  print('-' \* 20)  except ValueError: # Tratamento para valores digitados diferentes de inteiro  print('-' \* 46)  print('Valor inválido. Digite apenas números inteiros.')  print('-' \* 46)  elif funcaoconsulta == '3':  print('-' \* 40)  print('Opção Consultar Funcionário(s) por Setor')  val\_consulta = input('Digite o Setor do funcionário para consulta: ')  for funcionario in lista\_funcionario:  if funcionario['setor'] == val\_consulta: # Valor do campo Setor no diciónario é igual ao ID de consulta?  print('-' \* 20)  for key, value in funcionario.items(): # Varre todos os conjuntos chave e valor do dicionario funcionarios  print('{}: {}'.format(key, value)) # Imprime caso o setor digitado for igual ao consultado no laço  print('-' \* 20)  elif funcaoconsulta == '4':  print('Retornando ao Menu Principal...')  return  else:  print('-' \* 69)  print('Valor incorreto. Digite um número de 1 a 4 conforme o modelo exposto.')  print('-' \* 69)  continue  # Função remover funcionários def remover\_funcionario():  print('-' \* 33)  print('Menu para remoção de funcionários')  try:  val\_consulta = int(input('Entre com o ID do funcionário que deseja remover'))  for funcionario in lista\_funcionario:  if funcionario['ID'] == val\_consulta:  lista\_funcionario.remove(funcionario) # Removendo o funcionário indicado do cadastro  print('-' \* 32)  print('Funcionario removido do cadastro.')  print('-' \* 32)  except ValueError: # Tratamento para valores digitados diferentes de inteiro  print('-' \* 46)  print('Valor inválido. Digite apenas números inteiros.')  print('-' \* 46)   # Início do programa principal print('-'\*25,'Bem-vindo ao Registro de empregados da empresa.','-'\*25,' \n' +  'Desenvolvido em parceria com a empresa Get Over Here! do empresário Marcus Vinícius da Silva Debon.') while True: # Laço inicial expondo as funções do programa  print('\*'\*40)  funcao = input('Selecione uma das opções conforme segue: \n'+  '1. Cadastrar Funcionário \n' +  '2. Consultar Funcionário(s) \n' +  '3. Remover Funcionário \n' +  '4. Sair \n' +  '>>')  if funcao == '1':  id\_funcionario = id\_funcionario + 1 # Incremento para cadastrar o próximo funcionario da lista  cadastrar\_funcionario(id\_funcionario)  elif funcao == '2':  consultar\_funcionario() # Função consulta ativada  elif funcao == '3':  remover\_funcionario() # Função remover ativada  elif funcao == '4':  print('-' \* 37)  print('Saindo do programa. Tenha um bom dia!')  print('-' \* 37)  break  else: # Orientação para informar valores corretos  print('-' \* 69)  print('Valor incorreto. Digite um número de 1 a 4 conforme o modelo exposto.')  print('-' \* 69)  continue |

|  |
| --- |
| **Saída do Console (FORMATO IMAGEM – PRINT DA TELA)** |
| **Figuras 1 e 2: Saída de console - cadastro de 3 funcionários, sendo 2 no mesmo setor.**    **Figura 3: Saída de console com consulta de todos os funcionários.**    **Figura 4: Saída de console com consulta por ID.**    **Figura 5: Saída de console com consulta por Setor.**    **Figura 6: Saída de console com a remoção de um funcionário e posterior consulta de todos os funcionários.** |