

PB-TP3-25T3

Prof. LP Maia

O trabalho proposto deve implementar um caixa de supermercado de forma que o operador do caixa insira os produtos e as quantidades e, ao final, exiba uma “nota fiscal” com os itens comprados e o total da compra. Ao final do programa, deverá ser exibido um extrato com o total das vendas do caixa. Considere que existe apenas um caixa operando no supermercado.

A versão inicial do projeto, versão 1, não deverá implementar estruturas do tipo dicionário, apenas lista de listas. Também não deve ser utilizado o conceito de Orientação a Objetos, ou seja, classes. A versão 2 deverá implementar uma lista de dicionários. A versão 3 deverá implementar uma classe Produto e uma lista de produtos. A versão 4 deverá implementar um banco de dados, chamado mercado, com uma tabela de produtos, utilizando SQL puro. A versão 5, e última, deverá implementar o BD com SQLAlchemy.

1. Crie um diretório para armazenar todos os programas e arquivos desenvolvidos neste projeto, inclusive o arquivo de produtos, descrito posteriormente, que será processado na abertura e fechamento do caixa.
2. Inicialmente, o programa deverá abrir o caixa e ler um arquivo de produtos no formato CSV (produtos.csv) com os dados abaixo e armazenar em uma lista de produtos. Cada produto tem um id, um nome, a quantidade em estoque e o seu preço.

1,Produto 1,1,10
2,Produto 2,2,20
3,Produto 3,3,30
4,Produto 4,4,40
5,Produto 5,5,50
3. Ao final do programa, ou seja, no fechamento do caixa, a lista de produtos deverá ser gravada no mesmo arquivo com as alterações feitas durante o processamento, que serão detalhadas nos próximos itens.
4. Após a abertura do caixa, o atendente poderá realizar o atendimento de vários clientes. Para cada cliente, o programa deverá exibir a opção de “Iniciar atendimento” e ao final do atendimento exibir a opção “Finalizar atendimento”.
5. Cada cliente, deverá ser identificado como “Cliente #”, onde # deverá ser o número do cliente atendido. Por exemplo, o primeiro cliente deverá ser identificado como “Cliente 1”, o segundo como “Cliente 2” e assim por diante.
6. Para cada cliente, o atendente deve inserir uma lista de itens referentes a cada produto comprado. Cada item deve ser armazenado em uma lista de itens que deve conter o id do item, o nome do produto, a quantidade do produto, o preço unitário e o preço total do item.
7. O preço total do item é a multiplicação do preço unitário do produto e a quantidade do produto comprado pelo cliente.

8. Para cada item, o atendente deve inserir o id do produto e a quantidade a ser comprada. O id do produto deve constar da lista de produtos. Por exemplo, no arquivo produtos.csv, apresentado anteriormente, o id do produto deve estar entre um e cinco, caso contrário o programa deverá exibir uma mensagem de erro “Erro: produto não cadastrado.”
9. A quantidade de produtos comprados deve ser maior que zero, caso contrário o programa deverá exibir uma mensagem de erro “Erro: quantidade deve ser maior que zero”. Não se preocupe se a quantidade de produtos no estoque estará disponível para a compra.
10. Para cada item comprado, o programa deverá dar a baixa no estoque, ou seja, subtrair a quantidade do produto comprado da lista de produtos.
11. Quando o caixa finalizar o atendimento de um cliente, o programa deverá exibir uma “nota fiscal” com as informações abaixo. Para formatar os dados de saída em forma de tabela, você pode utilizar a formatação do próprio Python ou utilizar um pacote que facilite esse processo. Nos exemplos abaixo, foi utilizado o pacote [Tabulate](#).

```
Cliente 1
Data: 29/11/2024 10:47
```

Item	Produto	Quant.	Preço	Total
1	Produto 1	1	10	10
2	Produto 3	2	30	60

```
Itens: 2
Total: 70
```



```
Cliente 2
Data: 29/11/2024 10:50
```

Item	Produto	Quant.	Preço	Total
1	Produto 2	2	20	40
2	Produto 4	3	40	120

```
Itens: 2
Total: 160
```

12. Quando o atendente fechar o caixa, o programa deverá exibir uma lista de clientes atendidos com o total de compra de cada cliente e o total de vendas realizadas, como no exemplo abaixo.

```
Fechamento do caixa
Data: 29/11/2024 10:59

Cliente      Total
-----
Cliente 1    70
Cliente 2    160

Total de vendas: 230
```

13. Além disso, o programa deverá exibir uma lista de produtos que estejam sem estoque, ou seja, a quantidade do produto igual a zero. Considerando os exemplos anteriores, o programa exibirá a saída abaixo.

```
Produtos sem estoque:
Produto 1
Produto 2
```

14. Finalmente, o programa deverá gravar o arquivo de produtos atualizado, ou seja, com o novo estoque de cada produto vendido.
15. O sistema considerou que apenas um caixa estava operando no supermercado. Cite um problema caso houvesse mais de um caixa operando ao mesmo tempo no supermercado?
16. Qual seria uma possível solução para o problema anterior considerando o conteúdo apresentado no bloco?

Observações importantes:

- Utilize as melhores práticas de codificação e refatoração apresentadas em aula;
- Crie arquivos que organizem as funções do programa;
- Utilize funções pequenas que recebam poucos parâmetros e, de preferência, retorne apenas um valor;
- Realize tratamento de possíveis erros;
- Não utilize variáveis globais, apenas passagem de parâmetros.

A entrega do projeto desenvolvido nesta etapa deve ser feita em um arquivo único no formato zip, seguindo a nomenclatura: nome_sobrenome_PB_TP3.zip. Para facilitar a correção, crie uma pasta PB_TP 3 e, dentro desta pasta, crie cinco pastas uma para cada versão do projeto. Por exemplo, PB_TP3_v1, PB_TP3_v2 etc.