

Manual Técnico - Sistema de Biblioteca Digital

1. Visão Geral do Sistema

O sistema permite a gestão de documentos digitais (PDF, EPUB, MOBI, AZW, DOCX) em uma biblioteca universitária.

Funcionalidades:

- Listar documentos por tipo e ano de publicação.
- Adicionar, renomear e remover documentos.
- Testes automatizados para validação das funcionalidades.
- Integração com Git/GitHub.
- Suporte a execução no Google Colab.

2. Estrutura de Diretórios

```
projetobiblioteca/  
├─ library_manager/  
│   ├─ core.py  
│   └─ cli.py  
├─ scripts/  
│   └─ generate_dummy_docs.py  
├─ tests/  
│   └─ test_core.py  
├─ newfolder/  
└─ notebook.ipynb
```

3. Funcionalidades do Core

- find_documents(root): Lista todos os arquivos suportados.
- extract_year(path): Extrai ano de publicação/metadado.
- list_by_type_and_year(root): Organiza arquivos por tipo e ano.
- add_document(src_path, dest_dir): Copia arquivo para destino.
- rename_document(path, new_name): Renomeia arquivo.
- remove_document(path): Remove arquivo do sistema.

4. Interface de Linha de Comando (CLI)

Comando	Descrição	Exemplo
---------	-----------	---------

```
|-----|-----|-----|
| list <root> | Lista arquivos por tipo e ano | python -m library_manager.cli list
sample_docs |
| add <src> <dest> | Adiciona arquivo | python -m library_manager.cli add sample.pdf
sample_store/ |
| rename <path> <new_name> | Renomeia arquivo | python -m library_manager.cli
rename sample_store/sample.pdf novo.pdf |
| remove <path> | Remove arquivo | python -m library_manager.cli remove
sample_store/novo.pdf |
```

5. Scripts de Apoio

scripts/generate_dummy_docs.py:

- Gera arquivos de teste aleatórios (PDF, EPUB, DOCX, MOBI, AZW).
- Cria diretório sample_docs/.
- Uso: python scripts/generate_dummy_docs.py

6. Testes Automatizados

- Valida funções: add_document, rename_document, remove_document, list_by_type_and_year.

- Rodar no Colab:

```
import sys
from pathlib import Path
sys.path.append(str(Path.cwd()))
!pytest -q
```

7. Integração com Git/GitHub

- Branch, Commit e Push:

```
!git checkout -b feature/colab_setup
!git add .
!git commit -m "feat: inicializar projeto"
!git push origin feature/colab_setup
```

- Criar Pull Request:

```
!gh pr create --title "Feature: Configuração Colab + CLI + testes" --base main --head
feature/colab_setup --reviewer dallacortejr
```

8. Manual de Execução no Google Colab

1. Clone o repositório:

```
!git clone https://github.com/dallacortejr/projetobiblioteca.git
```

```
%cd projetobiblioteca
```

2. Instale dependências:

```
!pip install PyPDF2 python-docx ebooklib fpdf click pytest python-dateutil
```

```
!sudo apt-get install gh -y
```

3. Execute o notebook passo a passo: geração de arquivos, testes, CLI, commit e push.

9. Boas Práticas

- Use pastas separadas (sample_docs, sample_store, newfolder) para não sobrescrever arquivos.
- Execute testes antes de commitar.
- Para adicionar novos arquivos, use add_document ou add no CLI.
- Para remover ou renomear, use funções core ou CLI correspondentes.

10. Contato

Desenvolvedor: Luiz Dallacorte Jr

E-mail: dallacortejr@gmail.com

Repositório: <https://github.com/dallacortejr/projetobiblioteca>