Documentação – Sistema de Biblioteca

Desenvolvido por: Isabella Moura

1. Introdução

- Este sistema foi desenvolvido em Django para gerenciar uma biblioteca acadêmica, incluindo funcionalidades de autenticação de usuários, operações de biblioteca (como cadastro de livros e empréstimos) e o cálculo de multas. Ele foi implantado no PythonAnywhere para acesso remoto.

2. Ferramentas Utilizadas

- **Python 3.10**: Linguagem principal para o desenvolvimento do projeto.
- **Django 5.1.2**: Framework para desenvolvimento web.
- **Git e GitHub**: Controle de versão do código.
- **PythonAnywhere**: Serviço de hospedagem para implantação do sistema.

3. Estrutura de Pastas e Arquivos

- **SistemaBiblioteca**: Diretório principal do projeto.
    - **sistema-biblioteca**: Subdiretório do projeto.
        - settings.py: Arquivo de configuração principal com as definições do projeto.
        - urls.py: Mapeamento de rotas para cada funcionalidade da aplicação.
        - wsgi.py: Arquivo de configuração para rodar a aplicação no servidor WSGI do PythonAnywhere.
    - **catalogo**: Aplicação interna que gerencia as funcionalidades de biblioteca.
        - models.py: Define as tabelas do banco de dados, como Usuario, Livro, Exemplar, Emprestimo, Reserva e Multa.
        - views.py: Define as funções de cada endpoint.
        - admin.py: Configurações do Django Admin que facilitam a gestão do administrador.

- templates: Diretório que contém os arquivos HTML de cada página do sistema.

4. Conexão do Django com o Banco de Dados

- No arquivo settings.py, o Django foi configurado para usar o banco de dados SQLite.
- **Migrações**: As migrações foram geradas com python manage.py makemigrations e aplicadas com python manage.py migrate (quando o banco é migrado, qualquer alteração no modelo deve ser migrada novamente, ou seja, quando a estrutura é modificada, deve-se apagar o banco com as configurações iniciais e migrar novamente com o arquivo salvo).

5. Funcionalidades e Estrutura dos Modelos

- **Modelo Usuario**: Define tipos de usuários e permissões específicas. Modelo implementado para suportar os diferentes perfis da biblioteca (aluno, professor, bibliotecário, administrador).
- **Modelo Livro e Exemplar**: Livro representa a obra e Exemplar armazena os dados do exemplar físico.
- **Modelo Emprestimo e Reserva**: Controlam operações de empréstimo e reserva, com validações para verificar se o usuário está apto.
- **Modelo Multa**: Calcula multas automaticamente em devoluções atrasadas, usando um sinal (post_save) (lógica ainda em construção).

6. Implementação de Funcionalidades por Perfil

- **Usuário Externo**: Apenas consulta o catálogo.
- **Aluno e Professor**: Consultas, reservas e visualização de multas.
- **Bibliotecário**: Cadastra e lista livros, realiza empréstimos, devoluções, aplica multas e marca exemplares como indisponíveis.
- **Administrador**: Todas as funcionalidades, além de gerenciar usuários.

7. Configuração no Github e Deploy no PythonAnywhere

7.1 Configuração no PythonAnywhere

- Depois de configurar todas as alterações em views, models e templates, o projeto foi versionado e pushado para o GitHub com:

  git init

  git add .

git commit -m "Initial commit"

git remote add origin https://github.com/isabellasmou/sistema-biblioteca.git

git push -u origin master

7.3 Implantação no PythonAnywhere

- **Clonagem do Projeto**: O repositório foi clonado no PythonAnywhere com:

  git clone https://github.com/isabellasmou/sistema-biblioteca.git

- **Configuração do Ambiente Virtual**: Foi criado e ativado um ambiente virtual com as dependências do projeto:

  python3.10 -m venv venv

  source venv/bin/activate

  pip install -r sistema-biblioteca/requirements.txt

- **Configuração do WSGI**: No arquivo /var/www/isabellasmou_pythonanywhere_com_wsgi.py, o projeto foi configurado para o servidor WSGI do PythonAnywhere, apontando para o diretório do projeto e configurando o ambiente virtual:

```
import os
import sys

project_path = '/home/isabellasmou/SistemaBiblioteca/sistema-biblioteca'
if project_path not in sys.path:
    sys.path.append(project_path)

virtualenv_path = '/home/isabellasmou/SistemaBiblioteca/sistema-biblioteca/venv'
activate_env = os.path.join(virtualenv_path, 'bin/activate_this.py')
os.environ['VIRTUAL_ENV'] = virtualenv_path
os.environ['PATH'] = f"{virtualenv_path}/bin:" + os.environ['PATH']

os.environ['DJANGO_SETTINGS_MODULE'] = 'biblioteca.settings'

from django.core.wsgi import get_wsgi_application
application = get_wsgi_application()
```

- **Migrações e Coleta de Arquivos Estáticos**:
    - As migrações também precisam ser aplicadas no PythonAnywhere, então foi aplicada com python manage.py migrate.
    - Os arquivos estáticos foram coletados com python manage.py collectstatic.

8. Funcionalidades de Backend

- **Autenticação**: Implementada com django.contrib.auth, com cada tipo de usuário mapeado a um grupo específico.
- **Controle de Acesso**: O login_required protege páginas específicas, e verificações adicionais garantem que apenas usuários autorizados realizem determinadas operações.
- **Sistema de Empréstimo e Reserva**: Funções específicas para gerenciar o limite de empréstimos e disponibilidade de exemplares, além de lógica de verificação de multas para usuários em atraso.
- **Multas e Penalidades**: Quando um livro é devolvido com atraso, uma multa é gerada automaticamente e aplicada ao usuário.

9. Requisições e Endpoints Importantes

- **Cadastro e Login**:
    - /catalogo/login/: Permite login.
    - /catalogo/cadastrar_usuario/: Cadastro de novos usuários.
- **Operações de Biblioteca**:
    - /catalogo/listar_livros/: Exibe lista de livros para consulta.
    - /catalogo/emprestar_livro/: Realiza empréstimos, com verificação de permissões.
    - /catalogo/devolver_livro/: Processa devoluções.
    - /catalogo/reservar_livro/: Permite reservas.
    - /catalogo/visualizar_multas/: Mostra as multas pendentes.

10. Templates

- **HTML**: Cada funcionalidade do sistema possui uma página HTML dedicada, com layouts organizados para consulta de livros, reservas e visualização de multas. Optei por não utilizar CSS para não precisar de mais ferramentas externas.