

```

# Lista de produtos: cada item contém nome, quantidade em estoque e preço
produtos = [
    ["Coca-Cola", 2, 3.75],
    ["Pepsi", 5, 3.67],
    ["Monster", 1, 9.96],
    ["Café", 100, 1.25],
    ["Redbull", 2, 13.99]
]

# Lista de valores aceitos: cada item tem o valor da nota/moeda e a quantidade
disponível na máquina
valores = [
    [200, 1],
    [100, 10],
    [50, 10],
    [20, 10],
    [10, 10],
    [5, 10],
    [2, 10],
    [1, 10],
    [0.50, 10],
    [0.25, 10],
    [0.1, 10],
    [0.05, 10],
    [0.01, 10]
]

# Lista auxiliar para armazenar o troco a ser entregue na operação atual
lista_troco = []

# Função para checar e validar as entradas do usuário
def check_input(value, x):
    if x == 1: # Argumento passado para decidir se verá para inteiro ou para
float, nesse caso: inteiro
        try:
            entrada_checada = int(value)
            if entrada_checada < 1 or entrada_checada > len(produtos): # Checa se
escolha existe na Máquina
                print("\nEscolha inexistente.\n")
                return None
            # Verifica se o produto está esgotado
            if produtos[entrada_checada-1][1] == 0:
                print("\nProduto esgotado :/ Informe um Administrador.\n")
                return None
            return entrada_checada
        except ValueError:
            print("\nEntrada incorreta. Digite um número inteiro.\n")
            return None

    elif x == 2: # Verifica valor de pagamento
        try:
            pay_checado = float(value)
            return pay_checado
        except ValueError:
            print("\nEntrada incorreta. Digite um número real.\n")
            return None
    return None

# Função para calcular e separar o troco

```

```

def check_divide(value):
    lista_troco.clear() # Limpa lista auxiliar

    # Cria uma cópia da lista de valores para simular a entrega de troco sem afetar
    a original
    caixa_simulado = [copia.copy() for copia in valores]
    troco_simulado = [] # Lista para armazenar a simulação do troco
    valor_restante = round(value, 2) # Valor que falta devolver

    # Percorre as notas/moedas disponíveis
    for i in range(len(caixa_simulado)):
        # Enquanto ainda houver valor a devolver e notas/moedas disponíveis
        while valor_restante >= caixa_simulado[i][0] and caixa_simulado[i][1] > 0:
            valor_restante = round(valor_restante - caixa_simulado[i][0], 2) #
            Atualiza o valor restante
            caixa_simulado[i][1] -= 1 # Diminui a quantidade daquela nota/moeda.

            # Atualiza ou adiciona a quantia usada na simulação do troco
            for item in troco_simulado: # Se já tem aquela nota/moeda na lista de
            troco simulado, só aumenta a quantidade.
                if item[0] == caixa_simulado[i][0]:
                    item[1] += 1
                    break
            else:
                troco_simulado.append([caixa_simulado[i][0], 1]) # adiciona esse
                novo valor na lista com a quantidade 1.

    # Se não for possível devolver o valor com as notas/moedas disponíveis
    if valor_restante > 0.009:
        print(f"\nNão há troco suficiente para devolver R$ {value:.2f}. Venda
        cancelada.")
        return False

    # Exibe o troco que será entregue
    print(f"\nVOCÊ RECEBERÁ UM TROCO DE R$ {value:.2f} DA SEGUINTE FORMA:")

    for item in troco_simulado:
        # Decide pluralização adequada (nota, notas, moeda, moedas)
        if item[1] > 1 and item[0] > 1:
            notaOuMoeda = "Notas"
        elif item[1] == 1 and item[0] > 1:
            notaOuMoeda = "Nota"
        elif item[1] > 1 and item[0] <= 1:
            notaOuMoeda = "Moedas"
        else:
            notaOuMoeda = "Moeda"

        # Exibe quantidade e valor de cada item de troco
        print(f"{item[1]} {notaOuMoeda} de {item[0]:.2f} R$")
        lista_troco.append(item)

    # Desconta do caixa oficial o troco dado
    for atualizar in valores:
        if atualizar[0] == item[0]:
            atualizar[1] -= item[1]
            break

    print("\nTroco entregue com sucesso!")
    return True

```

```

# Loop principal do sistema
while True:
    # Exibe os produtos disponíveis
    print("\nProdutos disponíveis:\n")
    for i in range(len(produtos)):
        print(f"Código: {i+1} -> {produtos[i][0]} | {produtos[i][1]} unidades |
Preço: R$ {produtos[i][2]:.2f}")

    # Solicita código do produto ou comando
    escolha = input("\nDigite o código do produto que deseja (ou 'sair' para
encerrar): ")

    if escolha.lower() == "sair":
        print("\nEncerrando operação. Obrigado!\n")
        break

    # Modo administrador via senha '1234'
    if escolha == "1234":
        while True:
            try:
                # Menu administrador
                inputAdm = int(input("\nBem vindo administrador."
                "\nOpções: "
                "\n| Alterar produtos (1) | "
                "\n| Alterar quantia de dinheiro (2) | "
                "\n| Sair do modo administrador (3) | "
                "\n| Checar produtos (4) | "
                "\n| Checar dinheiro na maquina (5) |\n"
                "\nEscolha: "))

            except ValueError:
                print("Entrada inválida.")
                continue

            if inputAdm == 3: #Sair do loop caso desejado.
                break

            elif inputAdm == 1: # Alterar produtos
                try: #Try para failsafe
                    admProduto = int(input("| Alterar produtos a venda (1) |"
                    "\n| Alterar quantidades disponíveis (2)
| "
                    "\n| Voltar (3) |\n"))

                except ValueError:
                    print("Entrada inválida.")
                    continue

                if admProduto == 3:
                    continue

                elif admProduto == 1: #alterar catalogo de produtos
                    try:
                        alterarProd = int(input("| Adicionar novo produto (1) |\n|
Remover produto (2) |\n| Voltar (3) |\n"))
                    except ValueError:
                        print("Entrada inválida.")
                        continue

                    if alterarProd == 1: #Adicionar um novo produto ao catalogo

```

```

        newProd = input("Insira o nome do novo produto ou 3 para
voltar: ")
        if newProd != "3":
            try:
                qntNewProd = int(input("Quantidade a ser
adicionada: "))
                priceNewProd = float(input("Preço do novo produto:
").replace(",","."))
                produtos.append([newProd, qntNewProd,
priceNewProd]) #Insere na matriz de produtos a venda as variaveis entregue pelo
usuario, contendo nome, quantidade e preco de produtos
            except ValueError:
                print("Valor inválido.")

        if alterarProd == 2:
            print("\nProdutos disponíveis:\n") #Remover um produto de
catalogo
            for i in range(len(produtos)):
                print(f"Código: {i+1} -> {produtos[i][0]} |
{produtos[i][1]} unidades | Preço: R$ {produtos[i][2]:.2f}")

            try:
                removeProd = int(input("Insira o código do produto a
ser removido ou 3 para voltar: "))
                if removeProd != 3:
                    if 1 <= removeProd <= len(produtos):
                        produtos.pop(removeProd-1) #Remove o produto em
questao da lista de produtos
                    else:
                        print("Código inválido.")
            except ValueError:
                print("Valor inválido.")

        elif admProduto == 2: #Alterar quantidades armazenadas de
determinado produto na maquina
            try:
                addOrRemoveProdutos = int(input("| Adicionar quantidades
(1)| \n| Remover quantidades (2) | \n| Voltar (3) |\n"))
            except ValueError:
                print("Entrada inválida.")
                continue

            if addOrRemoveProdutos != 3:
                print("\nProdutos disponíveis:\n")
                for i in range(len(produtos)): #Print dos produtos
disponiveis para auxiliar monitoramento
                    print(f"Código: {i+1} -> {produtos[i][0]} |
{produtos[i][1]} unidades | Preço: R$ {produtos[i][2]:.2f}")

            try:
                addProd = int(input("\nInsira o código do produto que
deseja alterar ou 3 para voltar: "))
                if addProd != 3:
                    if 1 <= addProd <= len(produtos): #Adiciona ou
remove unidades de um produto a depender da escolha do user
                        qntProdutos = int(input("Quantas unidades? "))
                        if addOrRemoveProdutos == 1:
                            produtos[addProd-1][1] += qntProdutos
                            print("\nProduto adicionado com sucesso!")

```

```

        elif addOrRemoveProdutos == 2:
            produtos[addProd-1][1] -= qntProdutos
            print("\nProduto removido com sucesso!")
        else:
            print("Código inválido.")
    except ValueError:
        print("Valor inválido.")

    elif inputAdm == 2: # Alterar valores de dinheiro armazenado
        print("\nDinheiro na máquina:\n")
        for i in range(len(valores)): # Print da quantia de dinheiro
            armazenada
            print(f"Notas/moedas de {valores[i][0]} -> {valores[i][1]}
            (Código {i+1})")

        try:
            addOrRemoveTroco = int(input("| Adicionar dinheiro (1) | \n|
Retirar dinheiro (2) | \n| Voltar (3) |\n"))
        except ValueError:
            print("Entrada inválida.")
            continue

        if addOrRemoveTroco == 3:
            continue
        elif addOrRemoveTroco in [1, 2]: # Caso a escolha seja adicionr ou
retirar dinheiro:
            try:
                escolhaTroco = int(input("Qual moeda/nota deseja alterar?
"))
                if 1 <= escolhaTroco <= len(valores):
                    quantia = int(input("Quantia a alterar: "))
                    if addOrRemoveTroco == 1:
                        valores[escolhaTroco-1][1] += quantia
                        print("\nTroco adicionado com sucesso!")
                    else:
                        if quantia <= valores[escolhaTroco-1][1]:
                            valores[escolhaTroco-1][1] -= quantia
                            print("\nTroco removido com sucesso!")
                        else:
                            print("\nNão há essa quantia para remover.")
                    else:
                        print("Código inválido.")
                except ValueError:
                    print("Entrada inválida.")

            elif inputAdm == 4: # Listar produtos
                print("\nProdutos disponíveis:\n")
                for i in range(len(produtos)):
                    print(f"Código: {i+1} -> {produtos[i][0]} | {produtos[i][1]}
unidades | Preço: R$ {produtos[i][2]:.2f}")

            elif inputAdm == 5: # Listar dinheiro na máquina
                print("\nDinheiro na máquina:\n")
                for i in range(len(valores)):
                    print(f"Notas/moedas de {valores[i][0]} -> {valores[i][1]}")
            else:
                print("Input inválido.")

    if escolha == "1234": #Evita o print de "input invalido" para o codigo de adm

```

```

        continue
    else:
        cod_produto = check_input(escolha, 1) #Validar código de produto
selecionado pelo cliente

        if cod_produto is not None:
            produto_selecionado = produtos[cod_produto-1] # Define o produto a partir
do código digitado (-1 devido ao funcionamento de index das listas)
            pay = input(f"\nAgora digite qual o valor com que irá pagar (use ponto ou
vírgula): ").replace(",",".")
            valor_pago = check_input(pay, 2) # Checa para ver se o Input é válido
(dessa vez pra float)

            if valor_pago is not None: # None é a resposta do check_input() para caso
de falha nos tries
                preco = produto_selecionado[2]

                if valor_pago < preco: # Caso queiram pagar menos que o valor do
produto
                    print("\nValor insuficiente para realizar a compra. Compra
cancelada.\n")
                else:
                    troco = round(valor_pago - preco, 2) # Define o troco com duas
casas decimais
                    if troco == 0: # # Caso paguem exato
                        print("\nPagamento exato, sem troco. Obrigado!")
                        produtos[cod_produto-1][1] -= 1 # Remove 1 do produto escolhido
do estoque
                    elif check_divide(troco): # Caso precisem de troco
                        print("\nObrigado pela compra!")
                        produtos[cod_produto-1][1] -= 1 # Remove 1 do produto escolhido
do estoque
                    else:
                        print("\nCompra não realizada.") # Caso cancelamento

```