

ATIVIDADE PRÁTICA:

Lógica de Programação e Algoritmos

**ANO**

**2022**

Aluno: Gabriel Pinto Iório

|  |
| --- |
| **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 4 | **0% na unidade** | | Entre 5 e 19 | **3% na unidade** | | Entre 20 e 99 | **6% na unidade** | | Maior ou igual a 100 | **10% 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.**   **Segue o exemplo de SAIDA DE CONSOLE:** |

|  |
| --- |
| **Apresentação do Código (FORMATO TEXTO)** |
| # EXERCICIO 1  n = "Gabriel Pinto Iório / RU: 4179744"  # => IDENTIFICADOR PESSOAL, NOME E RU  print(*f*"Seja Bem-vindo a Loja do {n}")  *def* FunValor(): # Função que solicita e valida o valor do produto      while True:          try:              valor = *float*(input("Entre com o valor do produto: ").replace(",", ".")) # entrando com os dados do produto              if valor <= 0:                  print("\nDigite um valor maior que zero.\n")              else:                  break          except ValueError:              print("\nDigite um valor válido.\n")      return valor  *def* Quantidade(): # Função que solicita e valida a quantidade do produto      while True:          try:              quantidade = *float*(input("Entre com a quantidade do produto: ")) # entrando com os dados de quantidade do produto              if quantidade <= 0: # caso o usuario digite um numero menor que zero                  print("\nDigite uma quantidade maior que zero.\n")              else:                  break          except ValueError: # caso nao siga a orientação              print("\nDigite uma quantidade válida.\n")      return quantidade  *def* ParagetDesconto(*quantidade*: *float*): # Função que calcula o desconto        if quantidade > 0 and quantidade < 5: # se o valor da quantidade for maior que 0 e menor que 5, a variavel desconto recebe o valor 0          desconto = 0      elif quantidade >= 5 and quantidade < 20: # outras condições que seguem o mesmo raciocinio          desconto = 3      elif quantidade >= 20 and quantidade < 100:          desconto = 6      elif quantidade >= 100:          desconto = 10      else:          raise Exception("Não há desconto cadastrado para a quantidade informada.")      return desconto  # Entrada de dados  valor = FunValor()  quantidade = Quantidade()  # Obtenção do desconto  desconto = ParagetDesconto(quantidade)  # Cálculo dos valores SEM e COM desconto  valor\_sem\_desconto = valor \* quantidade  valor\_com\_desconto = valor\_sem\_desconto - (valor\_sem\_desconto \* desconto / 100)  # Impressão dos valores  print("")  print(*f*"O valor sem desconto foi: R$ {valor\_sem\_desconto:.2f}")  print(*f*"O valor com desconto foi: R$ {valor\_com\_desconto:.2f} \t (Desconto: {desconto}%)") |

|  |
| --- |
| **Saída do Console (FORMATO IMAGEM – PRINT DA TELA)** |
|  |

|  |
| --- |
| **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 pizzaria. Você ficou com a parte de desenvolver a interface do cliente para retirada do produto.**  **A Pizzaria possui seguinte tabela de sabores de pizzas listados com sua descrição, códigos e valores:**   |  |  |  |  | | --- | --- | --- | --- | | Código | Descrição | Pizza Média - M | Pizza Grande – G (30% mais cara) | | 21 | **Napolitana** | **R$ 20,00** | **R$ 26,00** | | 22 | **Margherita** | **R$ 20,00** | **R$ 26,00** | | 23 | **Calabresa** | **R$ 25,00** | **R$ 32,50** | | 24 | **Toscana** | **R$ 30,00** | **R$ 39,00** | | 25 | **Portuguesa** | **R$ 30,00** | **R$ 39,00** |   **Elabore um programa em Python que:**   1. **Entre com o tamanho da pizza** 2. **Entre com o código do produto 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 pizza e/ou NÚMERO diferente dos da tabela printar na tela: ‘opção inválida’ 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 duas pizzas** 9. **Colocar um exemplo de SAIDA DE CONSOLE com erro ao digitar código**   **Segue o exemplo de SAIDA DE CONSOLE:**    **Figura: Exemplo de programa com 2 pizzas pedidas, um erro no pedido do tamanho e um erro no código da pizza** |

|  |
| --- |
| **Apresentação do Código (FORMATO TEXTO)** |
| # EXERCICIO 2  n = "Gabriel Pinto Iório / RU: 4179744"  # => IDENTIFICADOR PESSOAL, NOME E RU  print(*f*"Bem vindo a pizzaria do {n}")  cardapio\_impressao = { #cardapio      21: ["Napolitana",  20, 26],      22: ["Margherita",  20, 26],      23: ["Calabresa",   25, 32.50],      24: ["Toscana",     30, 39],      25: ["Portuguesa",  30, 39]  }  *def* funCARDAPIO(*cardapio\_impressao*: *dict*): # Função que imprime o cardápio, para que o texto tenha boa formatação        cabecalho = *f*"| {'Código':^10s} | {'Descrição':^20s} | {'Pizza Média':^15s} | {'Pizza Grande':^15s} |" # para a formatação do cabeçalho do cardapio        print(*f*"{'Cardápio':-^{len(cabecalho)}s}")      print(cabecalho)      print("-"\*len(cabecalho))        for codigo, produtos in cardapio\_impressao.items():          print(*f*"| {codigo:^10} | {produtos[0]:^20} | {*f*'R$ {produtos[1]:.2f}':^15} | {*f*'R$ {produtos[2]:.2f}':^15} |")      print("-"\*len(cabecalho))  *def* selecionarTamanho(): # Função para seleção do tamanho da pizza      while True:          try:              tamanho = input("Qual tamanho de pizza que deseja (M/G): ").strip().upper()              if tamanho not in ["M", "G"]:                  print("\nTamanho inválido.\n")                  continue              else:                  break          except ValueError:              print("\nDigite um tamanho válido.\n")      return tamanho  *def* selecionarSabor(): # Função para seleção do sabor      while True:          try:              codigo = *int*(input("Entre com o código do sabor desejado: "))              if codigo not in cardapio\_impressao.keys():                  print("\nCódigo informado não existe.\n")                  continue              else:                  print(*f*"Você pediu uma pizza {cardapio\_impressao[codigo][0]}")                  break          except ValueError:              print("\nDigite um código válido.\n")      return codigo  total = 0  # Exibe o cardápio  funCARDAPIO(cardapio\_impressao.copy())  while True:      # Seleção de tamanho e sabor      tamanho = selecionarTamanho()      sabor = selecionarSabor()      # Soma o valor baseado no tamanho      if tamanho == "M":          total += cardapio\_impressao[sabor][1]      elif tamanho == "G":          total += cardapio\_impressao[sabor][2]      else:          total += 0      # Pergunta se deseja continuar      print("\nDeseja pedir mais alguma coisa?")      print("1 - Sim")      print("0 - Não")      # Valida a opção informada      while True:          try:              continuar = *int*(input("Escolha: "))                if continuar not in [0, 1]:                  print("\nOpção inválida.")                  continue              else:                  break          except ValueError:              print("\nOpção inválida.")      # Se a escolha for Zero - Imprime o total e encerra a execução      if continuar == 0:          print(*f*"\nO total a ser pago é: R$ {total:.2f}")          break |

|  |
| --- |
| **Saída do Console (FORMATO IMAGEM – PRINT DA TELA)** |
|  |

|  |
| --- |
| **QUESTÃO 3 de 4 (25 pontos)**  **CONTEÚDO ATÉ AULA 05** |
| **Enunciado: Imagina-se que você e sua equipe foram contratados por um restaurante que serve feijoada para desenvolver a solução de software. Você ficou encarregado da parte de retirar pedido por parte do cliente.**  **O valor que a empresa cobra por feijoada é dado pela seguinte equação:**  **Em que cada uma das variáveis que compõe o preço total é quantizada da seguinte maneira:**   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Quadro 1: Volume versus Valor**   |  |  | | --- | --- | | volume (ml) | valor (R$) | | volume < 300 | **Não é aceito** | | 300 <= volume <= 5000 | **volume \* 0.08** | | volume > 5000 | **Não é aceito** | | **Quadro 2: Opção versus multiplicador**   |  |  | | --- | --- | | peso(kg) | multiplicador | | b - Básica (Feijão + paiol + costelinha) | **1.00** | | p - Premium (Feijão + paiol + costelinha + partes de porco) | **1.25** | | s - Suprema (Feijão + paiol + costelinha + partes do porco + charque + calabresa + bacon) | **1.50** | | | **Quadro 3: Acompanhamento versus Valor**   |  |  | | --- | --- | | rota | Valor (R$) | | 0- Não desejo mais acompanhamentos (encerrar pedido) | **0,00** | | 1- 200g de arroz | **5,00** | | 2- 150g de farofa especial | **6,00** | | 3- 100g de couve cozida | **7,00** | | 4- 1 laranja descascada | **3,00** | | |   **Elabore um programa em Python que:**   1. **Pergunte o volume (em ml).Se digitar um valor não numérico e/ou volume for menor/maior que o limite aceito repetir a pergunta;** 2. **Pergunte a opção da feijoada. Se digitar uma opção não válida deve repetir a pergunta** 3. **Pergunte o acompanhamento. Deve-se perguntar se o usuário quer mais um acompanhamento até digitar a opção 0** 4. **Encerre o total a ser pago com base na equação desse enunciado;** 5. **Deve-se codificar uma função volumeFeijoada (EXIGÊNCIA 1 de 3);**    * **Deve-se perguntar dentro da função o volume da porção (em ml);**    * **Deve-se ter um if/else ou if/elif ou if/else/elif para verificar se o usuário não digitou um volume fora da faixa com que o restaurante 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 opcaoFeijoada (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 acompanhamentoFeijoada (EXIGÊNCIA 3 de 3);**    * **Deve-se perguntar dentro se deseja ou não mais algum acompanhamento**    * **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 um pedido com volume, opção e 2 acompanhamentos válidos** 9. **Colocar um exemplo de SAIDA DE CONSOLE com o tratamento de erro quando digitado um valor não numérico é digitado e uma opção não permitida no menu opção de feijoada**   **Segue o exemplo de SAIDA DE CONSOLE:**    **Figura: Exemplo de programa com 1 pedido de feijoada suprema + 2 acompanhamentos, um erro no pedir um volume menor que o exigido e um erro de digitação.** |

|  |
| --- |
| **Apresentação do Código (FORMATO TEXTO)** |
| # EXERCICIO 3  n = "Gabriel Pinto Iório / RU: 4179744"  # => IDENTIFICADOR PESSOAL, NOME E RU  print(*f*"Seja bem vindo a feijoada do {n}")  opcoes = {      "b": ["Básica (Feijão + Paiol + Costelinha)", 1],      "p": ["Premium (Feijão + Paiol + Costelinha + Partes de porco)", 1.25],      "s": ["Suprema (Feijão + Paiol + Costelinha + Partes do porco + Charque + Calabresa + Bacon)", 1.50]  }  acompanhamentos = {      0: ["Não quero mais acompanhamentos (encerrar pedido)", 0],      1: ["200g de arroz", 5],      2: ["150g de farofa especial", 6],      3: ["100g de couve cozida", 7],      4: ["1 laranja descascada", 3]  }  *def* volume(): # Função que solicita e valida o volume      valor = 0      while True:          try:              volume = *float*(input("Entre com a quantidade que deseja (ml): ").strip().upper())                if volume < 300 or volume > 5000:                  print("\nNão aceitamos porções menores que 300ml ou maiores que 5L. Faça uma nova tentativa de escolha!\n")              else:                  valor = volume \* 0.08                  break          except ValueError:              print("\nDigite um volume válido.\n")      return valor  *def* opcaoFeijoada(): # Função que solicita e valida a opção        multiplicador = None      while True:          try:              print("\nEntre com a opção de Feijoada:")              for key, values in opcoes.items():                  print(*f*"{key} - {values[0]}")              opcao = input().strip().lower()                if opcao not in opcoes.keys():                  print("\nOpção inválida\n")              else:                  multiplicador = opcoes[opcao][1]                  break          except ValueError:              print("\nDigite um opcao válida.\n")      return multiplicador  *def* acompanhamentoFeijoada(): # Função que solicita e valida os acompanhamentos      total\_acompanhamento = 0      while True:          try:              print("\nDeseja mais algum acompanhamento:")              for key, values in acompanhamentos.items():                  print(*f*"{key} - {values[0]}")              opcao = *int*(input())                if opcao not in acompanhamentos.keys():                  print("\nOpção inválida\n")              elif opcao == 0:                  break              else:                  total\_acompanhamento += acompanhamentos[opcao][1]          except ValueError:              print("\nDigite um acompanhamento válido.\n")      return total\_acompanhamento  valor = volume()  opcao = opcaoFeijoada()  adicionais = acompanhamentoFeijoada()  total = (valor \* opcao) + adicionais  print(*f*"\nO valor total a ser pago pelo seu pedido é de (R$): {total:.2f} (Volume = {valor:.2f} \* opcao = {opcao:.2f} + Acompanhamento = {adicionais:.2f})") |

|  |
| --- |
| **Saída do Console (FORMATO IMAGEM – PRINT DA TELA)** |
|  |

|  |
| --- |
| **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 mercearia. Este software deve ter o seguinte menu e opções:**   1. **Cadastrar Produto** 2. **Consultar Produto(s)**    1. **Consultar Todas as Produto**    2. **Consultar Produto por Código**    3. **Consultar Produto(s) por Fabricante**    4. **Retornar** 3. **Remover Produto** 4. **Sair**   **Elabore um programa em Python que:**   1. **Deve-se codificar uma função cadastrarProduto (código) (EXIGÊNCIA 1);**    * **Essa função recebe como parâmetro um código exclusivo para cada produto cadastrado (DICA: utilize um contador como parâmetro)**    * **Dentro da função perguntar o nome do produto;**    * **Dentro da função perguntar o fabricante do produto;**    * **Dentro da função perguntar o valor do produto**    * **Cada produto 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 consultarProduto(EXIGÊNCIA 2);**    * **Dentro da função ter um menu com as seguintes opções:**      + **Consultar Todos os Produtos**      + **Consultar Produtos por Código**      + **Consultar Produtos por Fabricante**      + **Retornar** 3. **Deve-se codificar uma função chamada removerProduto (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) produto. Sendo que 2 delas do mesmo fabricante – ver figura 1** 5. **Colocar um exemplo de SAIDA DO CONSOLE com a consulta a todos os produtos cadastrados – 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 produto cadastrado e mostrando depois todos os produtos – ver figura 5**   **Segue os exemplos de SAIDA DE CONSOLE:**    **Figura 1: Exemplo de Cadastro de três produtos. Perceba que 2 deles tem o mesmo fabricante.**    **Figura 2: Exemplo de Consulta Todas os Produtos.**    **Figura 3: Exemplo de Consultar Produto por Código.**    **Figura 4: Exemplo de Consultar Produtos por Fabricante**    **Figura 5: Exemplo de Remover Produto 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)** |
| # EXERCICIO 4  n = "Gabriel Pinto Iório / RU: 4179744"  # => IDENTIFICADOR PESSOAL, NOME E RU  print(*f*"Seja bem vindo ao Controle de Estoque da loja do {n}")  menu\_da\_mercearia = {      1: 'Cadastrar Produto',      2: ['Consultar Produto', {          1: 'Consultar Todos os Produtos',          2: 'Consultar Produto por Código',          3: 'Consultar Produto(s) por Fabricante',          4: 'Retornar',      }],      3: 'Remover Produto',      4: 'Sair'  }  *def* FUN\_menu(*menu*: *dict*) -> None: # Função que imprime as opções dos menus        for key, val in menu.items():          print(*f*"{key} - {val if *type*(val) == *str* else val[0]}")        print('-' \* 30)  *def* mostrar\_menu(*menu*: *dict*) -> *int*: # Função que imprime e valida a opção do menu        print(*f*"{'Menu Principal':-^30s}")      FUN\_menu(menu)      while True:          try:              opcao = *int*(input("Escolha a opção desejada: "))              # Verifica se a opção está presente no dicionário do menu              # Se sim, limpa a tela e retorna a opção selecionada              if opcao in menu.keys():                  break              else:                  print("\nOpção inválida!\n")          except ValueError:              print("\nOpção inválida.\n")        return opcao  *def* cadastrarProduto(*codigo*: *int*) -> None: # Função que cadastra produtos        print("\nOpção Cadastrar Produto\n")        print("Código da Produto {:>03}".format(codigo))      nome = input("Digite o nome do produto: ").strip()      fabricante = input("Digite o fabricante do produto: ").strip()        while True:          try:              valor = *float*(input("Digite o valor (R$) do produto: "))              # Verifica se o valor é maior que zero              if valor <= 0:                  print("\nALERTA!!!! Digite um valor maior que zero.\n")              else:                  break          except ValueError:              print("\nALERTA!!! Digite um número válido.\n")      produtos[codigo] = []      produtos[codigo].append(nome)      produtos[codigo].append(fabricante)      produtos[codigo].append(valor)  *def* removerProduto() -> None: # Função que remove produtos        while True:          try:              codigo = *int*(input("\nEscreva o código do produto a ser removido: "))              # Verifica se existe o produto com o código informado              # Se sim, remove o produto do dicionário              if codigo not in produtos.keys():                  print("\nNenhum produto possui esse código.\n")              else:                  produtos.pop(codigo)              break          except ValueError:              print("\nALERTA!!!! Digite um código válido.\n")  *def* consultarProduto() -> None: # Função para consulta de produtos        # Variável usada para controlar a exibição do menu de consulta      exibir\_menu = True      cabecalho = *f*"| {'Código':<10} | {'Nome':<25} | {'Fabricante':<20} | {'Valor (R$)':>10} |"      len\_cabecalho = len(cabecalho)      while True:            try:              # Exibe o menu de consulta              if exibir\_menu:                  print("\nVocê selecionou a opção Consultar Produtos\n")                    print(*f*"{'Consultar Produtos':-^30s}")                  FUN\_menu(menu\_da\_mercearia[2][1])              opcao = *int*(input("Escolha a opção desejada: "))              # Opção 1 - Consultar Todas os produtos              if opcao == 1:                    # Lista todas os produtos cadastrados, se houver                  if len(produtos) > 0:                      print("\nTodos os produtos cadastrados:\n")                        print("-" \* len\_cabecalho)                      print(cabecalho)                      print("-" \* len\_cabecalho)                        for key, val in produtos.items():                          print(*f*"| {key:<10} | {val[0]:<25} | {val[1]:<20} | {val[2]:>10.2f} |")                        print("-" \* len\_cabecalho)                  else:                      print("\nNão existem produtos cadastradas.")                  print("")                # Opção 2 - Consultar produto por Código              elif opcao == 2:                  while True:                      try:                          codigo = *int*(input("\nDigite o código do produto a ser consultado: "))                          # Exibe o produto com o respectivo código, se houver                          if codigo in produtos.keys():                              print("-" \* len\_cabecalho)                              print(cabecalho)                              print("-" \* len\_cabecalho)                              print(*f*"| {codigo:<10} | {produtos[codigo][0]:<25} | {produtos[codigo][1]:<20} | {produtos[codigo][2]:>10.2f} |")                                print("-" \* len\_cabecalho)                          else:                              print("\nNão existe produto com esse código.\n")                          print("")                          break                      except ValueError:                          print("\nDigite um código válido.\n")              # Opção 3 - Consultar produto por Fabricante              elif opcao == 3:                  while True:                      try:                          fabricante = input("\nDigite o fabricante da produto a ser consultada: ")                          # Variável usada para controlar a exibição do cabeçalho da lista                          existe\_produto = False                          for key, val in produtos.items():                              # Imprime os dados do produto do fabricante informado                              if val[1].upper() == fabricante.upper():                                  if not existe\_produto:                                      print("-" \* len\_cabecalho)                                      print(cabecalho)                                      print("-" \* len\_cabecalho)                                    print(*f*"| {key:<10} | {val[0]:<25} | {val[1]:<20} | {val[2]:>10.2f} |")                                    existe\_produto = True                          if not existe\_produto:                              print("\nNão existe produto com esse fabricante.\n")                          else:                              print("-" \* len\_cabecalho)                            break                      except ValueError:                          print("Digite um fabricante válido.")              elif opcao == 4:                  break              else:                  print("\nAtenção! Você digitou uma opção inválida.\n")              exibir\_menu = True          except ValueError:              print("\nAtenção! Você digitou uma opção inválida.\n")              exibir\_menu = False  produtos = {}  while True:      opcao\_selecionada = mostrar\_menu(menu\_da\_mercearia)      if opcao\_selecionada == 1:          cod\_produtos = len(produtos) + 1          cadastrarProduto(cod\_produtos)      elif opcao\_selecionada == 2:          consultarProduto()      elif opcao\_selecionada == 3:          removerProduto()      else:          break |

|  |
| --- |
| **Saída do Console (FORMATO IMAGEM – PRINT DA TELA)** |
|  |