Documento Descritivo do Sistema de Gerenciamento de Tênis Online

Introdução

Empresa Modelada

A empresa modelada é a Tênis Online, especializada na venda de tênis esportivos e casuais. O objetivo do sistema é gerenciar tanto os usuários quanto os produtos da empresa, proporcionando uma plataforma eficiente para a administração de informações.

Tipos de Usuário
O sistema possui os seguintes tipos de usuários:

Administrador: Tem acesso completo a todas as funcionalidades do sistema, incluindo gerenciamento de usuários e produtos.

Funcionário: Tem permissões limitadas, podendo adicionar, listar e atualizar produtos, mas não gerenciar usuários.

Estagiário: Tem permissões ainda mais restritas, com acesso apenas para listar produtos e usuários.

Produtos e Serviços
A empresa oferece uma variedade de produtos, principalmente tênis de diferentes marcas e estilos. Os produtos são categorizados por tipo, descrição, preço e quantidade em estoque.

Implementação

Gerenciamento de Usuários

1. Estrutura de Dados
Para armazenar as informações dos usuários, utilizamos um dicionário em Python onde a chave é o ID do usuário e o valor é um dicionário contendo os detalhes do usuário (nome, email, senha, e permissão).
Exemplo:

```
usuarios = {
 1: {'nome': 'Alice Silva', 'email': 'alice@example.com', 'senha': 'hash_da_senha', 'permissao': 'gerente'},
 2: {'nome': 'Bob', 'email': 'bob@example.com', 'senha': 'hash_da_senha', 'permissao': 'funcionario'}
}
```

2. Estrutura do Arquivo de Registro
Os dados dos usuários são armazenados em um arquivo CSV (usuarios.csv) com a seguinte estrutura de colunas:

ID: Identificador único do usuário.
Nome: Nome completo do usuário.
Email: Endereço de e-mail do usuário.
Senha: Senha do usuário (armazenada em formato hash).
Permissão: Tipo de permissão do usuário (e.g., gerente, funcionário, estagiário)

Exemplo de Conteúdo do Arquivo CSV:
graphql
ID,Nome,Email,Senha,Permissão
1,Alice Silva,alice@example.com,hash_da_senha,gerente
2,Bob,bob@example.com,hash_da_senha,funcionario

3. Funcionalidades
Create: Adicionar novos usuários (adicionar_usuario).
Read: Listar todos os usuários (listar_usuarios).
Update: Atualizar informações de um usuário existente (atualizar_usuario). Delete: Remover um usuário existente (remover_usuario).

Gerenciamento de Produtos

1. Estrutura de Dados
Para armazenar as informações dos produtos, utilizamos um dicionário em Python onde a chave é o ID do produto e o valor é um dicionário contendo os detalhes do produto (nome, descrição, preço e quantidade). Exemplo:
produtos = {
1: {'nome': 'Tênis Nike Air', 'descricao': 'Tênis de corrida', 'preco': '349.99', 'quantidade': '100'},
2: {'nome': 'Tênis Adidas', 'descricao': 'Tênis casual', 'preco': '199.99', 'quantidade': '30'}
}

2. Estrutura do Arquivo de Registro
Os dados dos produtos são armazenados em um arquivo CSV (produtos.csv) com a seguinte estrutura de colunas:
ID: Identificador único do produto.
Nome: Nome do produto.
Descrição: Descrição do produto.
Preço: Preço do produto.
Quantidade: Quantidade em estoque.

Exemplo de Conteúdo do Arquivo CSV:
ID,Nome,Descrição,Preço,Quantidade
1,Tênis Nike Air,Tênis de corrida,349.99,100
2,Tênis Adidas,Tênis casual,199.99,30

3. Funcionalidades
Create: Adicionar novos produtos (adicionar_produto).
Read: Listar todos os produtos (listar_produtos).
Update: Atualizar informações de um produto existente (atualizar_produto). Delete: Remover um produto existente (remover_produto).

Conclusão

Durante o desenvolvimento do sistema de gerenciamento de "Tênis Online", várias etapas foram desafiadoras, mas proporcionaram um grande aprendizado.

Dificuldades Encontradas

Validação de Dados: Garantir que todos os dados inseridos pelos usuários fossem válidos e seguros foi uma tarefa desafiadora. Implementar a validação de ID's únicos para evitar duplicação foi uma das

principais dificuldades. Gerenciamento de Arquivos: Manipular arquivos CSV para leitura e escrita de dados de forma eficiente e segura exigiu atenção aos detalhes, principalmente para garantir que o formato dos dados fosse mantido corretamente.
Hash de Senhas: Implementar a função de hashing para senhas foi essencial para garantir a segurança dos usuários, mas demandou um entendimento profundo sobre as bibliotecas de criptografia em Python. Muitos desses tive que pesquisar em outros lugares, pois não estava conseguindo fazer.

Escolhas Bem Sucedidas

Uso de Dicionários: A escolha de dicionários para armazenar dados de usuários e produtos foi eficaz, proporcionando acesso rápido e eficiente aos dados.
Estrutura Modular: A separação das funcionalidades em funções distintas tornou o código mais organizado e fácil de manter.
Hashing de Senhas: Implementar hashing para senhas foi uma escolha acertada para aumentar a segurança dos dados dos usuários, tive que pesquisar um pouco para seguir essas escolhas.

O Que Faltou Fazer

Interface Gráfica: O sistema atualmente funciona apenas em modo texto. Implementar uma interface gráfica tornaria o sistema mais amigável e fácil de usar para pessoas sem conhecimento técnico.
Testes Automatizados: Adicionar testes automatizados garantiria que as funcionalidades permanecessem corretas à medida que o sistema evolui, mas ainda não consegui fazer.

O Que Faria Diferente

Estrutura de Banco de Dados: Em vez de usar arquivos CSV, um banco de dados relacional, poderia ter sido utilizado para armazenar dados de forma mais robusta e escalável.
Controle de Acesso Mais Detalhado: Implementar um sistema de controle de acesso mais granular, com permissões específicas para diferentes ações, aumentaria a segurança e flexibilidade do sistema.
Documentação: Criar uma documentação mais detalhada do código e das funcionalidades ajudaria na manutenção e expansão futura do sistema, mas também não consigo fazer ainda.

Esse projeto foi uma excelente oportunidade para aplicar e consolidar conhecimentos em Python, Gerência de arquivos, Estruturas de dados(lista, set, tupla, dicionário), Estruturas de condicionais, Funções, Strings e Estruturas de repetição. As escolhas feitas foram direcionadas para alcançar um equilíbrio entre funcionalidade e simplicidade, e as lições aprendidas serão valiosas para projetos futuros.