Documentação do Sistema de Biblioteca em Python

Objetivo Geral:

Este código foi desenvolvido para criar um **sistema de biblioteca** utilizando a linguagem de programação Python com a interface gráfica fornecida pelo **Tkinter**. Ele permite que **usuários se cadastrem, façam login, reservem livros e removam reservas**.

A aplicação está dividida em duas partes principais:

1. **Sistema de Login e Cadastro de Usuários**.
2. **Sistema de Cadastro e Reserva de Livros**.

Agora, vamos explicar cada parte com mais detalhes.

### **Sistema de Login e Cadastro de Usuários**

#### **Arquivo: usuarios.csv**

Este arquivo é usado para armazenar os dados dos usuários cadastrados no sistema. Ele salva três informações de cada usuário:

* Nome
* E-mail
* Senha

#### **Funções:**

1. **inicializar\_dados\_usuarios()**:
   * Esta função verifica se o arquivo usuarios.csv já existe.
   * Se o arquivo não existir, ele será criado com os cabeçalhos: "Nome", "E-mail", "Senha".
2. **cadastrar\_usuario()**:
   * Responsável por cadastrar novos usuários.
   * Verifica se todos os campos estão preenchidos.
   * Verifica se o e-mail já foi cadastrado anteriormente.
   * Se estiver tudo correto, o novo usuário é registrado no arquivo usuarios.csv.
3. **login\_usuario()**:
   * Permite que um usuário faça login no sistema.
   * Verifica se o e-mail e a senha inseridos correspondem a um usuário cadastrado.
   * Se o login for bem-sucedido, o sistema de acervo de livros é acessado.
4. **acessar\_acervo()**:
   * Esta função é chamada após o login bem-sucedido e fecha a janela de login, iniciando o sistema de gerenciamento de livros.

### **Sistema de Cadastro e Reserva de Livros**

#### **Arquivos:**

1. **livros.csv**: Armazena os livros cadastrados, com as seguintes informações:
   * Título do livro
   * Autor
   * Ano de publicação
   * Disponibilidade (se o livro está disponível ou não)
2. **reservas.csv**: Armazena as reservas feitas pelos usuários:
   * Título do livro reservado
   * Usuário que fez a reserva (neste exemplo, um nome de usuário fictício "Usuário Exemplo" é usado).

#### **Funções:**

1. **inicializar\_dados()**:
   * Cria os arquivos livros.csv e reservas.csv se eles ainda não existirem, com os cabeçalhos apropriados.
2. **carregar\_livros() e carregar\_reservas()**:
   * Carregam os livros e as reservas dos arquivos CSV para que possam ser exibidos e manipulados no sistema.
3. **cadastrar\_livro()**:
   * Permite que novos livros sejam adicionados ao acervo.
   * Os livros cadastrados ficam disponíveis para reserva.
4. **atualizar\_tabela()**:
   * Atualiza a tabela de livros exibida na interface gráfica com as informações dos livros cadastrados.
5. **pesquisar\_livros()**:
   * Permite pesquisar livros pelo título em tempo real, conforme o usuário digita.
6. **reservar\_livro()**:
   * Permite que o usuário reserve um livro disponível.
   * Após a reserva, o livro fica marcado como "Indisponível".
7. **remover\_reserva()**:
   * Remove a reserva de um livro, tornando-o novamente disponível para outros usuários.
8. **atualizar\_disponibilidade\_livro()**:
   * Atualiza o status de disponibilidade de um livro no arquivo CSV.
9. **exibir\_notificacao()**:
   * Exibe mensagens de notificação para o usuário, como confirmações de reservas ou erros.

### **Interface Gráfica:**

O código utiliza **Tkinter** para criar a interface gráfica. A interface possui:

* **Campos de texto** para inserir informações (nome, e-mail, título de livro, etc.).
* **Botões** para ações (login, cadastrar, reservar livro, remover reserva).
* Uma **tabela** para visualizar os livros cadastrados e suas disponibilidades.
* O sistema usa um **tema personalizado** (arc, breeze, clam, etc.) fornecido pela biblioteca **ttkthemes**, que dá uma aparência mais moderna ao programa.

### **Resumo:**

Este código cria um sistema básico de biblioteca com interface gráfica, onde os usuários podem se cadastrar, fazer login e reservar livros. Os livros são gerenciados em arquivos CSV, que simulam um banco de dados simples. O código é dividido em dois módulos principais: um para gerenciamento de usuários e outro para o gerenciamento de livros e reservas. A interface gráfica facilita a interação do usuário com o sistema.

Se algum campo necessário não for preenchido, ou se ocorrer algum erro, o sistema exibe mensagens amigáveis de erro e sucesso para guiar o usuário.