

PROJETO DE PROGRAMAÇÃO

SISTEMA DE GESTÃO DE INVENTÁRIO COM SEGURANÇA DA INFORMAÇÃO EM PYTHON

Objetivo:

Desenvolver um sistema de gestão de inventário utilizando **listas, dicionários, algoritmos de busca e ordenação, tabelas, manipulação de arquivos, criptografia** e com ênfase em **segurança da informação**. O sistema será implementado em **Python**, sem o uso de banco de dados externo, e todas as operações serão realizadas sobre arquivos de texto e em memória RAM.

É importante conhecer conceitos básicos sobre criptografia, por isso visite e leia o conteúdo que está nestes links:

- <https://medium.com/curso-de-programacao-python/curso-de-criptografia-com-python-88e274d8e8af>
- <https://medium.com/curso-de-programacao-python/criptografia-com-python-1-bef5c5cf3783>
- <https://medium.com/curso-de-programacao-python/criptografia-com-fermet-no-python-7008d0c614bf>

Requisitos de programação:

1. Manipulação de arquivos de texto:

- O sistema deverá manter os produtos do inventário em um arquivo de texto chamado `inventario.csv`, ou seja, será um arquivo de texto simples, porém no formato CSV (*Comma-Separated Values*), em que o separador dos valores será um ponto e vírgula.

2. Dicionários e listas:

- Logo após o usuário ter acesso ao sistema, o arquivo `inventario.csv` deverá ser carregado para a memória do computador e os registros dos produtos serão armazenados em um dicionário. Cada produto deve ser representado por um **item** do dicionário `chave:valor` contendo as seguintes informações:
 - **ID** (número inteiro e único, é identificador do produto e será a chave de cada item);
 - **Nome** (*string*);
 - **Quantidade** (inteiro);
 - **Preço** (*float*);
 - **Importado** (booleano, indicará se o produto é importado ou não).
- O dicionário deve permitir as seguintes operações:
 - Adicionar um novo produto ao dicionário;
 - Remover um produto pelo **ID**;
 - Atualizar o nome, quantidade, preço ou se é importado ou não pelo ID;
 - Verificar a existência de um produto (por ID ou nome).

- Exibir a quantidade de produtos cadastrados e o valor total do estoque.
- Perceba que todas as operações são feitas apenas no dicionário. A transferência do que ocorreu no dicionário para o arquivo de texto deverá ser feita apenas quando o usuário escolher encerrar o programa.
- Isso é o que chamamos de processamento em lote, pois o arquivo `inventario.csv` só será atualizado após diversas operações serem feitas nos dados originais e não em tempo real, após cada operação. Isso tem vantagens e desvantagens, reflita sobre elas com sua equipe.

3. Tabelas (Dicionários):

- Note que cada produto será um registro em uma tabela, representada em um dicionário, assim, os itens do dicionário, como sempre, terão duas partes, **chave:valor**, em que a **chave** é o ID e o **valor** é uma lista com os demais campos do produto.

4. Busca e Ordenação:

- O sistema deve implementar pelo menos dois algoritmos de **busca**:
 - **Busca Linear** (para pesquisar um produto por nome, a busca será diretamente no dicionário).
 - **Busca Binária** (para pesquisar rapidamente um produto por nome em uma lista ordenada). Note que nesse caso será necessário primeiro transformar o dicionário em uma lista, ordená-la e, a partir daí, realizar a busca binária.
- O sistema deve implementar pelo menos dois algoritmos de **ordenação** e escolha de qual deles será aplicado deverá ser feita automaticamente pelo sistema:
 - **Bubble Sort, Selection Sort** ou **Insertion Sort** (para ordenar dicionários com até 100 produtos).
 - **Merge Sort** (quando a quantidade de produtos no dicionário ultrapassar 100).

5. Segurança da Informação:

- **Autenticação de usuários:**
 - O sistema deve exigir um **login simples** para que o usuário tenha acesso aos dados do inventário. Note que o usuário e a senha que permitirão acesso ao sistema ficarão guardados em um arquivo chamado `login.txt`¹ no mesmo diretório onde está o programa. Por questão de simplicidade, haverá apenas um login neste arquivo e que poderá ser alterado com uma modificação direto do programa (a primeira vez que o programa for executado esse arquivo estará vazio e será solicitado um usuário e senha iniciais).
 - Tanto o usuário quanto a senha devem ser armazenados de forma segura, utilizando **hashing**² para evitar que sejam armazenadas em texto claro.
 - Deverá existir uma opção no menu do programa para editar e sobrescrever o usuário e senha que estão no arquivo `login.txt`.
- **Hashing do arquivo de login:**

¹ Verifique o documento disponibilizado pelo professor para compreender como manipular arquivos em Python.

² <https://medium.com/@leocavalcante/hash-digest-n%C3%A3o-%C3%A9-criptografia-940374fe565c>

- Tanto o usuário quanto a senha devem ser armazenados no arquivo usando uma função de **hashing**, como **SHA-256**, para garantir que as senhas não sejam armazenadas em texto claro. Em Python, você pode usar o módulo `hashlib` para **SHA-256**

```
import hashlib

senha = 'minha_senha_secreta'

hash_senha = hashlib.sha256(senha.encode()).hexdigest()

print(hash_senha)
```

- **Criptografia do arquivo do inventário:**

- O sistema deverá cifrar todos os registros de produtos antes de armazená-los no arquivo `inventario.csv` usando um algoritmo de Cifra de Cesar criado por você mesmo³.
- Evidentemente, como os dados serão armazenados cifrados no arquivo `inventario.csv`, antes de serem carregados para o dicionário deverão ser decifrados, desse modo, enquanto estiverem no dicionário, os dados estarão desprotegidos. Reflita com sua equipe as consequências disso.
- Observação: note que cada campo do registro será cifrado quando for armazenado no arquivo, mas não o registro todo, isso quer dizer que será possível visualizar as linhas do arquivo e que cada campo está separado do outro por um ponto e vírgula, porém será incompreensível o que está em cada campo.

- **Validação de entradas:**

- As entradas devem ser validadas para garantir que o **ID** seja único, que o preço e a quantidade sejam números válidos.

6. Interface do Usuário:

- O sistema deve ter interface via terminal (não poderá ter interface gráfica), permitindo ao usuário:
 - Realizar o login com autenticação de nome de usuário e senha.
 - Adicionar, remover ou atualizar produtos no inventário.
 - Exibir todos os produtos cadastrados, ordenados por nome.
 - Buscar produtos por nome ou ID.
 - Exibir estatísticas do inventário (ex.: número total de produtos, valor total do inventário etc. seja criativo).

Exemplo de Fluxo de Trabalho:

1. Ao usar o sistema pela primeira vez:

- Caso o sistema detecte que o arquivo `login.txt` está vazio, deve pedir um usuário e senha iniciais, guardá-los no arquivo e, em seguida, continuar com o fluxo normal.
- O usuário realiza o login, informando nome de usuário e senha.
- A senha é verificada utilizando o *hash* armazenado.

³ <https://medium.com/vacatronics/cifra-de-c%C3%A9sar-em-python-8d02d3bc7d42>

- Se autenticado, o usuário tem acesso ao menu de operações e um arquivo `inventario.txt` vazio será criado.

2. Supondo que o sistema já tenha sido usado alguma vez:

- Caso o sistema detecte que o arquivo `login.txt` não está vazio, deve solicitar o usuário e senha para *login*.
- O usuário e senha fornecidos devem passar pela função de hash e o sistema deve verificar se os dados são compatíveis com os do arquivo `login.txt`, caso afirmativo, o acesso deverá ser permitido, senão, o acesso é negado e solicitado novamente o usuário e senha.
- Se o arquivo `inventario.txt` for encontrado, será decifrado e carregado para a memória do computador e o menu de operações será liberado para acesso.
- Se o arquivo `inventario.txt` não for encontrado, isso deverá ser alertado ao usuário e dada a opção de criar um novo arquivo ou, caso o usuário não queira, que ele possa reiniciar o programa após colocar o arquivo no local correto.

Entregáveis:

- **Código fonte** completo e comentado, explicando as escolhas de implementação de algoritmos de busca, ordenação e segurança.
- **Relatório técnico** explicando como a segurança foi implementada (*hashing* de senhas, criptografia de dados sensíveis, validação de entradas etc.).
- **Exemplos de entradas e saídas**, demonstrando as funcionalidades do sistema e como a segurança foi aplicada (apenas na apresentação, não deverá enviar esses dados na atividade).
- **Apresentação**, em que a equipe terá 20 minutos para mostrar o sistema funcionando para a turma e explicar detalhes de implementação.