



Universidade de Brasília

Tópicos em Redes de Comunicação 3

2023.1

Desafio 2 – Registradora de Produtos

O objetivo principal do desafio 2 será criar um sistema registrador de produtos conectado a um banco de dados através da linguagem de programação Python. Ele deverá ter 5 funções básicas:

1. Listar o estoque (**listar_estoque**);
2. Adicionar um item (**adicionar_item**);
3. Remove-lo (**remover_item**); ou
4. Atualizá-lo (**atualizar_item**)
5. Realizar compra (**realizar_compra**).

Inicialmente, cria-se uma conexão ao banco de dados para armazenar os produtos em estoque. Para este desafio, foi utilizada a biblioteca **SQLite3**[1], que possibilita criar um banco de dados de maneira simplificada e sem a necessidade de um servidor separado.

Logo após a conexão, é verificado se a tabela de estoque já existe e se está vazia. O programa deve ser capaz de criar uma tabela, caso não exista uma, e preenche-la com produtos, caso esteja vazia.

O programa devera exibir uma tela de controle para que o usuário determine qual operação se deseja realizar no banco de dados e, de acordo com sua resposta, a determinada função será executada.

```
Controle de estoque
0 - Sair
1 - Listar estoque
2 - Adicionar item
3 - Remover item
4 - Atualizar item
5 - Realizar compra
-----
Atenção: os seguintes produtos estão com estoque baixo:
Tênis (ID 3): 3 unidades
```

a) **listar_estoque()**

Esta função terá dois trabalhos principais: uma busca de todos os registros na tabela do estoque e uma exibição de cada um deles. Para este último, é necessário elaborar uma estrutura de repetição *for* que percorrerá a tabela e exibirá cada coluna em ordem de acordo com a linha de cada produto.

Ela deve exibir todos os itens em estoque, junto com seu id, a quantidade e o preço, ou seja, toda a tabela que foi montada, por isso não necessitará de parâmetros específicos.

```
Escolha uma opção: 1
1 - Camisa: 10 unidades por R$ 29.99
2 - Calça: 5 unidades por R$ 59.99
3 - Tênis: 3 unidades por R$ 99.99
```

b) adicionar_item(prodoto, quantidade, preco)

A função adicionar_item deve adicionar ao estoque um produto qualquer. Esta função recebe três parâmetros (produto, quantidade e preço), que especifica qual produto será adicionado, sua quantidade e seu preço.

c) remover_item(id)

A função remover_item recebe como parâmetro apenas o id do produto. Desta forma, a função deve buscar no estoque o id escolhido e remover todas informações referentes a ele (apagar a linha inteira do banco de dados).

Para evitar problemas, será necessário alertar o usuário sempre que um produto em estoque (Quantidade > 0) estiver sendo deletado. Deve ser solicitada uma confirmação para que a operação seja executada e o produto removido do estoque.

d) atualizar_item(id, produto=None, quantidade =None, preco =None)

A função atualizar_item serve para alterar as informações de um produto dentro do estoque. Esta função pode receber até 4 parâmetros, sendo o id um parâmetro obrigatório e a base para identificar qual linha deve ser alterada.

Para atualizar um item é necessário criar um método para que se verifique se determinada coluna deve alterada ou não, de acordo com os parâmetros inseridos ao se chamar a função. Sendo assim, se houver algum valor passado, o produto, a quantidade e/ou o preço devem ser alterados.

IMPORTANTE: Qualquer função que acesse a tabela de estoque (.db), deve primeiro iniciar uma conexão com o banco de dados. Ao final de qualquer operação, é necessário executar um commit() e encerrar a conexão com o estoque.

Desafio Bônus:

Criação de uma função **verificar_estoque_baixo()** que, definindo uma quantidade mínima para cada item armazenado, gera um alerta ao usuário sempre que algum produto estiver neste limite ou abaixo, indicando o estoque em baixa.

Ela verificará a existência de algum produto com a quantidade menor que a determinada e obterá os seus dados, para em seguida gerar o aviso.

Referência:

[1] <https://docs.python.org/3/library/sqlite3.html>