

# Trabalho Avaliativo - Sistema de Gerenciamento de Biblioteca

**Curso:** Análise e Desenvolvimento de Sistemas - ADS

**Disciplina:** Lógica de Programação Algorítmica

**Professor:** Francisco Coelho

**Data:** 10 de junho de 2025

## Integrantes:

- Kauê Victor
  - Sara Beatriz
  - Tainá Miranda
  - Guilherme Nascimento
  - Matheus Pacífico
- 

## 1. Introdução

Este documento detalha o desenvolvimento de um trabalho que pede um sistema de gerenciamento para uma pequena biblioteca, como parte da segunda avaliação da disciplina. O objetivo do projeto foi criar uma aplicação em Python capaz de registrar e gerenciar as principais operações de uma biblioteca, como o controle de livros, usuários e empréstimos.

Para garantir que as informações não sejam perdidas ao fechar o sistema, todos os dados são armazenados de forma organizada em arquivos no formato JSON, facilitando consultas e futuras manipulações.

## 2. Estrutura e Organização do Código

Para facilitar o desenvolvimento em equipe e a manutenção do código, optamos por modularizar o projeto em arquivos distintos, onde cada um possui uma responsabilidade clara:

- **biblioteca.py (Arquivo Principal):** É o coração do programa. Ele controla o menu, gerencia as listas de dados principais e chama as funções dos outros módulos para executar as tarefas solicitadas pelo usuário.
- **livros\_pacifico.py:** Contém todas as funções relacionadas ao gerenciamento de livros, como adicionar, listar e buscar.
- **usuarios\_sara.py:** Agrupa as funções para gerenciar os usuários, como o cadastro, listagem e alteração de dados.
- **emprestimos\_taina.py:** Responsável por toda a lógica de negócio que envolve os empréstimos e as devoluções de livros.
- **menu\_gui.py:** Um módulo simples, cuja única função é exibir o menu principal de opções ao usuário.

Essa separação tornou o código mais limpo e permitiu que cada integrante do grupo focasse em uma parte específica do sistema.

## 3. Gerenciamento de Dados

Seguindo a especificação do trabalho, os dados do sistema são armazenados em listas de dicionários, que ficam carregadas na memória durante a execução do programa.

- **catalogo\_livros:** Uma lista que armazena um dicionário para cada livro.
- **cadastro\_usuarios:** Lista que guarda os dicionários de todos os usuários cadastrados.
- **registros\_emprestimos:** Lista para registrar cada operação de empréstimo realizada.

Para a persistência dos dados, utilizamos três arquivos: `livros.json`, `usuarios.json` e `emprestimos.json`. No início do programa, o `main.py` lê esses arquivos para carregar os dados nas listas. Toda vez que uma alteração é feita (como um novo cadastro ou um empréstimo), a lista correspondente é salva de volta no arquivo JSON. A leitura dos arquivos é feita dentro de um bloco `try/except` para que o programa não apresente erro caso os arquivos ainda não existam, simplesmente iniciando com listas vazias.

#### 4. Descrição das Funções Implementadas

Foram implementadas todas as funções solicitadas, divididas por responsabilidade:

##### Funções de Gerenciamento de Livros

- **adicionar\_livro():** Pede ao usuário os dados do livro, gera um novo ID sequencial e o adiciona à lista `catalogo_livros`.
- **listar\_livros():** Percorre a lista de livros e exibe na tela os dados de cada um, incluindo seu status de disponibilidade ("Disponível" or "Emprestado").
- **buscar\_livro\_por\_titulo():** Permite que o usuário digite parte do nome de um livro e exibe todos os livros que correspondem à busca.
- **atualizar\_disponibilidade\_livro():** Altera o status de um livro (de disponível para indisponível, e vice-versa) através do seu ID.

##### Funções de Gerenciamento de Usuários

- **adicionar\_usuario():** Cadastra um novo usuário. A função gera um ID único e, antes de salvar, verifica se o e-mail inserido já não pertence a outro usuário, garantindo que cada e-mail seja único no sistema.
- **listar\_usuarios():** Exibe uma lista com os dados de todos os usuários cadastrados.
- **buscar\_usuario\_por\_id():** Procura e exibe as informações de um usuário específico a partir do seu ID.
- **altera\_email():** Permite a modificação do e-mail de um usuário já cadastrado.

##### Funções de Gerenciamento de Empréstimos

- **realizar\_emprestimo():** Esta função registra um novo empréstimo. Antes de concluir, ela realiza validações importantes: verifica se o ID do livro e do usuário são válidos e se o livro está realmente disponível. Se tudo estiver correto, ela cria o registro do empréstimo e atualiza a disponibilidade do livro para `False`.
- **devolver\_livro():** Encontra um empréstimo pelo seu ID, preenche a data de devolução e, crucialmente, atualiza a disponibilidade do livro correspondente para `True`, deixando-o livre para um novo empréstimo.
- **listar\_emprestimos\_ativos():** Mostra uma lista de todos os empréstimos que ainda não foram devolvidos, exibindo o título do livro e o nome do usuário que o pegou.
- **historico\_emprestimos\_usuario():** Exibe todos os empréstimos (tanto os ativos quanto os já devolvidos) de um usuário específico.

## **5. Considerações finais**

Fizemos o script com dedicação, fizemos o código se atentando a cada requisito, espero que o senhor tenha gostado, estamos sempre abertos a feedbacks, se tiver erros, não ficaremos tristes e sim aprenderemos com ele!