

RELATÓRIO TÉCNICO – PROJETO FINAL (AED I)

Disciplina: Algoritmos e Estruturas de Dados I – COML0302

Professor: Dr. João da Mata Libório Filho

Período: 2025/2

Equipe: Carolina Dos S. Leão, Danielly S. Ferreira, Gabriele G. Martins, Grazielle C. de Gouveia, Higo da S. Menezes, Maria Gabrielle L. de Oliveira, Mizzia Eduarda T. de Mello, Luiz Roberto P. Pinto, Rosielly S. Ferreira, Samanta R. da Fonseca.

Projeto Acervo+: Um Sistema de Gerenciamento para a Biblioteca do NESNAP/UEA

1. Descrição do Problema ou Contexto da Aplicação

A Biblioteca do Núcleo de ensino Superior de Novo Aripuanã, da Universidade do Estado do Amazonas (UEA), enfrenta o desafio de gerenciar de forma eficiente seu acervo de livros físicos, incluindo empréstimos e devoluções. As principais dificuldades estão na gestão do acervo físico, que ainda é realizada manualmente, registros em papel e muitas vezes nem são registrados os empréstimos de livros, o que torna o processo propenso a erros, atrasos e até extravio de livros. Além disso, a biblioteca encontra-se parcialmente abandonada, necessitando de organização e manutenção, o que compromete o acesso e o gerenciamento adequado dos exemplares. Este projeto surge como uma iniciativa para revitalizar a biblioteca, tornando-a mais funcional, organizada e acessível para usuários e funcionários, propondo uma solução digital para modernizar, organizar e facilitar o acesso aos exemplares, bem como a revitalização do espaço.

O desenvolvimento desse projeto é justificado pensando em solucionar esse problema mediante um sistema computacional interativo, que permite ao bibliotecário registrar, editar e monitorar empréstimos e devoluções dos livros de forma automatizada. A aplicação foi implementada em Python, utilizando a biblioteca *Streamlit* para criar uma interface gráfica simples, moderna e funcional.

2. Descrição Geral da Aplicação

O Sistema de Gerenciamento da Biblioteca do NESNAP/UEA é uma aplicação desenvolvida em Python, utilizando o framework *Streamlit* para criação da interface gráfica. O objetivo do sistema é automatizar o controle de empréstimos e devoluções de livros, além de facilitar o gerenciamento do acervo físico da biblioteca.



2.1 Principais Funcionalidades

O sistema permite:

- Cadastrar novos livros, informando título, autor, gênero e quantidade total;
- Visualizar e buscar livros disponíveis no acervo, com filtros por título, autor e gênero;
- Realizar empréstimos e devoluções de exemplares, controlando automaticamente os contadores de disponíveis e emprestados;
- Editar ou remover livros do sistema;
- Visualizar estatísticas gerais, como total de exemplares, livros disponíveis e mais emprestados.

2.2 Operações CRUD Disponíveis

O sistema implementa todas as operações CRUD (Create, Read, Update, Delete):

- Create (Criar): cadastro de novos livros.
- Read (Ler): visualização da lista de livros e busca com filtros.
- Update (Atualizar): edição de informações de livros já cadastrados (título, autor, gênero e quantidade).
- Delete (Remover): remoção de livros da biblioteca.

Essas operações são realizadas diretamente pela interface interativa, sem necessidade de comandos manuais de programação.

Todas as informações são armazenadas em estruturas de dados temporárias (em memória), mas o sistema pode ser facilmente expandido para integração com bancos de dados no futuro.

3. TAD Utilizado e Justificativa da Escolha

Foram utilizados os seguintes **Tipos Abstratos de Dados (TADs)**:

- **Lista (list)** – Lista Sequencial (Estática), a principal estrutura usada para armazenar os livros. Cada livro é representado como um dicionário contendo seus atributos (título, autor, gênero, quantidade total, emprestados, disponíveis e status). A lista permite inserção, remoção e acesso direto aos elementos, sendo ideal para coleções pequenas e médias, de crescimento dinâmico.
- **Fila (collections.deque)** – utilizada para controlar os empréstimos ativos. A escolha da fila se deve ao comportamento FIFO (First In, First Out), que reflete a ordem cronológica dos empréstimos feitos pelo bibliotecário.
- **Pilha (list)** – usada para registrar as devoluções, seguindo a lógica LIFO (Last In, First Out), o que representa bem o histórico mais recente de devoluções.

Esses TADs foram escolhidos por serem simples, eficientes e perfeitamente adequados à natureza da aplicação. Como o foco está no aprendizado de Estruturas de Dados, o uso de listas, pilhas e filas demonstra claramente os conceitos fundamentais de armazenamento e manipulação de dados dinâmicos.

4. Algoritmos de Ordenação Utilizados

O algoritmo de ordenação escolhido foi o Bubble Sort, por sua simplicidade e caráter didático, ele é usado para ordenar os livros com base em diferentes critérios (título, autor ou gênero). Apesar de não ser o mais eficiente em termos de desempenho para grandes volumes de dados, ele é amplamente utilizado em contextos educacionais, pois demonstra claramente o processo de comparação e troca de elementos adjacentes.

- I. Onde: Na função `bubble_sort()` e no botão “Ordenar”.
- II. Pra quê: Pra organizar a tabela de livros por título, autor ou gênero.
- III. Por que é importante: Deixa a visualização mais clara e facilita encontrar livros — além de demonstrar, na prática, o uso de um algoritmo de ordenação clássico dentro de um projeto real.

4.1 Complexidade:

- a) Melhor caso: $O(n)$ (quando já está ordenado)
- b) Pior caso: $O(n^2)$

A aplicação tem um número reduzido de livros, o que torna o custo computacional aceitável. Além disso, o objetivo principal é demonstrar a implementação prática do algoritmo em um sistema real, reforçando o aprendizado teórico da disciplina.

5. Decisões de Projeto e Arquitetura do Sistema

O sistema foi desenvolvido em Python, adotando uma arquitetura simples e organizada em blocos lógicos, seguindo uma arquitetura modular, com a camada de manipulação de dados implementada na pasta TADs e a camada de interface concentrada no arquivo `app.py`, garantindo separação de responsabilidades entre lógica e interface:

- **Camada de Interface – App.py (Frontend):** Desenvolvida com a biblioteca **Streamlit**, que oferece uma interface amigável e interativa. Essa camada é responsável pela exibição das tabelas, formulários e botões.
- **Camada Lógica - Tads/biblioteca.py (Backend):** Contém as funções de manipulação dos dados: adicionar, editar, remover, emprestar e devolver livros. Toda

a lógica de atualização dos status e contadores foi centralizada em funções reutilizáveis, como `atualizar_status()` e `atualizar_contadores_gerais()`.

- **Camada de Dados (Estruturas TAD):** As informações são mantidas em listas e dicionários, armazenados em sessão (`st.session_state`) para persistirem enquanto o aplicativo está aberto.

Essa estrutura modular permite fácil manutenção, leitura e expansão futura do sistema.

6. Limitações e Próximos Passos

Apesar de funcional e visualmente agradável, o sistema ainda apresenta algumas **limitações**:

- Os dados são armazenados apenas em memória temporária, ou seja, são perdidos ao encerrar a sessão.
- Não há autenticação de usuários (qualquer pessoa que acessar pode usar o sistema).
- O histórico de movimentações ainda é básico (não registra datas ou horários dos empréstimos e devoluções).

Próximos passos:

1. Integrar o sistema a um banco de dados SQLite ou Firebase para persistência de dados.
2. Implementar login e controle de acesso para o bibliotecário.
3. Adicionar histórico completo de movimentações, com data e hora de cada ação.
4. Permitir empréstimos múltiplos por usuário e geração de relatórios em PDF.
5. Melhorar o design visual e a usabilidade da interface.

Considerações Finais

O Sistema de Gerenciamento da Biblioteca da UEA, desenvolvido no Núcleo de Ensino Superior de Novo Aripuanã, surgiu a partir da necessidade de solucionar problemas enfrentados na gestão manual de livros na localidade, como dificuldades de controle, lentidão nas atualizações e desorganização do acervo. O sistema demonstra, na prática, a aplicação dos conceitos de TADs e algoritmos de ordenação no desenvolvimento de uma solução real e funcional. Além de automatizar o trabalho do bibliotecário, o projeto reforça a importância de uma boa modelagem de dados, do uso adequado das estruturas computacionais e da organização lógica dos processos, contribuindo tanto para a eficiência da biblioteca quanto para o aprendizado prático dos estudantes ao conectar teoria e realidade local.

Apêndice



Figura 1 Codigos em camadas --> Tads/ - app.py

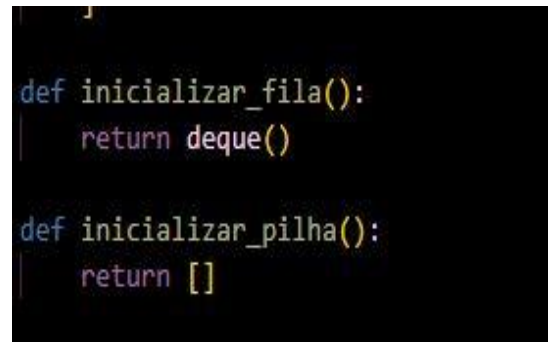


Figura 2. TADs utilizados

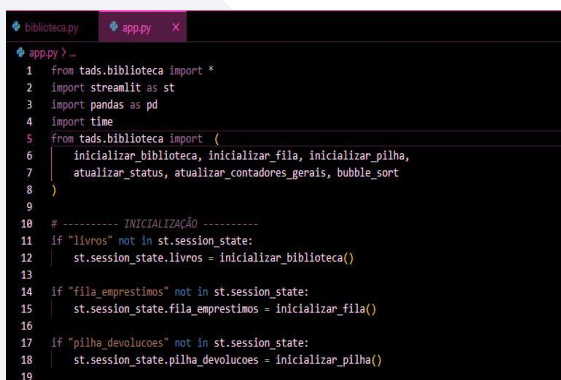


Figura 3. Importação da Tads/biblioteca

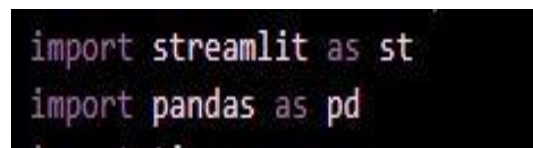


Figura 4. Import do Streamlit

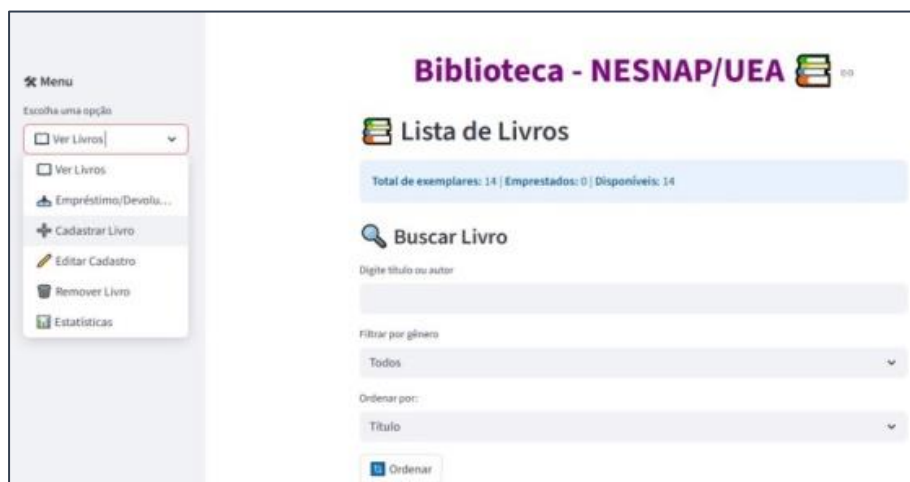


Figura 5. Interface do sistema no Streamlit