

Ano/Semestre	Professor(a)
2025/2	Dr. Joao da Mata Libório Filho
Período	Disciplina
3º	Algoritmos e Estruturas de Dados I - Coml0302
Acadêmicos: 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.	

Relatório Técnico – 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**. Atualmente, o controle é realizado de forma manual, por meio de planilhas ou registros em papel, o que torna o processo **propenso a erros, atrasos e dificuldades na atualização dos dados**. 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.

Para resolver esse problema, foi desenvolvido um **sistema computacional interativo**, que permite ao **bibliotecário** registrar, editar e monitorar todo o ciclo de vida 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 e Principais Funcionalidades

O **Sistema de Gerenciamento da Biblioteca da UEA** é uma aplicação web simples, voltada para o uso interno do bibliotecário. Através dela, é possível executar as seguintes funções:

- **Visualizar todos os livros** cadastrados, com seus respectivos status, autores, gêneros e quantidades.
- **Pesquisar livros** por título, autor ou gênero, com retorno dinâmico e mensagens caso não haja resultados.
- **Cadastrar novos livros**, informando título, autor, gênero e quantidade total de exemplares.

- **Editar livros existentes**, alterando título, autor, gênero ou número de exemplares disponíveis.
- **Remover livros** que foram cadastrados incorretamente ou que saíram do acervo.
- **Registrar empréstimos**, associando o livro a um aluno.
- **Registrar devoluções**, atualizando o status e o número de exemplares disponíveis.
- **Visualizar estatísticas** com gráficos dos livros mais emprestados e contadores gerais do acervo.

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)** – 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.

Justificativa geral: 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 e Justificativa

O algoritmo de ordenação escolhido foi o **Bubble Sort**, por sua **simplicidade e caráter didático**.

O Bubble Sort 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.

Complexidade:

- Melhor caso: $O(n)$ (quando já está ordenado)

- Pior caso: $O(n^2)$

Justificativa:

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:

- **Camada de Interface (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 (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 planejados:

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.

Conclusão

O Sistema de Gerenciamento da Biblioteca da UEA cumpre o papel de demonstrar, na prática, a aplicação dos conceitos de **TADs e algoritmos de ordenação** no contexto de uma solução real e útil. Além de automatizar o trabalho do bibliotecário, o projeto reforça a importância de uma boa modelagem de dados e do uso de estruturas adequadas para cada tipo de problema computacional.