

ATIVIDADE PRÁTICA:

Lógica de Programação e Algoritmos

**ANO**

**2023**

|  |
| --- |
| **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. Uma das estratégias de vendas dessa empresa X é dar desconto maiores por unidade conforme a tabela abaixo:**   |  |  | | --- | --- | | Quantidades | Desconto | | Até 9 | **0% na unidade** | | Entre 10 e 99 | **5% na unidade** | | Entre 100 e 999 | **10% na unidade** | | De 1000 para mais | **15% na unidade** |   **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 desconto;** 4. **O programa deve retornar o valor total após o desconto;** 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 10 und. (para mostrar que o desconto foi aplicado)**   **Segue o exemplo de SAIDA DE CONSOLE:**    Colocar o seu nome  **OBS: Para os números decimais ficarem com somente duas casas depois da vírgula utilize {:.2f). Exemplo:**  print('O valor sem desconto foi: R$ {:.2f}' .format(sub\_total)) |

|  |
| --- |
| **Apresentação do Código (FORMATO TEXTO)** |
| print('Bem vindo a loja do Vinicius Alves Gotti Nascimento') # identificador pessoal    valor\_unitario = float(input("Digite o valor unitário do produto: ")) # valor de cada item  quantidade = int(input("Digite a quantidade do produto: ")) # quantidade de itens comprados  desconto = 0    valor\_total\_sem\_desconto = valor\_unitario \* quantidade # calcula o valor sem o desconto    if quantidade < 10: # numero de itens comprados    desconto = 0    elif 10 <= quantidade <= 99: # numero de itens comprados    desconto = 0.05 # % de desconto    elif 100 <= quantidade <= 999: # numero de itens comprados    desconto = 0.10 # % de desconto    else:    desconto = 0.15 # % de desconto    valor\_total\_com\_desconto = valor\_total\_sem\_desconto - (valor\_total\_sem\_desconto \* desconto) # calcula o valor com desconto    desconto\_em\_porcentagem = desconto \*100    print('Valor total sem desconto: R$ {:.2f}'.format(valor\_total\_sem\_desconto))  print('Valor total com desconto: R$ {:.2f}'.format(valor\_total\_com\_desconto))  print('Desconto: {:.0f}%'.format(desconto\_em\_porcentagem)) # apresenta quantos % foi o desconto no resultado final |

|  |
| --- |
| **Saída do Console (FORMATO IMAGEM – PRINT DA TELA)** |
| Print do resultado no console |

|  |
| --- |
| **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 lanchonete. Você ficou com a parte de desenvolver a interface do cliente para retirada do produto.**  **A lanchonete possui seguinte tabela de produtos listados com sua descrição, códigos e valores:**   |  |  |  | | --- | --- | --- | | Código | Descrição | Valor(R$) | | 100 | **Cachorro-Quente** | **9,00** | | 101 | **Cachorro-Quente Duplo** | **11,00** | | 102 | **X-Egg** | **12,00** | | 103 | **X-Salada** | **13,00** | | 104 | **X-Bacon** | **14,00** | | 105 | **X-Tudo** | **17,00** | | 200 | **Refrigerante Lata** | **5,00** | | 201 | **Chá Gelado** | **4,00** |   **Elabore um programa em Python que:**   1. **Entre com o código do produto desejado;** 2. **Pergunte se o cliente quer pedir mais alguma coisa (se sim repetir o passo item 2. Caso contrário ir para próximo passo);** 3. **Encerre a conta do cliente com o valor total;** 4. **Deve-se utilizar estruturas if, elif e else (EXIGÊNCIA 1 de 3);** 5. **Se a pessoa digitar um NÚMERO diferente dos da tabela printar na tela: ‘opção inválida’ e voltar para o menu (EXIGÊNCIA 2 de 3);** 6. **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 elif que verifica a opção sair)** 7. **Colocar um exemplo de SAIDA DE CONSOLE com dois pedidos** 8. **Colocar um exemplo de SAIDA DE CONSOLE com erro ao digitar no pedido**   **Segue o exemplo de SAIDA DE CONSOLE:**    **Figura: Exemplo de programa com 2 itens pedido (Sendo que um deles teve uma tentativa com erro) e no final o valor final é apresentado.** |

|  |
| --- |
| **Apresentação do Código (FORMATO TEXTO)** |
| print('Bem vindo a Lanchonete do Vinícius Alves Gotti Nascimento') # identificador pessoal  print('-' \* 15 , 'CARDAPIO' , '-' \* 15) #visual do cardápio  print('|CÓDIGO| DESCRIÇÃO | VALOR(R$) |')  print('-' \* 45)  print('| 100 | Cachorro-Quente | 9,00 |')  print('| 101 | Cachorro-Quente Duplo | 11,00 |')  print('| 102 | X-Egg | 12,00 |')  print('| 103 | X-Salada | 13,00 |')  print('| 104 | X-Bacon | 14,00 |')  print('| 105 | X-Tudo | 17,00 |')  print('| 200 | Refrigerante Lata | 5,00 |')  print('| 201 | Chá Gelado | 4,00 |')  print('-' \* 45)    acumulador = 0 # soma os valores dos produtos    # dicionario de itens  nomes = {'100':'Cachorro-Quente' ,  '101': 'Cachorro-Quente Duplo',  '102': 'X-Egg',  '103': 'X-Salada',  '104': 'X-Bacon',  '105': 'X-Tudo',  '200': 'Refrigerante Lata',  '201': 'Chá Gelado'}  codigos = {'100',  '101',  '102',  '103',  '104',  '105',  '200',  '201'}    while True: # código de repetição  codigo = input('Digite o código do produto desejado: ')  if codigo not in codigos: # se o código não estiver dentro dos codigos apresentados no dicionario    print('Opção inválida. Utilize um dos códigos de produto acima.') # para caso não corresponda aos números programados esta mensagem aparece  continue # se o usuário digitar algo inválido, volta para o começo    if codigo == '100' : # se o código digitado for tal número, imprime o produto que acabou de escolher e soma o valor dele no acumulador  print('Você escolheu o produto: {}'.format(nomes[codigo]))  acumulador = acumulador + 9.00 # ir somando os valores dos produtos  elif codigo == '101' :  print('Você escolheu o produto: {}'.format(nomes[codigo]))  acumulador = acumulador + 11.00  elif codigo == '102' :  print('Você escolheu o produto: {}'.format(nomes[codigo]))  acumulador = acumulador + 12.00  elif codigo == '103' :  print('Você escolheu o produto: {}'.format(nomes[codigo]))  acumulador = acumulador + 13.00  elif codigo == '104' :  print('Você escolheu o produto: {}'.format(nomes[codigo]))  acumulador = acumulador + 14.00  elif codigo == '105' :  print('Você escolheu o produto: {}'.format(nomes[codigo]))  acumulador = acumulador + 17.00  elif codigo == '200' :  print('Você escolheu o produto: {}'.format(nomes[codigo]))  acumulador = acumulador + 5.00  elif codigo == '201' :  print('Você escolheu o produto: {}'.format(nomes[codigo]))  acumulador = acumulador + 4.00    pedir\_mais = input('Deseja pedir mais alguma coisa? (Responda com S/N) ')  pedir\_mais = pedir\_mais.upper() # digitar 's' maiúsculo ou minúsculo não muda o resultado  if pedir\_mais == "S" :  continue # volta pro começo  else:  print('O valor total a ser pago: R$ {:.2f}' .format (acumulador))  break |

|  |
| --- |
| **Saída do Console (FORMATO IMAGEM – PRINT DA TELA)** |
| Print do resultado no console |
| **QUESTÃO 3 de 4 (25 pontos)**  **CONTEÚDO ATÉ AULA 05** |
| **Enunciado: Imagina-se que você e sua equipe foram contratados por uma empresa de logística que acabou de entrar no ramo. Essa empresa trabalha com encomendas de pequeno e médio porte e opera somente entre 3 cidades.**  **O valor que a empresa cobra por objeto é dado pela seguinte equação:**  **que você e sua equipe**  **Em que cada uma das variáveis que compõe o preço total é quantizada da seguinte maneira:**     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Quadro 1: Dimensões versus Valor**   |  |  | | --- | --- | | dimensões (cm³) | valor (R$) | | volume < 1000 | **10** | | 1000 <= volume < 10000 | **20** | | 10000 <= volume < 30000 | **30** | | 30000 <= volume < 100000 | **50** | | volume >= 100000 | **Não é aceito** | | **Quadro 2: Peso versus multiplicador**   |  |  | | --- | --- | | peso(kg) | multiplicador | | peso <= 0.1 | **1** | | 0.1 <= peso < 1 | **1.5** | | 1 <= peso < 10 | **2** | | 10 <= peso < 30 | **3** | | peso => 30 | **Não é aceito** | | | **Quadro 3: Rota versus multiplicador**   |  |  | | --- | --- | | rota | multiplicador | | RS - De Rio de Janeiro até São Paulo | **1** | | SR - De São Paulo até Rio de Janeiro | **1** | | BS - De Brasília até São Paulo | **1.2** | | SB - De São Paulo até Brasília | **1.2** | | BR - De Brasília até Rio de Janeiro | **1.5** | | RB - Rio de Janeiro até Brasília | **1.5** |   **Obs.: Pode-se mudar o nome das cidades e siglas. Utilizando 3 cidades está ótimo** | |   **Elabore um programa em Python que:**   1. **Pergunte a altura (em cm), comprimento (em cm) e largura (em cm) do objeto. Se digitar um valor não numérico e/ou as dimensões passarem do limite aceito repetir a pergunta;** 2. **Pergunte o peso do objeto (em kg). Se digitar um valor não numérico e/ou o peso passar do limite aceito repetir a pergunta;** 3. **Pergunte a rota do objeto. Se digitar uma opção que não esteja na tabela repetir a pergunta;** 4. **Encerre o total a ser pago com base na equação desse enunciado;** 5. **Deve-se codificar uma função dimensoesObjeto (EXIGÊNCIA 1 de 3);**    * **Dentro da função perguntar altura do objeto (em cm);**    * **Dentro da função perguntar o comprimento do objeto (em cm);**    * **Dentro da função perguntar a largura do objeto (em cm)**    * **Calcular o volume (em cm) da caixa p/a objeto (altura\*largura\*comprimento);**    * **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 pesoObjeto (EXIGÊNCIA 2 de 3);**    * **Dentro da função perguntar peso do objeto (em kg);**    * **Deve-se ter um try/except para o caso de o usuário digitar um valor não numérico;**    * **Deve-se retornar o multiplicador conforme o Quadro 2** 7. **Deve-se codificar uma função rotaObjeto (EXIGÊNCIA 3 de 3);**    * **Dentro da função perguntar a rota do objeto desejada (Sugestão: utilize as siglas para facilitar os testes);**    * **OBS: PODE MUDAR O NOME DAS CIDADES E SUAS SIGLAS**    * **Deve-se retornar o multiplicador conforme o Quadro 3** 8. **Colocar um exemplo de SAIDA DE CONSOLE uma encomenda com peso, dimensões e rota válidos** 9. **Colocar um exemplo de SAIDA DE CONSOLE com o tratamento de erro quando digitado um valor não numérico é digitado no campo peso ou dimensões)**   **Segue o exemplo de SAIDA DE CONSOLE:**  Colocar o seu nome    **Figura: Exemplo de programa com tratamento de valor e erro de digitação** |
| **Apresentação do Código (FORMATO TEXTO)** |
| #---------- Início da função dimensoesObjeto() ----------  def dimensoesObjeto():  print('-' \* 12 , 'Menu 1 de 3 - Dimensão Objeto', '-' \* 12)  while True:  try:  alturacm = int(input('Digite a altura em centimetros(cm) desejada: ')) # pede a altura em cm  comprimentocm = int(input('Digite o comprimento em centimetros(cm) desejado: ')) # pede o comprimento em cm  larguracm = int(input('Digite a largura em centimetros(cm) desejado: ')) # pede a largura em cm  volume = print('Esse é o volume do objeto em centimetros(cm): {} '.format (alturacm \* comprimentocm \* larguracm))  volume\_produto = alturacm \* comprimentocm \* larguracm # calcula o volume  dimensoesObjeto = volume\_produto    if volume\_produto <= 1000 : # precifica de acordo com o volume  print('O valor é R$10.00')  valor = 10  return valor  elif 1000 <= volume\_produto <= 10000 : # precifica de acordo com o volume  print('O valor é R$20.00')  valor = 20  return valor  elif 10000 <= volume\_produto <= 30000 : # precifica de acordo com o volume  print('O valor é R$30.00')  valor =30  return valor  elif 30000 <= volume\_produto <= 100000 : # precifica de acordo com o volume  print('O valor é R$50.00')  valor = 50  return valor  else:  print('Não é aceito.')  continue  return dimensoesObjeto  except ValueError: # Mostra para caso se digite pontos flutuantes ou letras ao invés de números inteiros  print('Não digite valores não inteiros.')        #---------- Fim da função dimensoesObjeto() ----------    #---------- Início da função pesoObjeto() ----------  def pesoObjeto(valor):  print('-' \* 12 , 'Menu 2 de 3 - Peso Objeto', '-' \* 12)  while True:  try:  pesokg = int(input('Digite o peso do objeto em kg: '))    if pesokg <= 0.1 :  return valor # volume da dimensao \* 1  elif 0.1 <= pesokg <= 1 :  return valor \* 1.5 # volume da dimensao \* 1.5  elif 2 <= pesokg <= 10 :  return valor \*2 # volume da dimensao \* 2  elif 11 <= pesokg <= 30 :  return valor \*3 # volume da dimensao \* 3  else:    print('Não é aceito.')  continue    except ValueError: # Mostra para caso se digite pontos flutuantes ou letras ao invés de números inteiros  print('Não digite valores não inteiros.')      #---------- Fim da função pesoObjeto() ----------    #---------- Início Função rotaObjeto() ----------  def rotaObjeto(peso):  print('-' \* 12, 'Menu 3 de 3 - Rota', '-' \* 12)  while True:  rotaopcoes = input( # menu com as rotas  'Qual opção de rota deseja?\n'  'RS - De Rio de Janeiro até São Paulo\n'  'SR - De São Paulo até Rio de Janeiro\n'  'BS - De Brasília até São Paulo\n'  'SB - De São Paulo até Brasília\n'  'BR - De Brasília até Rio de Janeiro\n'  'RB - Rio de Janeiro até Brasília\n'  '').upper().strip()    if rotaopcoes == 'RS' or rotaopcoes == 'SR': #opções de rotas  valor = peso \* 1  print()  print(f'O valor é de R${valor:.2f}')  print()  return valor  elif rotaopcoes == 'BS' or rotaopcoes == 'SB':  valor = peso \* 1.2  print()  print(f'O valor é de R${valor:.2f}')  print()  return valor  elif rotaopcoes == 'BR' or rotaopcoes == 'RB':  valor = peso \* 1.5  print()  print(f'O valor é de R${valor:.2f}')  print()  return valor  else:  print('Você digitou uma rota que não existe.')  continue      #---------- Fim da função rotaObjeto() ----------    #---------- Inicio do Main ----------  print('-' \* 12, 'Bem vindo a companhia de logística de Vinícius Alves Gotti Nascimento', '-' \* 12) # identificador Pessoal    dimensao = dimensoesObjeto()    peso = pesoObjeto(dimensao)    rota = rotaObjeto(peso) |

|  |
| --- |
| **Saída do Console (FORMATO IMAGEM – PRINT DA TELA)** |
| Print do resultado no console |

|  |
| --- |
| **QUESTÃO 4 de 4 (25 pontos)**  **CONTEÚDO ATÉ AULA 06** |
| **Enunciado: Imagina-se que você está desenvolvendo um software de controle de estoque para uma bicicletaria. Este software deve ter o seguinte menu e opções:**   1. **Cadastrar Peça** 2. **Consultar Peça**    1. **Consultar Todas as Peças**    2. **Consulta Peças por Código**    3. **Consulta Peças por Fabricante**    4. **Retornar** 3. **Remover Peça** 4. **Sair**   **Elabore um programa em Python que:**   1. **Deve-se codificar uma função cadastrarPeca (código) (EXIGÊNCIA 1);**    * **Essa função recebe como parâmetro um código exclusivo para cada peça cadastrado (DICA: utilize um contador como parâmetro)**    * **Dentro da função perguntar o nome da peça;**    * **Dentro da função perguntar o fabricante da peça;**    * **Dentro da função perguntar o valor da peça**    * **Cada peça cadastrada 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 consultarPeca(EXIGÊNCIA 2);**    * **Dentro da função ter um menu com as seguintes opções:**      + **Consultar Todas as Peças**      + **Consultar Peças por Código**      + **Consultar Peças por Fabricante**      + **Retornar** 3. **Deve-se codificar uma função chamada removerPeca (EXIGÊNCIA 3);**    * **Dentro da função perguntar qual o código do produto 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) peças. Sendo que 2 delas do mesmo fabricante – ver figura 1** 5. **Colocar um exemplo de SAIDA DO CONSOLE com a consulta a todas as peças cadastradas – ver figura 2** 6. **Colocar um exemplo de SAIDA DO CONSOLE com uma consulta por código – ver figura 3** 7. **Colocar um exemplo de SAIDA DO CONSOLE com uma consulta por fabricante – ver figura 4** 8. **Colocar um exemplo de SAIDA DO CONSOLE ao remover um cadastro e mostrando depois todos os cadastros– ver figura 5**   **Segue os exemplos de SAIDA DE CONSOLE:**    **Figura 1: Exemplo de Cadastrar de três peças. Perceba que 2 delas tem o mesmo fabricante.**    **Figura 2: Exemplo de Consulta Todas as Peças.**    **Figura 3: Exemplo de Consultar por Código.**    **Figura 4: Exemplo de Consultar Peças por Fabricante**    **Figura 5: Exemplo de Remover Peça do cadastro e depois Consultar Todas as Peças. Veja que a peça de código 1 foi removida e não aparece mais no sistema.** |

|  |
| --- |
| **Apresentação do Código (FORMATO TEXTO)** |
| #---------- Início das Variáveis Globais ----------  lista\_peca = []  codigo\_peca = 0  #---------- Fim das Variáveis Globais ----------    #---------- Início de cadastrar\_peca() ----------  def cadastrar\_peca(codigo):  print('Bem-vindo ao menu de Cadastrar Peça')  print('Código da Peça: {}'.format(codigo))  nome = input ('Entre com o NOME da peça:')  fabricante = input ('Entre com o FABRICANTE da peça:')  valor = int(input ('Entre com o VALOR(R$) da peça:'))  dicionario\_peca = {'codigo' : codigo, # dicionario para identificação dos itens  'nome' : nome,  'fabricante' : fabricante,  'valor' : valor}  lista\_peca.append (dicionario\_peca.copy())  #---------- Fim de cadastrar\_peca() ----------    #---------- Início de consultar\_peca() ----------  def consultar\_peca():  print('Bem-vindo ao menu de Consultar Peça')  while True:  opcao\_consultar = input('\nEscolha a opção desejada:\n'+ # menu para consulta de peças  '1-Consultar TODAS as Peças\n'+  '2-Consultar Peça por CÓDIGO\n'+  '3-Consultar Peça(s) por FABRICANTE\n'+  '4-Retornar\n'+  '>>')  if opcao\_consultar == '1':  print('Você escolheu a opção Consultar TODAS as Peças')  for peca in lista\_peca: # peca vai varrer toda a lista de peças  print('-------------------------') # melhoria para a visualização do resultado  for key, value in peca.items(): # varrer todos os conjuntos chave e valor do dicionario peca  print('{}: {}'.format(key,value))  print('-------------------------') # melhoria para a visualização do resultado  elif opcao\_consultar == '2':  print('Você escolheu a opção Consultar Peça por CÓDIGO')  valor\_desejado = int(input('Entre com o CÓDIGO desejado: ')) # codigo é uma variavel do tipo inteira, caso contrario não achara listagem  for peca in lista\_peca:  if peca['codigo'] == valor\_desejado: # o valor do campo código desse dicionário é igual ao valor desejado  print('-------------------------')  for key, value in peca.items(): # varrer todos os conjuntos chave e valor do dicionario peca  print('{}: {}'.format(key,value))  print('-------------------------')  elif opcao\_consultar == '3':  print('Você escolheu a opção Consultar Peça(s) por FABRICANTE')  valor\_desejado = input('Entre com o FABRICANTE desejado: ')  for peca in lista\_peca:  if peca['fabricante'] == valor\_desejado: # o valor do campo fabricante desse dicionário é igual ao valor desejado  print('-------------------------')  for key, value in peca.items(): # varrer todos os conjuntos chave e valor do dicionario peca  print('{}: {}'.format(key,value))  print('-------------------------')  elif opcao\_consultar == '4':  return # sai da função consultar e retorna para o MAIN (menu principal)  else:  print('Opção Inválida. Tente Novamente') # para caso não corresponda aos números programados esta mensagem aparece  continue # volta para o inicio do laço  #---------- Fim de consultar\_peca() ----------    #---------- Início de remover\_peca() ----------  def remover\_peca():  print('Bem-vindo ao menu de Remover Peça')  valor\_desejado = int(input('Entre com o CÓDIGO da peça que deseja remover: ')) # estrutura parecida com as anteriores, codigo deve ter uma variavel do tipo inteira  for peca in lista\_peca:  if peca['codigo'] == valor\_desejado: # o valor do campo código desse dicionário é igual ao valor desejado  lista\_peca.remove(peca)  print('Peça Removida') # confirmação de remoção da peça    #---------- Fim de remover\_peca() ----------    #---------- Início de Main ----------  print('Bem-vindo a Bicicletaria do Vinícius Alves Gotti Nascimento') # identificador pessoal  while True:  opcao\_principal = input('\nEscolha a opção desejada:\n'+ # menu principal  '1-Cadastrar Peça\n'+  '2-Consultar Peça\n'+  '3-Remover Peça\n'+  '4-Sair\n'+  '>>')  if opcao\_principal == '1':  codigo\_peca = codigo\_peca + 1 # para diferenciar as peças cadastradas ( peça 1 / peça 2 / peça 3 / ... )  cadastrar\_peca(codigo\_peca)  elif opcao\_principal == '2':  consultar\_peca()  elif opcao\_principal == '3':  remover\_peca()  elif opcao\_principal == '4':  break # encerra o laço principal e o programa acaba  else:  print('Opção Inválida. Tente Novamente') # para caso não corresponda aos números programados esta mensagem aparece  continue # volta para o inicio do laço  #---------- Fim de Main ---------- |

|  |
| --- |
| **Saída do Console (FORMATO IMAGEM – PRINT DA TELA)** |
| Figura 1 – Cadastrar as peças    Figura 2 – Consultar todas da peças    Figura 3 – Consultar peça por código    Figura 4 – Consultar peça por fabricante    Figura 5 – Remover Peça do cadastro e depois Consultar Todas as Peças |