

MAC0110/0115 — EP 2 - 2023

Junior Barrera (MAC115)

Nelson Lago (MAC115)

Paulo Meirelles (MAC110)

Roberto Marcondes (MAC110)

29 de novembro de 2023

1 Introdução

Após uma bem sucedida campanha de investimentos de alta rentabilidade para aquisição de novos clientes, o Bank 110 popularizou sua marca com caixas eletrônicos em pontos estratégicos, e novamente te contratou para implementar seu sistema.

Os vários caixas eletrônicos espalhados pela cidade deverão ser capazes de realizar depósitos em dinheiro, saques e transferências entre contas do próprio banco. Ao final do dia, o banco precisa de um relatório com a lista de clientes e seu saldo, além de um relatório contendo a quantidade atualizada de notas em cada caixa eletrônico para atualizar sua reposição no próximo dia.

2 Descrição

Você deverá implementar um programa em Python que recebe: (1) a quantidade de caixas eletrônicos; e (2) as quantidades de cada nota disponíveis inicialmente em cada caixa eletrônico. Após essa configuração inicial, o programa deve realizar as operações solicitadas pelos clientes, como veremos abaixo. A cada operação realizada, o saldo do(s) cliente(s) envolvido(s) e a quantidade de notas do caixa em que a operação foi realizada serão atualizados.

Você deverá representar os clientes e seus respectivos saldos como um par de **listas** (uma lista para os nomes e outra lista para os saldos) e a quantidade de notas dos caixas eletrônicos como uma **matriz** (as linhas são os caixas e as colunas são as quantidades de cada nota em cada caixa).

O depósito deverá ser implementado na função *deposito(cliente, notas, cn)*, em que *cliente* se refere ao nome do cliente cuja conta será creditada, *notas* se refere a uma lista com a quantidade de notas de cada valor e *cn* se refere ao número do caixa em que o depósito será realizado. Se o

cliente não tiver uma conta, ela será criada no ato do depósito. Nesta função, o saldo do cliente e a quantidade de notas do caixa serão atualizados.

Na função *saque(cliente, valor, cn)*, o seu programa deverá calcular e imprimir a menor quantidade de notas de cada valor necessárias para o saque de acordo com a sua disponibilidade no caixa *cn* e o saldo do cliente. Caso não haja notas suficientes, o saque não será realizado. Nesta função, o saldo do cliente e a quantidade de notas do caixa serão atualizados.

A função *transferencia(cliente, valor, cliente_credito)* deverá transferir o *valor* do *cliente* para o *cliente_credito*. Nesta função, os saldos dos clientes serão atualizados, mas a quantidade de notas no caixa não.

Importante: Você deve usar apenas os recursos do Python previstos em nosso cronograma até a Prova 2 (P2).

2.1 Entrada

A quantidade de caixas eletrônicos é dada pelo inteiro $2 \leq N \leq 10$. Para cada um dos N caixas eletrônicos, os valores q_{100} , q_{50} , q_{20} , q_{10} , q_5 , q_2 e q_1 são inteiros correspondentes às notas disponíveis de 100, 50, 20, 10, 5, 2 e 1 reais respectivamente.

Seu programa deve iniciar “carregando” os caixas eletrônicos com as notas:

Digite a quantidade de caixas eletrônicos: N

Digite a quantidade de notas do caixa 1: q_{100} q_{50} q_{20} q_{10} q_5 q_2 q_1

Digite a quantidade de notas do caixa 2: q_{100} q_{50} q_{20} q_{10} q_5 q_2 q_1

...

Digite a quantidade de notas do caixa N : q_{100} q_{50} q_{20} q_{10} q_5 q_2 q_1

A seguir, o programa deve permitir aos clientes realizar as operações, que são identificadas por inteiros (1 é depósito, 2 é saque e 3 é transferência):

Escolha a operação desejada: [1|2|3]

Digite o número do caixa: cn

Digite seu nome: *cliente*

E então, dependendo da operação:

Digite a quantidade de notas para depósito: q_{100} q_{50} q_{20} q_{10} q_5 q_2 q_1

ou

Digite o valor do saque: V

ou

Digite o valor da transferência: T

Digite o nome da pessoa que será creditada: *cliente_credito*

Se a operação for de depósito (1), o cliente vai entrar com a quantidade de notas de cada valor: q_{100} , q_{50} , q_{20} , q_{10} , q_5 , q_2 e q_1 . Se a operação for de saque (2), o cliente vai entrar com o valor do saque: V . Se a operação for de transferência (3), o cliente vai entrar com o valor da transferência T e o nome da pessoa *cliente_credito* que receberá esse valor.

Como sempre, os dados de entrada e saída deverão seguir **estritamente** o modelo e a ordem apresentados. Para mais detalhes, veja o exemplo de entrada e o exemplo de saída.

Importante: Neste EP assumimos que todas as entradas estarão dentro dos limites estabelecidos neste documento e não será necessário testá-las. Na operação de transferência o cliente a ser creditado terá uma conta em banco necessariamente, então não é preciso testar isso.

2.2 Saída

Se a operação solicitada for 0, o sistema deverá gerar o relatório final, com (1) a lista de clientes e seus saldos; e (2) a quantidade final de notas de cada caixa eletrônico.

A saída deve conter a lista de clientes e seus respectivos saldos. Primeiro deverá ser impressa uma linha vazia seguida pela frase: “Dados dos clientes:” e uma nova linha com a frase: “Nome: saldo (R\$)”. Depois deverá ser impressa a lista de clientes e saldos conforme o exemplo de saída na Seção 2.4.

Após o último cliente, deve ser impressa uma linha vazia seguida pela frase: “Relatório de caixas:”, seguida por “caixa: n100 n50 n20 n10 n5 n2 n1” em uma nova linha, e finalmente uma tabela contendo em cada linha a quantidade final de notas de cada caixa, conforme o exemplo de saída.

2.3 Exemplo de entrada

```
Digite a quantidade de caixas eletrônicos: 2
Digite a quantidade de notas do caixa 1: 10 10 10 10 10 10 10
Digite a quantidade de notas do caixa 2: 1 2 5 10 20 50 100
Escolha a operação desejada: 1
Digite o número do caixa: 1
Digite seu nome: Alan
Digite a quantidade de notas para depósito: 0 0 0 10 0 0 0
Escolha a operação desejada: 1
Digite o número do caixa: 1
Digite seu nome: Ada
Digite a quantidade de notas para depósito: 10 0 0 0 0 0 0
Escolha a operação desejada: 2
Digite o número do caixa: 2
Digite seu nome: Ada
Digite o valor do saque: 333
Escolha a operação desejada: 3
Digite o número do caixa: 1
Digite seu nome: Ada
Digite o valor da transferência: 100
Digite o nome da pessoa que será creditada: Alan
Escolha a operação desejada: 0
```

2.4 Exemplo de saída

Dados dos clientes:

Nome: saldo (R\$)

Alan: 200

Ada: 567

Relatório de caixas:

caixa: n100 n50 n20 n10 n5 n2 n1

1: 20 10 10 20 10 10 10

2: 0 0 0 7 20 49 99

3 Entrega

3.1 Plágio e cópia de EPs

Plágio é a cópia/modificação não autorizada e/ou sem o conhecimento do autor original. O plágio é um problema grave que pode levar até a expulsão do aluno da universidade. Leia o Código de Ética da USP (em particular, a seção V): http://www.mp.usp.br/sites/default/files/arquivosanexos/codigo_de_etica_da_usp.pdf.

A cópia de EPs entre alunos implicará na anulação da nota de todos os envolvidos. O caso poderá ser levado à Comissão de Graduação do Instituto.

3.2 Instruções de entrega, data de entrega e notas

Você deverá entregar um único arquivo python denominado `ep2.py` na atividade do EP2 no e-Disciplinas seguindo as “**Instruções para entrega de EPs**”.

A entrega deverá ser feita até as 23:59h do dia 11 de dezembro de 2023.

A correção será feita de forma automática e vai considerar 10 casos de teste, cada um valendo 1 ponto.

3.3 Ponto extra na Prova 2

Embora o assunto de ordenação seja um assunto da disciplina e que será ministrado nas últimas aulas, decidimos desconsiderá-lo no EP2. No entanto, para aqueles que já fizeram a ordenação ou que querem superar este desafio, disponibilizaremos a entrega do “EP2-extra” com um caso de teste que considera a ordenação da lista de clientes no relatório final. A impressão do relatório final deve conter a lista de clientes em ordem alfabética e seus respectivos saldos. Quem fizer o EP2-extra passando no teste ganhará um ponto extra na Prova 2.