

RELATÓRIO DO PROJETO — Sistema de Biblioteca “Livro Aberto”

1. Introdução

Este relatório apresenta o desenvolvimento de um sistema completo de gerenciamento para a biblioteca fictícia “**Livro Aberto**”, implementado em Java com persistência em arquivos texto. O projeto foi construído com foco em organização, modularidade e atendimento aos requisitos estabelecidos no documento “S6a - Biblioteca”.

O objetivo do sistema é oferecer funcionalidades de controle de empréstimos, devoluções, cálculo de multas, cadastro de associados e livros, controle de livros raros, relatórios detalhados e sistema de pagamento.

2. Objetivos do Sistema

O sistema visa:

- Automatizar o processo de empréstimo e devolução de livros.
 - Controlar livros raros com acesso restrito a associados VIP.
 - Gerenciar prazos diferentes para associados regulares e VIP.
 - Aplicar multas conforme regras específicas.
 - Registrar pagamentos e manter histórico.
 - Emitir relatórios completos sobre uso e frequência.
 - Salvar dados em arquivos externos para manter as informações entre execuções.
-

3. Especificações Técnicas

3.1 Linguagem e Tecnologias Utilizadas

- Linguagem: **Java 17+**
- Persistência: **Arquivos .txt**
- Entrada e saída: **Scanner + PrintWriter + BufferedReader**
- Paradigma utilizado: **Programação orientada a objetos**
- Padrões:
 - Encapsulamento
 - Classes especializadas
 - Persistência simplificada
 - Modelagem separada por responsabilidade

3.2 Principais Classes Implementadas

- **Main:** responsável pelo menu, fluxo operacional e interação com o usuário.
 - **SistemaBiblioteca:** gerencia listas, salva e lê dados.
 - **Associado / AssociadoVIP:** modela os tipos de usuários.
 - **Livro / LivroRaro:** representa os livros cadastrados.
 - **Emprestimo:** detalha o processo de empréstimo, datas e multa.
 - **Pagamento:** registra multas pagas.
 - **Relatorio:** geração de relatórios detalhados.
-

4. Requisitos Atendidos

1. Cadastro de associados, livros e eventos literários

Associados e livros podem ser cadastrados pelo menu. Eventos podem ser adicionados caso necessário.

2. Empréstimo e devolução com cálculo de multas

O programa permite informar:

- Data do empréstimo
- Data da devolução

O sistema calcula automaticamente:

- Dias decorridos
- Prazo individual do associado
- Multa conforme atraso

3. Controle de acesso a livros raros e reservas

Livros raros só podem ser emprestados por associados **VIP**.

4. Programa de benefícios por tipo de associado

- **Regular:** prazo de 10 dias, sem desconto.
- **VIP:** prazo de 15 dias, 50% de desconto em multas.

5. Sistema de pagamento e relatórios detalhados

O sistema registra:

- Pagamento de multas
- Empréstimos realizados
- Devoluções
- Total de multas pagas
- Histórico por associado

Os relatórios exibem:

- Quantidade de empréstimos
 - Livros em aberto
 - Multas pagas
 - Datas de cada empréstimo
-

5. Arquitetura e Funcionamento

O sistema segue uma **arquitetura baseada em camadas de responsabilidade**:

5.1 Camada de Modelos

Define entidades reais:

- Livro, LivroRaro
- Associado, AssociadoVIP
- Emprestimo
- Pagamento

5.2 Camada de Controle

Responsável por:

- Validar acesso a livros raros
- Registrar empréstimos e devoluções
- Aplicar prazos diferenciados
- Calcular multas automaticamente

5.3 Camada de Persistência

Controlada pela classe `SistemaBiblioteca`:

- Salva dados em arquivos
- Carrega todos os registros ao iniciar
- Mantém histórico real do sistema

5.4 Camada de Interface

Classe `Main`:

- Menu baseado em texto
 - Entrada pelo Scanner
 - Gerenciamento do fluxo geral
-

6. Dificuldades Encontradas

Durante o desenvolvimento foram observadas várias dificuldades técnicas, incluindo:

6.1 Integração entre tipos de associados

A lógica de prazos diferentes exigiu ajustes nos métodos:

- construtor do empréstimo
- cálculo da multa
- registro da devolução

6.2 Persistência em arquivos

Foi necessário:

- Criar funções dedicadas de leitura e escrita
- Converter objetos para texto
- Reconstruí-los ao iniciar o sistema
- Garantir compatibilidade de formato

6.3 Localização correta do empréstimo ao devolver

Inicialmente, o sistema não sabia **qual associado** estava devolvendo o livro.
Alteramos o menu para solicitar o **nome do associado**, garantindo:

empréstimo pertence a (associado + livro + não devolvido)

6.4 Manter coerência do estado do livro

Livros deveriam alternar entre:

- Disponível
- Emprestado

Exigiu controle rigoroso nas funções de empréstimo e devolução.

7. Pontos Fortes

7.1 Arquitetura modular

Cada classe possui responsabilidade clara, facilitando expansão e manutenção.

7.2 Persistência funcional

Os dados não se perdem ao encerrar o programa.

7.3 Regras de negócio completas

Incluindo:

- Descontos VIP
- Prazos diferenciados
- Controle de livros raros

7.4 Relatórios detalhados

Mostram informações essenciais sobre:

- Uso
- Frequência
- Histórico
- Pagamentos

7.5 Fácil de expandir

A estrutura permite adicionar:

- Banco de dados
 - Interface gráfica
 - APIs REST
-

8. Pontos Fracos

8.1 Persistência limitada

O uso de arquivos .txt não é o ideal para:

- Grandes volumes de dados
- Concorrência
- Integridade transacional

Um banco de dados seria mais robusto.

8.2 Interface textual

A falta de GUI reduz:

- Experiência do usuário
- Usabilidade
- Intuitividade

8.3 Falta de exceções detalhadas

Tratamento de erros pode ser expandido para:

- Erros específicos
- Mensagens orientadas ao usuário

8.4 Sem testes automatizados

O projeto não inclui:

- Testes unitários
 - Testes de regressão
-

9. Conclusão

O sistema da biblioteca “Livro Aberto” atende plenamente os requisitos estipulados, apresentando uma implementação sólida em Java, organizada em classes funcionais e com persistência integrada. Apesar das limitações naturais de um sistema baseado em arquivos, ele cumpre o papel de forma eficaz e gera valor através de automação de processos e controle detalhado de informações.

O projeto demonstra domínio em programação orientada a objetos, modelagem de dados, e implementação de regras de negócio complexas..