**JOÃO VICTOR SANTOS  
LARA HELLEN MARQUES  
FÁBIO SANTOS  
GABRIELLE ANGELINA  
DANIEL COSTA**

**DOCUMENTAÇÃO TÉCNICA DA PLATAFORMALIVROS  
APLICAÇÃO STREAMLIT COM REDE NEURAL**

SALVADOR  
2025

JOÃO VICTOR SANTOS  
LARA HELLEN MARQUES  
FÁBIO SANTOS  
GABRIELLE ANGELINA  
DANIEL COSTA

DOCUMENTAÇÃO TÉCNICA DA PLATAFORMALIVROS:  
APLICAÇÃO STREAMLIT COM REDE NEURAL

Trabalho apresentado à disciplina Tópicos de Big Data em Python, da turma ARA0168, sob orientação do professor Roney Malaguti.

SALVADOR  
2025

**SUMÁRIO**

1. Introdução

2. Referencial Teórico

3. Arquitetura de Software

4. Descrição de Módulos

5. Instalação e Configuração

6. Uso da Aplicação

7. Testes e Validação

8. Manutenção e Extensões

9. Conclusão

Referências

Apêndice A – Código-fonte completo

**1. INTRODUÇÃO**

Este documento descreve o projeto PlataformaLivros, desenvolvido na disciplina Tópicos de Big Data em Python, da turma ARA0168, sob orientação do professor Roney Malaguti. A proposta consiste em uma aplicação web interativa, construída com Streamlit, com foco em recomendar livros personalizados aos usuários, ao mesmo tempo em que fornece dados sobre suas preferências aos revendedores.

O Brasil enfrenta desafios históricos relacionados à leitura. Segundo a pesquisa Retratos da Leitura no Brasil (Instituto Pró-Livro, 2020), o país perdeu cerca de 4,6 milhões de leitores entre 2015 e 2019, e apenas 52% da população se declara leitora. Entre os principais motivos para não ler estão a falta de tempo (47%) e a falta de interesse (32%).

A internet, ao mesmo tempo em que compete com os livros pela atenção dos jovens, também oferece oportunidades para o engajamento e a difusão da leitura. Plataformas como Kindle, Wattpad e Google Books já demonstraram como a tecnologia pode ser uma aliada nesse processo.

A PlataformaLivros aproveita esse potencial, utilizando dados e uma rede neural para oferecer recomendações personalizadas. Com uma interface simples e amigável, promove uma experiência de leitura mais alinhada à realidade dos jovens brasileiros, estimulando o hábito da leitura por meio da personalização, facilidade de acesso e sugestões relevantes.

**2. REFERENCIAL TEÓRICO**

Sistemas de recomendação são conjuntos de técnicas que visam prever as preferências dos usuários com base em seu histórico de interações. Segundo Ricci et al. (2015), esses sistemas são comumente classificados em:

* Baseados em conteúdo (content-based);
* Colaborativos (collaborative filtering);
* Híbridos.

A inteligência artificial e o aprendizado de máquina (machine learning) constituem a base de modelos mais eficientes, sendo as redes neurais profundas (deep learning) uma abordagem moderna e promissora.

A biblioteca Streamlit é uma ferramenta de desenvolvimento front-end em Python voltada para cientistas de dados. Ela permite a criação de dashboards e interfaces web sem a complexidade de frameworks tradicionais. Sua sintaxe simples e a integração com bibliotecas como pandas, numpy e sklearn tornam o desenvolvimento mais ágil.

Segundo a documentação oficial do Streamlit (2024), seu objetivo é “transformar scripts de dados em aplicações web com apenas algumas linhas de código”.

**3. ARQUITETURA DE SOFTWARE**

A estrutura do projeto se baseia em três camadas:

* **Interface:** Construída com Streamlit, permite a interação direta do usuário.
* **Dados:** Manipulação de arquivos CSV usando pandas.
* **Recomendação:** Lógica simples baseada em leitura dos dados de entrada.

Essa arquitetura modular, torna o sistema flexível para manutenção e futuras expansões.

**4. DESCRIÇÃO DE MÓDULOS**

A seguir, estão descritos os arquivos e diretórios que compõem o projeto PlataformaLivros, organizados por funcionalidade:

**4.1. streamlit\_app.py**

Arquivo principal da aplicação. Implementa a interface gráfica utilizando a biblioteca Streamlit. Suas funções principais incluem:

* Configurar a página inicial com “st.set\_page\_config()”;
* Exibir o menu de navegação lateral com as opções "Início", "Recomendações" e "Perfil";
* Carregar os dados dos livros e apresentá-los ao usuário de forma interativa;
* Utilizar pandas para leitura e manipulação dos dados em formato CSV.

**4.2. streamlit\_app\_login\_cadastro.py**

Versão estendida da aplicação, com funcionalidades de autenticação, como login e cadastro de usuários. Utiliza Streamlit e SQLite para armazenamento de usuários, permitindo um controle básico de acesso.

**4.3. gamificacao.py**

Este módulo implementa elementos de gamificação com o objetivo de aumentar o engajamento do usuário. Suas funções incluem:

* Atribuição de pontos por atividades dentro da plataforma;
* Criação de níveis ou conquistas com base no desempenho do usuário;
* Integração futura com a interface de perfil para exibição de progresso.

**4.4. mover\_logo.py**

Script auxiliar voltado para a manipulação visual do logotipo da aplicação. Pode ser utilizado para definir o posicionamento, redimensionamento ou carregamento do logotipo a partir do diretório “assets/”.

**4.5. requirements.txt**

Lista todas as dependências utilizadas no projeto. Entre as bibliotecas principais estão:

* **streamlit:** Para criação da interface web;
* **pandas:** Para tratamento de dados;
* **sqlite3 (via Python nativo):** Para armazenamento de dados de login.

Esse arquivo permite instalar todas as dependências com um único comando:

pip install -r requirements.txt

**4.6. .gitignore**

Define os arquivos e pastas que devem ser ignorados pelo controle de versão do Git. Evita o versionamento de arquivos temporários, senhas, bancos de dados locais e outros arquivos desnecessários.

**4.7. LICENSE**

Arquivo de licença do projeto. A PlataformaLivros utiliza a licença Apache-2.0, que permite uso, modificação e redistribuição do código com poucas restrições, desde que os termos da licença sejam mantidos.

**4.8. README.md**

Arquivo de documentação inicial do projeto. Contém informações como:

* Resumo do projeto;
* Tecnologias utilizadas;
* Instruções básicas de instalação e execução;
* Créditos e informações de contato.

**4.9. Diretório .streamlit/**

Contém arquivos de configuração da aplicação Streamlit. O mais comum é o config.toml, que permite personalizar o tema da aplicação (cores, fonte, layout) e definir opções como a porta padrão ou título do navegador.

Estes elementos são utilizados para enriquecer a experiência visual da plataforma.

**4.10. Diretório static/**

Semelhante ao diretório “assets/”, armazena recursos estáticos adicionais que podem incluir:

* Arquivos CSS;
* Scripts JavaScript;
* Arquivos auxiliares de design ou interatividade.

Esse diretório pode ser utilizado futuramente para aprimorar o visual da interface por meio de customizações externas.

**5. INSTALAÇÃO E CONFIGURAÇÃO**

Para executar o projeto PlataformaLivros localmente, siga os passos abaixo:  
  
1. Clone o repositório: git clone <https://github.com/helleenlara/Plataformalivros.git>  
2. Acesse a pasta clonada: cd Plataformalivros  
3. Instale as dependências: pip install -r requirements.txt  
4. Execute a aplicação: streamlit run streamlit\_app.py  
  
Certifique-se de ter o Python 3.10+ instalado, bem como as bibliotecas Streamlit e Pandas.

**6. USO DA APLICAÇÃO**

Após executar o comando `streamlit run streamlit\_app.py`, será aberta uma página no navegador com a interface da plataforma.  
  
A aplicação contém três páginas acessíveis pelo menu lateral:

**Início:** Introdução e objetivos do sistema;

**Recomendações:** Lista de livros disponíveis com exibição do título e autor;

**Perfil:** Espaço reservado para futuras funcionalidades, como login, gamificação e progresso de leitura.  
  
Na versão login\_cadastro, o usuário também pode criar contas e autenticar-se.

**7. TESTES E VALIDAÇÃO**

O sistema foi testado manualmente em ambiente local. Verificou-se:  
  
- Execução correta da aplicação sem erros de carregamento;  
- Navegação entre as páginas funcionando normalmente;  
- Exibição correta dos dados dos livros;  
- Compatibilidade com diferentes navegadores (Chrome, Edge).  
  
Testes automatizados ainda não foram implementados.

**8. MANUTENÇÃO E EXTENSÕES**

Melhorias planejadas para versões futuras incluem:  
  
- Sistema de avaliações (notas, comentários);  
- Ranking de leitores e pontuação por leitura (gamificação avançada);  
- Integração com APIs públicas como Google Books ou Open Library.

**9. CONCLUSÃO**

A PlataformaLivros representa uma aplicação prática dos conceitos de Big Data, aprendizado de máquina e desenvolvimento web. Utilizando ferramentas acessíveis como Python e Streamlit, foi possível construir uma interface funcional, voltada à promoção da leitura.

O projeto também aborda um problema social relevante: o baixo índice de leitura no Brasil. Ao combinar personalização, usabilidade e interatividade, a aplicação se apresenta como uma alternativa viável para estimular o hábito da leitura, especialmente entre os jovens conectados à internet.

**REFERÊNCIAS**

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 6023: Informação e documentação – Referências – Elaboração. Rio de Janeiro, 2018.  
INSTITUTO PRÓ-LIVRO. Retratos da Leitura no Brasil. 5ª edição. São Paulo, 2020. Disponível em: <https://prolivro.org.br/relatorios/>  
RICCI, F.; ROKACH, L.; SHAPIRA, B. Recommender Systems Handbook. Springer, 2015.  
STREAMLIT. Documentação oficial. Disponível em: <https://docs.streamlit.io/>

**APÊNDICE A – CÓDIGO-FONTE COMPLETO**

```python  
import streamlit as st  
import pandas as pd  
  
st.set\_page\_config(page\_title='Plataforma de Recomendação de Livros', layout='wide')  
st.title('Plataforma de Recomendação de Livros')  
  
menu = ['Início', 'Recomendações', 'Perfil']  
opcao = st.sidebar.selectbox('Menu', menu)  
  
if opcao == 'Início':  
 st.subheader('Bem-vindo à Plataforma de Recomendação de Livros!')  
 st.write('Explore livros recomendados com base em seus interesses.')  
  
elif opcao == 'Recomendações':  
 st.subheader('Livros Recomendados')  
 livros = pd.read\_csv('dados/livros.csv')  
 for i, row in livros.iterrows():  
 st.markdown(f"\*\*{row['titulo']}\*\* - {row['autor']}")  
  
elif opcao == 'Perfil':  
 st.subheader('Seu Perfil')  
 st.text('Funcionalidade de login ainda em desenvolvimento.')  
```