QUESTÃO 1 de 4 - 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 aceita cartões de crédito. Uma das estratégias de vendas dessa empresa X é cobrar um Juros maior conforme a quantidade de parcelas que o cliente desejar, conforme a **listagem abaixo**:

* Se a quantidade de parcelas for **menor** que **4**, o Juros será de **0%** (0 / 100);
* Se a quantidade de parcelas for **igual ou maior** que **4** e **menor que** **6**, o Juros será de **4%** (4 / 100);
* Se a quantidade de parcelas for **igual ou maior** que **6** e **menor que** **9**, o Juros será de **8%** (8 / 100);
* Se a quantidade de parcelas for **igual ou maior** que **9** e **menor que** **13**, o Juros será de **16%** (16 / 100);
* Se a quantidade de parcelas for **igual ou maior** que **13**, o Juros será de **32%** (32 / 100);

O valor da parcela é calculado da seguinte maneira:

O valor total parcelado é calculado da seguinte maneira:

Elabore um programa em Python que:

1. Deve-se implementar o **print** com o seu **nome completo** (somente print, não usar input aqui).

Por exemplo: **print(“Bem-vindos a loja do Bruno Kostiuk”)** [EXIGÊNCIA DE CÓDIGO 1 de 6];

1. Deve-se implementar o input do **valorDoPedido** e da **quantidadeParcelas** [EXIGÊNCIA DE CÓDIGO 2 de 6];
2. Deve-se implementar o Juros **conforme a enunciado acima** (obs.: atente-se as condições de menor, igual e maior) [EXIGÊNCIA DE CÓDIGO 3 de 6];
3. Deve-se implementar o **valorDaParcela** e **valorTotalParcelado** [EXIGÊNCIA DE CÓDIGO 4 de 6];
4. Deve-se implementar as estruturas **if, elif e else (todas elas)** [EXIGÊNCIA DE CÓDIGO 5 de 6];
5. Deve-se inserir comentários **relevantes** no código [EXIGÊNCIA DE CÓDIGO 6 de 6];
6. Deve-se apresentar na saída de console uma mensagem com seu nome completo [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 2];
7. Deve-se apresentar na saída de console um parcelamento com Juros (**quantidadeParcelas** maior ou igual a 4) apresentando o valor da Parcela e o valor Total Parcelado [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 2];

EXEMPLO DE SAÍDA DE CONSOLE:

A imagem apresenta a seguinte saída de código: 
Bem-vindo a Loja do Bruno Kostiuk
Entre com o valor do pedido: 1000.00
Entre com a quantidade de parcelas: 14
O valor das parcelas é de:R$ 94.29
O valor Total Parcelado é de:R$ 1320.00

**Figura 1.1: Exemplo de saída de console que o aluno deve fazer. Em que se perguntar o valor do pedido (pode ser qualquer valor) a quantidade de parcelas (maior ou igual a 4** [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 2]**), e é apresentado o valorDaParcela e o valorTotalParcelado.**

Apresentação de **Código da Questão 1**:

# PROGRAMA PRINCIPAL: CALCULANDO PARCELAS

print("Bem vindo a loja do Caleb James Hall!")

# Usando loops para garantir que os dados recebidos sao validos

while True:

    try:

        valorDoPedido = float(input("Qual é o valor do seu pedido? "))  # deve ser um NUMERO

        if (valorDoPedido > 0):  # NUMERO deve ser maior que zero

            break  # se for NUMERO maior que zero, break.

        else:

            print("Tem que ser um valor maior que 0")

    except ValueError:

        print("Por favor, insira um número válido. (acima de 0)")

while True:

    try:

        qtdDeParcelas = int(input("Em quantas vezes quer parcelar? "))  # deve ser um NUMERO

        if (qtdDeParcelas > 0):  # NUMERO deve ser maior que zero

            break  # se for NUMERO maior que zero, break.

        else:

            print("Por favor, insira um número válido. (acima de 0)")

    except ValueError:

        print("Por favor, insira um número válido. (acima de 0)")

# Calculando os juros baseado no numero de parcelas

juros = 0

if (qtdDeParcelas < 4):

    juros = 1

elif (qtdDeParcelas >= 4 and qtdDeParcelas < 6):

    juros = 1.04

elif (qtdDeParcelas >= 6 and qtdDeParcelas < 9):

    juros = 1.08

elif (qtdDeParcelas >= 9 and qtdDeParcelas < 13):

    juros = 1.16

else:

    juros = 1.32

# Calculando o valor de cada parcela e o total a pagar

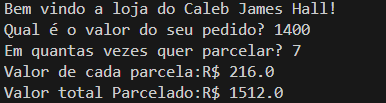
valorTotalParcelado = valorDoPedido \* juros # total a pagar

valorDaParcela = valorTotalParcelado / qtdDeParcelas # total a pagar POR PARCELA

print(f"Valor de cada parcela:R$ {round(valorDaParcela, 2)}")

print(f"Valor total Parcelado:R$ {round(valorTotalParcelado, 2)}")

Apresentação de **Saída do Console da Questão 1**:



QUESTÃO 2 de 4 - Conteúdo até aula 04

Enunciado: Você e sua equipe de programadores foram contratados para desenvolver um app de vendas para uma loja que vende Marmitas de Bife Acebolado ou Filé de Frango. Você ficou com a parte de desenvolver a interface do cliente para retirada do produto.

A Loja possui seguinte relação:

* Tamanho **P** de Bife Acebolado (**BA**) custa 16 reais e o Filé de Frango (**FF**) custa 15 reais;
* Tamanho **M** de Bife Acebolado (**BA**) custa 18 reais e o Filé de Frango (**FF**) custa 17 reais;
* Tamanho **G** de Bife Acebolado (**BA**) custa 22 reais e o Filé de Frango (**FF**) custa 21 reais;

Elabore um programa em Python que:

1. Deve-se implementar o **print** com o seu **nome completo** (somente print, não usar input aqui).

Por exemplo: **print(“Bem vindos a loja de Marmitas do Bruno Kostiuk”)**

Além do seu nome completo, deve-se implementar um **print com um Menu** para o cliente. [EXIGÊNCIA DE CÓDIGO 1 de 8];

1. Deve-se implementar o input do **sabor** (BA/FF) e o print “Sabor inválido. Tente novamente" se o usuário entra com valor diferente de BA e FF [EXIGÊNCIA DE CÓDIGO 2 de 8];
2. Deve-se implementar o input do **tamanho** (P/M/G) e o print “Tamanho inválido. Tente novamente" se o usuário com entra valor diferente de P, M ou G [EXIGÊNCIA DE CÓDIGO 3 de 8];
3. Deve-se implementar **if, elif e/ou else**, utilizando o modelo **aninhado** (aula 3 – Tema 4) com cada uma das combinações de **sabor** e **tamanho** [EXIGÊNCIA DE CÓDIGO 4 de 8];
4. Deve-se implementar um **acumulador** para somar os valores dos pedidos [EXIGÊNCIA DE CÓDIGO 5 de 8];
5. Deve-se implementar o input com a pergunta: “Deseja pedir mais alguma coisa?”. Se sim **repetir a partir do item B**, senão encerrar o programa executar o print do **acumulador** [EXIGÊNCIA DE CÓDIGO 6 de 8];
6. Deve-se implementar as estruturas de **while, break, continue (todas elas)** [EXIGÊNCIA DE CÓDIGO 7 de 8];
7. Deve-se inserir comentários **relevantes** no código [EXIGÊNCIA DE CÓDIGO 8 de 8];
8. Deve-se apresentar na saída de console uma mensagem com o seu nome completo e o menu para o cliente conhecer as opções [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 4];
9. Deve-se apresentar na saída de console um pedido em que o usuário errou o **sabor** [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 4];
10. Deve-se apresentar na saída de console um pedido em que o usuário errou o **tamanho** [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 4];
11. Deve-se apresentar na saída de console um pedido com duas opções sabores diferentes e com tamanhos diferentes [EXIGÊNCIA DE SAÍDA DE CONSOLE 4 de 4];

EXEMPLO DE SAÍDA DE CONSOLE:

A imagem apresenta a seguinte saída de código:

------ Bem-vindo a Loja de Marmitas do Bruno Kostiuk -----------
------------------------------Cardápio--------------------------
----------------------------------------------------------------
---| Tamanho  |  Bife Acebolado(BA)  |  Filé de Frango(FF)  |---
---|    P     |       R$ 16.00       |       R$ 15.00       |---
---|    M     |       R$ 18.00       |       R$ 17.00       |---
---|    G     |       R$ 22.00       |       R$ 21.00       |---
----------------------------------------------------------------
Entre com o sabor desejado (BA/FF): BF
Sabor inválido. Tente novamente

Entre com o sabor desejado (BA/FF): BA
Entre com o tamanho desejado (P/M/G): EXGG
Tamanho inválido. Tente novamente

Entre com o sabor desejado (BA/FF): BA
Entre com o tamanho desejado (P/M/G): M
Você pediu um Bife Acebolado no tamanho M: R$ 18.00

Deseja mais alguma coisa? (S/N): S
Entre com o sabor desejado (BA/FF): FF
Entre com o tamanho desejado (P/M/G): G
Você pediu um Filé de Frango no tamanho G: R$ 21.00

Deseja mais alguma coisa? (S/N): N

O valor total a ser pago: R$ 39.00

**Figura 2.1: Exemplo de saída de console que o aluno deve fazer. Em que se perguntar o sabor e o tamanho. Há uma tentativa de pedido que se errou o sabor e outra que se errou o tamanho. Há também um pedido com dois itens com sabores e tamanhos diferentes.**

Apresentação de **Código da Questão 2**:

# VARIAVEIS GLOBAIS

totalValor = 0  # VALOR TOTAL DAS COPMRAS QUE SERA O PRINT() NO FINAL

# FAZENDO O MENU/CARDAPIO

print("------------ Bem vindos a loja do Caleb James Hall. ------------")

print("-"\*27, "Cardapio", "-"\*27)

print("-"\*64)

print('---|  Tamanho  | Bife Acebolado(BA)  |  File de Frango(FF)  |---')

print('---|     P     |      R$ 16.00       |       R$ 15.00       |---')

print('---|     M     |      R$ 18.00       |       R$ 17.00       |---')

print('---|     G     |      R$ 22.00       |       R$ 21.00       |---')

print("-"\*64)

# COMPRANDO AS CARNES

while True:

    while True:  # VERIFICAR QUE E SABOR E TAMANHO E VALIDA

        # VERIFICAR SABOR VALIDO

        sabor = input("Sabor desejado: (BA/FF) ")

        if sabor.upper() == "BA" or sabor.upper() == "FF":

            sabor = sabor.upper()

        else:

            print("Sabor invalido, seleciona (BA/FF)\n")

            continue

        # VERIFICAR TAMNAHO VALIDO

        tamanho = input("Tamanho desejado: (P/M/G) ")

        if tamanho.upper() == "P" or tamanho.upper() == "M" or tamanho.upper() == "G":

            tamanho = tamanho.upper()

            break

        else:

            print("Tamanho invalido. Seleciona (P/M/G)\n")

            continue

    if sabor.upper() == 'BA':  # PRECO PARA BIFE ACEBOLADO BASEADO NO TAMANHO

        sabor = "Bife Acebolado"

        if tamanho == "P":

            totalValor += 16.00

            valorUnitario = 16.00

        elif tamanho == "M":

            totalValor += 18.00

            valorUnitario = 18.00

        else:

            totalValor += 22.00

            valorUnitario = 22.00

    else:  # PRECO PARA FILE DE FRANGO BASEADO NO TAMANHO

        sabor = "File de Frango"

        if tamanho.upper() == "P":

            totalValor += 15.00

            valorUnitario = 15.00

        elif tamanho.upper() == "M":

            totalValor += 17.00

            valorUnitario = 17.00

        else:

            totalValor += 21.00

            valorUnitario = 21.00

    print(f"Voce comprou {sabor} do tamanho {tamanho.upper()}: {valorUnitario}")

    # VER SE QUISER COMPRAR MAIS COISAS

    continuarComprando = input("\nDeseja comprar mais algo? (S/N) \n")

    if continuarComprando.upper() == "S":

        continue

    else:

        break

# O VALOR DEPOIS DAS COPMRAS

print(f"O total foi R${totalValor}")

Apresentação de **Saída do Console da Questão 2**:

A screenshot of a computer

Description automatically generated

QUESTÃO 3 de 4 - Conteúdo até aula 05

Enunciado: Você foi contratado para desenvolver um sistema de cobrança de serviços de uma fábrica que vende Camisetas em atacado. Você ficou com a parte de desenvolver a interface com o funcionário.

A Fábrica opera as vendas da seguinte maneira:

* Camiseta Manga Curta Simples (MCS), o valor unitário é de um real e oitenta centavos;
* Camiseta Manga Longa Simples (MLS), o valor unitário é de dois reais e dez centavos;
* Camiseta Manga Curta Com Estampa (MCE), o valor unitário é de dois reais e noventa centavos;
* Camiseta Manga Longa Com Estampa (MLE), o valor unitário é de três reais e vinte centavos;
* Se número de camisetas for **menor** que 20 não há desconto na venda;
* Se número de camisetas for **igual ou maior** que 20 e **menor** que 200, o desconto será de 5%;
* Se número de camisetas for **igual ou maior** que 200 e **menor** que 2000, o desconto será de 7%;
* Se número de camisetas for **igual ou maior** que 2000 e **menor ou igual** que 20000, o desconto será de 12%;
* Se número de camisetas for **maior** que 20000, não é aceito pedidos nessa quantidade de camisetas;

* Para o **adicional** de frete por transportadora (1) é cobrado um valor **extra** de 100 reais;
* Para o **adicional** de frete por Sedex (2) é cobrado um valor **extra** de 200 reais;
* Para o **adicional** de retirar o pedido na fábrica (0) é cobrado um valor **extra** de 0 reais;

O valor final da conta é calculado da seguinte maneira:

total = (**modelo \* num\_camisetas) + frete**

Elabore um programa em Python que:

1. Deve-se implementar o **print** com o seu **nome completo** (somente print, não usar input aqui).

Por exemplo: **print(“Bem vindos a Fábrica de Camisetas do Bruno Kostiuk”)** [EXIGÊNCIA DE CÓDIGO 1 de 7];

1. Deve-se implementar a função **escolha\_modelo()** em que**:** [EXIGÊNCIA DE CÓDIGO 2 de 7];
   1. Pergunta o **modelo** desejado;
   2. **Retorna** **o valor** **do modelo** com base na escolha do usuário (use **return**);
   3. Repete a pergunta do item **B.a** se digitar uma opção diferente de: MCS/MLS/MCE/MLE;
2. Deve-se implementar a função **num\_camisetas()** em que**:** [EXIGÊNCIA DE CÓDIGO 3 de 7];
   1. Pergunta o **número de camisetas**;
   2. **Retorna** (use **return**) o **número de camisetas** com desconto seguindo a regra do enunciado (desconto calculado em cima do número de camisetas);
   3. Repete a pergunta do item **C.a** se digitar um valor acima de 20000 ou valor não numérico (use try/except para não numérico)
3. Deve-se implementar a função **frete()** em que**:** [EXIGÊNCIA DE CÓDIGO 4 de 7];
   1. Pergunta pelo serviço **adicional de frete**;
   2. **Retorna** (use **return**) o **valor** de apenas uma das **opções** de **frete**
   3. Repetir a pergunta item **D.a** se digitar uma opção diferente de: 1/2/0;
4. Deve-se implementar o total a pagar no código principal (**main**), ou seja, não pode estar dentro de função, conforme o enunciado [EXIGÊNCIA DE CÓDIGO 5 de 7];
5. Deve-se implementar **try/except** [EXIGÊNCIA DE CÓDIGO 6 de 7];
6. Deve-se inserir comentários relevantes no código [EXIGÊNCIA DE CÓDIGO 7 de 7];
7. Deve-se apresentar na saída de console uma mensagem com o seu nome completo [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 4];
8. Deve-se apresentar na saída de console um pedido no qual o usuário errou a opção de modelo [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 4];
9. Deve-se apresentar na saída de console um pedido no qual o usuário digitou ultrapassou no número de camisetas [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 4];
10. Deve-se apresentar na saída de console um pedido com opção de modelo, número de camisetas e frete válidos [EXIGÊNCIA DE SAÍDA DE CONSOLE 4 de 4];

EXEMPLO DE SAÍDA DE CONSOLE:

A imagem apresenta uma saída de console com o seguinte texto:

Bem vindo a Fábrica de Camisetas do do Bruno Kostiuk

Entre com o modelo desejado
MCS - Manga Curta Simples
MLS - Manga Longa Simples
MCE - Manga Curta Com Estampa
MLE - Manga Longa Com Estampa
>>MLL
Escolha inválida, entre com o modelo novamente


Entre com o modelo desejado
MCS - Manga Curta Simples
MLS - Manga Longa Simples
MCE - Manga Curta Com Estampa
MLE - Manga Longa Com Estampa
>>MCS
Entre com o número de camisetas: 300000
Não aceitamos tantas camisetas de uma vez.
Por favor, entre com o número de camisetas novamente.

Entre com o número de camisetas: 10000

Escolha o tipo de frete:
1 - Frete por transportadora - R$ 100.00
2 - Frete por Sedex - R$ 200.00
0 - Retirar pedido na fábrica - R$ 0.00
>>2
Total: R$ 16040.00 (Modelo: 1.80 * Quantidade(com desconto): 8800 + frete: 200.00)

**Figura 3.1: Exemplo de saída de console que o aluno deve fazer. Em que se pergunta pelo modelo e se erra opção inicialmente, que se passa do número de camisetas acima do aceito. Na sequência, o usuário digitou um modelo, número de camisetas e frete válidos.**

Apresentação de Código da Questão 3:

# FUNCOES

# ======================== PRIMEIRA FUNCAO - MODELO =========================

def escolha\_modelo():

    print("Qual eh o modelo desejado?")

    print("MCS - Manga Curta Simples: R$ 1.80")

    print("MLS - Manga Longa Simples: R$ 2.10")

    print("MCE - Manga Curta Com Estampa: R$ 2.90")

    print("MLE - Manga Longa Com Estampa: R$ 3.20")

    # loop para confirmar que seja um modelo valido

    while True:

        modelo = input(">> ")

        # confirmar que eh um modelo valido

        if modelo.upper() in ("MCS", "MLS", "MCE", "MLE"):

            # Definindo o valor do modelo

            if modelo.upper() == "MCS":

                valor\_modelo = 1.80

            elif modelo.upper() == "MLS":

                valor\_modelo = 2.10

            elif modelo.upper() == "MCE":

                valor\_modelo = 2.90

            elif modelo.upper() == "MLE":

                valor\_modelo = 3.20

            break

        else:

            print("Seleciona um modelo valido.\n")

    # retornar o valor do modelo

    return valor\_modelo

# =================== SEGUNDA FUNCAO - NUMERO DE CAMISAS ====================

def num\_camiseta():

    # Loop para confirmar que seja um numero valido

    while True:

        # Try o numero de camisetas, deve ser entre 0 e 20,000

        try:

            camisetas = int(input("Entre com o numero de camisetas (0-20,000): "))

            # if para confirmar o valor esta certo

            if camisetas > 0 and camisetas <= 20000:

                # if para calcular desconto baseado no numero de camisas compradas

                if camisetas < 20:

                    camisetas \*= 1

                elif camisetas >= 20 and camisetas < 200:

                    camisetas \*= 0.95

                elif camisetas >= 200 and camisetas < 2000:

                    camisetas \*= 0.93

                elif camisetas >= 2000 and camisetas <= 20000:

                    camisetas \*= 0.88

                break

            else:

                print("Entrar com numero valido.\n")

        # pegar o error caso for error valor

        except ValueError:

            print("Precisa Entrar com numero. \n ")

    # retornar o numero de camisas COM DESCONTO

    return camisetas

# ======================== TERCEIRA FUNCAO - FRETE ========================

def escolha\_frete():

    print("Escolha o tipo de frete:")

    print("0 - Retirar na fabrica - R$ 0.00")

    print("1 - Frete por transportadora - R$ 100.00")

    print("2 - Frete por Sedex - R$ 200.00")

    # Loop para confirmar que seja um frete valido

    while True:

        # Try o valor do input, deve ser entre 0 e 2

        try:

            frete = int(input(">> "))

            # calcular o valor do frete beseado no input 'frete'

            if frete in (0, 1, 2):

                if frete == 0:

                    valor\_frete = 0.00

                elif frete == 1:

                    valor\_frete = 100.00

                else:

                    valor\_frete = 200.00

                break

            else:

                print("Escolha um tipo valido.\n")

        # pegar o error case for error valor

        except ValueError:

            print("Escolha um tipo valido.\n")

    return valor\_frete

#=========================== PROGRAMA PRINCIPAL =============================

print("Bem vindos a Fabrica de Camisas do Caleb Hall!")

# Chamando as funcoes e colocando os valores retornados num variavel

valor\_modelo = escolha\_modelo()

camisetas = num\_camiseta()

frete = escolha\_frete()

# calcular total

total = (valor\_modelo \* camisetas) + frete

# fazer print do total e mostrar o calculo de todos os valores

print(f"Valor total: {total} (Modelo: {valor\_modelo} \* Quantidade(com desconto): {camisetas}) + frete: {frete})")

Apresentação de Saída do Console da Questão 3:

SUBSTITUIR ESSE TEXTO QUE ESTÁ EM LARANJA PELA A SAÍDA DO CONSOLE DO EXERCÍCIO 3

NÃO ESQUECER DE CUMPRIR AS EXIGÊNCIAS

**SERÁ ACEITO SOMETE SAÍDAS DO CONSOLE NO FORMATO IMAGEM (NADA DE TEXTO AQUI! ZERA ESSA PARTE DA QUESTÃO!)**

QUESTÃO 4 de 4 - Conteúdo até aula 06

Enunciado: Você e sua equipe de programadores foram contratados por pequena empresa para desenvolver o software de gerenciamento de funcionários. Este software deve ter o seguinte menu e opções:

1. Cadastrar Funcionário
2. Consultar Funcionário
   1. Consultar Todos
   2. Consultar por Id
   3. Consultar por setor
   4. Retornar ao menu
3. Remover Funcionário
4. Encerrar Programa

Elabore um programa em Python que:

1. Deve-se implementar o **print** com o seu **nome completo** (somente print, não usar input aqui).

Por exemplo: **print(“Bem vindos a empresa do Bruno Kostiuk”)** [EXIGÊNCIA DE CÓDIGO 1 de 8];

1. Deve-se implementar uma lista com o nome de **lista\_funcionarios** e a variável **id\_global** com valor inicial igual ao número de seu RU [EXIGÊNCIA DE CÓDIGO 2 de 8];
2. Deve-se implementar uma função chamada **cadastrar\_funcionario(id)** em que**:** [EXIGÊNCIA DE CÓDIGO 3 de 8];
   1. Pergunta **nome**, **setor**, **salario** do funcionário;
   2. Armazena o **id** (este é fornecido via parâmetro da função), **nome**, **setor**, **salario** dentro de um dicionário;
   3. **Copiar** o dicionário para dentro da **lista\_funcionarios** (utilizar o **copy**);
3. Deve-se implementar uma função chamada **consultar\_funcionarios()** em que**:** [EXIGÊNCIA DE CÓDIGO 4 de 8];
   1. Deve-se perguntar qual opção deseja (1. Consultar Todos / 2. Consultar por Id / 3. Consultar por Setor / 4. Retornar ao menu):
      1. Se Consultar Todos, apresentar todos os funcionários com todos os seus dados cadastrados;
      2. Se Consultar por Id, solicitar ao usuário que informe um id, e apresentar o funcionário específico com todos os seus dados cadastrados;
      3. Se Consultar por Setor, solicitar ao usuário que informe o setor, e apresentar o(s) funcionário(s) do setor com todos os seus dados cadastrados;
      4. Se Retornar ao menu, deve-se **retornar** ao menu principal (return);
      5. Se Entrar com um valor diferente de 1, 2, 3 ou 4, printar “Opção inválida" e repetir a pergunta **D.a**.
      6. Enquanto o usuário não escolher a opção 4, o menu consultar funcionários deve se repetir.
4. Deve-se implementar uma função chamada **remover\_funcionario()** em que**:** [EXIGÊNCIA DE CÓDIGO 5 de 8];
   1. Deve-se pergunta pelo **id** do funcionário a ser removido;
   2. Remover o funcionário da **lista\_funcionarios**;
   3. Se o id fornecido não for de um funcionário da lista, printar “Id inválido” e repetir a pergunta **E.a**.
5. Deve-se implementar uma estrutura de menu no código principal (**main**), ou seja, **não pode estar dentro de função**, em que: [EXIGÊNCIA DE CÓDIGO 6 de 8];
   1. Deve-se pergunta qual opção deseja (1. Cadastrar Funcionário / 2. Consultar Funcionário / 3. Remover Funcionário / 4. Encerrar Programa):
      1. Se Cadastrar Funcionário, **incrementar** em um **id\_ global** e chamar a função **cadastrar\_funcionario(id\_ global)**;
      2. Se Consultar Funcionário, chamar função **consultar\_funcionario ()**;
      3. Se Remover Funcionário, chamar função **remover\_funcionario()**;
      4. Se Encerrar Programa, sair do menu (e com isso acabar a execução do código);
      5. Se Entrar com um valor diferente de 1, 2, 3 ou 4, printar “Opção inválida" e repetir a pergunta **F.a**.
      6. Enquanto o usuário não escolher a opção 4, o menu deve se repetir.
6. Deve-se implementar uma **lista de dicionários** (uma lista contento dicionários dentro)[EXIGÊNCIA DE CÓDIGO 7 de 8];
7. Deve-se inserir comentários **relevantes** no código [EXIGÊNCIA DE CÓDIGO 8 de 8];
8. Deve-se apresentar na saída de console uma mensagem com o seu nome completo [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 6];
9. Deve-se apresentar na saída de console um cadastro de 3 funcionários (sendo **2** deles no mesmo setor) [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 6];
10. Deve-se apresentar na saída de console uma consulta de todos os funcionários [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 6];
11. Deve-se apresentar na saída de console uma consulta por código (id) de um dos funcionários [EXIGÊNCIA DE SAÍDA DE CONSOLE 4 de 6];
12. Deve-se apresentar na saída de console uma consulta por setor em que **2** funcionários sejam do mesmo setor [EXIGÊNCIA DE SAÍDA DE CONSOLE 5 de 6];
13. Deve-se apresentar na saída de console uma remoção de um dos funcionários seguida de uma consulta de todos os funcionários [EXIGÊNCIA DE SAÍDA DE CONSOLE 6 de 6];

EXEMPLO DE SAÍDA DE CONSOLE:

A imagem apresenta o seguinte texto de saída de console: 

Bem vindo a Empresa do Bruno Kostiuk
--------------------------------------------------
--------------- MENU PRINCIPAL -------------------
Escolha a opção desejada:
1 - Cadastrar Funcionários
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - Sair
>>1
--------------------------------------------------
---------- MENU CADASTRAR FUNCIONÁRIO ------------------
Id do Funcionário: 4297914
Por favor entre com o nome do Funcionário: Bruno
Por favor entre com o setor do Funcionário: Configurações
Por favor entre com o salário do Funcionário: 1000.00
--------------------------------------------------

**Figura 4.1: Exemplo de saída de console que o aluno deve fazer. Apresenta o print com seu nome completo e é realizado o cadastro do primeiro funcionário, note que o ID do funcionário inicia em 4297914 pois o id\_global é inicializado em 4297913.**

A imagem apresenta o seguinte texto de saída de console:

--------------- MENU PRINCIPAL -------------------
Escolha a opção desejada:
1 - Cadastrar Funcionários
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - Sair
>>1
--------------------------------------------------
---------- MENU CADASTRAR FUNCIONÁRIO ------------------
Id do Funcionário: 4297915
Por favor entre com o nome do Funcionário: Tamy
Por favor entre com o setor do Funcionário: Tutorias
Por favor entre com o salário do Funcionário: 1000
--------------------------------------------------

--------------------------------------------------
--------------- MENU PRINCIPAL -------------------
Escolha a opção desejada:
1 - Cadastrar Funcionários
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - Sair
>>1
--------------------------------------------------
---------- MENU CADASTRAR FUNCIONÁRIO ------------------
Id do Funcionário: 4297916
Por favor entre com o nome do Funcionário: Osmar
Por favor entre com o setor do Funcionário: Tutorias
Por favor entre com o salário do Funcionário: 1000.00
--------------------------------------------------

**Figura 4.2: Exemplo de saída de console que o aluno deve fazer. São cadastrados mais dois funcionários com mesmo setor.**

**A imagem apresenta o seguinte texto de saída de console:

--------------------------------------------------
--------------- MENU PRINCIPAL -------------------
Escolha a opção desejada:
1 - Cadastrar Funcionários
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - Sair
>>2
--------------------------------------------------
---------- MENU CONSULTAR FUNCIONÁRIO ------------------
Escolha a opção desejada:
1 - Consultar Todos os Funcionários
2 - Consultar Funcionário por id
3 - Consultar Funcionário(s) por setor
4 - Retornar
>>1
----------------
id: 4297914
nome: Bruno
setor: Configurações
salário: 1000.0

id: 4297915
nome: Tamy
setor: Tutorias
salário: 1000.0

id: 4297916
nome: Osmar
setor: Tutorias
salário: 1000.0

----------------**

**Figura 4.3: Exemplo de saída de console que o aluno deve fazer. Em que se consulta Todos os funcionários cadastrados.**

A imagem apresenta o seguinte texto de saída de console: 

--------------------------------------------------
---------- MENU CONSULTAR FUNCIONÁRIO ------------------
Escolha a opção desejada:
1 - Consultar Todos os Funcionários
2 - Consultar Funcionário por id
3 - Consultar Funcionário(s) por setor
4 - Retornar
>>2
Digite o id do funcionário: 4297914
----------------
id: 4297914
nome: Bruno
setor: Configurações
salario: 1000.0

----------------
--------------------------------------------------

--------------------------------------------------
---------- MENU CONSULTAR FUNCIONÁRIO ------------------
Escolha a opção desejada:
1 - Consultar Todos os Funcionários
2 - Consultar Funcionário por id
3 - Consultar Funcionário(s) por setor
4 - Retornar
>>3
Digite o setor do(s) funcionário(s): Tutorias
----------------
id: 4297915
nome: Tamy
setor: Tutorias
salario: 1000.0

id: 4297916
nome: Osmar
setor: Tutorias
salario: 1000.0

----------------

Figura 4.4: Exemplo de saída de console que o aluno deve fazer. Em que se consulta o funcionário com id número 4297914 e consulta pelo nome do setor (Tutorias).

A imagem apresenta o seguinte texto de saída do console:

--------------- MENU PRINCIPAL -------------------
Escolha a opção desejada:
1 - Cadastrar Funcionários
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - Sair
>>3
--------------------------------------------------
------------ MENU REMOVER FUNCIONÁRIO ------------------
Digite o id do funcionario a ser removido: 4297914
Funcionário removido com sucesso!
--------------------------------------------------
--------------- MENU PRINCIPAL -------------------
Escolha a opção desejada:
1 - Cadastrar Funcionários
2 - Consultar Funcionário(s)
3 - Remover Funcionário
4 - Sair
>>2
--------------------------------------------------
---------- MENU CONSULTAR FUNCIONÁRIO ------------------
Escolha a opção desejada:
1 - Consultar Todos os Funcionários
2 - Consultar Funcionário por id
3 - Consultar Funcionário(s) por setor
4 - Retornar
>>1
----------------
id: 4297915
nome: Tamy
setor: Tutorias
salário: 1000.0

id: 4297916
nome: Osmar
setor: Tutorias
salário: 1000.0

Figura 4.5: Exemplo de saída de console que o aluno deve fazer. Em que se remove o funcionário de Id número 4297914 e depois se faz uma consulta de todos os funcionários.

Apresentação de **Código da Questão 4**:

# =============================== IMPORTS ==================================

from pprint import pprint

# ============================== VARIAVEIS =================================

listar\_funcionarios = []

id\_global = 5016927

# =============================== FUNCOES ==================================

# funcao para cadastra um funcionario

def cadastrar\_funcionario(id):

    print("---------- MENU DE CADASTRAR FUNCIONARIOS ---------")

    print(f"Id do funcionario: {id}")

    nome = input("Nome do funcionario: ")

    setor = input("Nome do setor: ")

    while True:

        # try para confirmar que seja um input valido

        try:

            salario = float(input("Salario do funcionario: "))

            break

        except ValueError:

            print("Error. Try again")

    # Colocando os dados dentro de um Dicionario

    funcionario = {

        "id": id,

        "nome": nome,

        "setor": setor,

        "salario": salario

    }

    # Colocando um dicionario dentro de uma lista, EXIGENCIA 7??

    listar\_funcionarios.append(funcionario.copy())

def consultar\_funcionarios():

    print("------------- CONSULATR FUNCIONARIOS -------------")

    while True:

        # try para confirmar que seja um input valido

        try:

            option = int(input("Qual opção deseja?\n - 1. Consultar Todos\n - 2. Consultar por Id\n - 3. Consultar por Setor\n - 4. Retornar ao menu\n >> "))

            if option in (1, 2, 3, 4):

                # CONSULTAR TODOS

                if option == 1:

                    pprint(listar\_funcionarios)

                # CONSULTAR POR ID

                elif option == 2:

                    while True:

                        # try para confirmar que seja um ID valido

                        try:

                            funcionario\_id = int(input("Entre o ID do funcionario: "))

                            found = False

                            # loop para ver se o ID exisite

                            for funcionario in listar\_funcionarios:

                                if funcionario["id"] == funcionario\_id:

                                    pprint(funcionario)

                                    found = True

                                    break

                            # Caso ID nao existe vai informar.

                            if not found:

                                print("Id doesn't exists. Try a different one.")

                            # Caso existe vai sair do loop

                            else:

                                break

                        except ValueError:

                            print("Id invalida, tenta de novo.")

                # CONSULTAR POR SETOR

                elif option == 3:

                    while True:

                        try:

                            funcionario\_setor = input("Entre o setor do funcionário: ")

                            found = False

                            # Loop through the list to find all employees in the specified sector

                            for funcionario in listar\_funcionarios:

                                if funcionario["setor"] == funcionario\_setor:

                                    pprint(funcionario)

                                    found = True

                            # If no employees are found in the sector, inform the user

                            if not found:

                                print("Setor não existe. Tente de novo.")

                            else:

                                break  # Exit the loop after printing all employees in the sector

                        except ValueError:

                            print("Setor inválido, tente de novo.")

                # sair da da funcao

                elif option == 4:

                    return

            else:

                print("Not a valid option.")

        except ValueError:

            print("opcao invalida, tenta de novo.")

def remover\_funcionario():

    print("--------------- REMOVER FUNCIONARIO --------------")

    while True:

        # try para confirmar que seja um input valido

        try:

            funcionario\_id = int(input("Digite o ID do funcionario que deseja excluir: "))

            # loop para encontrar o funcionario com o mesmo id

            for funcionario in listar\_funcionarios:

                # se o funcionario com o ID existe sera removido.

                if funcionario["id"] == funcionario\_id:

                    listar\_funcionarios.remove(funcionario)

                    print(f"Funcionario com id {funcionario\_id} foi removido.")

                    # sair da funcao

                    return

            # caso nao existe, vai pedir pra fazer novamente

            print("Id invalido. Tente novamente")

        # caso nao existe, vai pedir pra fazer novemente

        except ValueError:

            print("Id invalido. Tente novamente.")

# ========================== PROGRAMA PRINCIPAL ============================

print("Bem vindos a empresa do Caleb Hall!")

print("-----------------------------------")

while True:

    print("=========== MENU PRINCIPAL =========")

    # try pra confirmar que seja input valido

    try:

        option = int(input("Qual opção deseja\n - 1. Cadastrar Funcionário\n - 2. Consultar Funcionário\n - 3. Remover Funcionário\n - 4. Encerrar Programa\n >>"))

        if option == 1:

            # chamar funcao

            id\_global += 1

            cadastrar\_funcionario(id\_global)

        elif option == 2:

            # chamar funcao

            consultar\_funcionarios()

        elif option == 3:

            # chamar funcao

            remover\_funcionario()

        else:

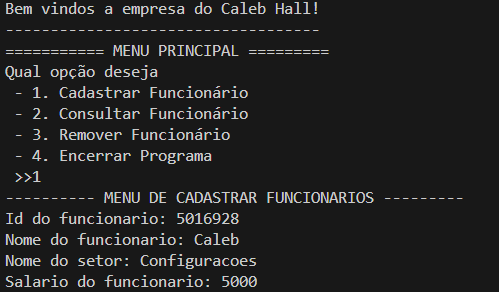
            # sair

            break

    except ValueError:

        print("Invalid option. Try again.")

Apresentação de **Saída do Console da Questão 4**:



A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated