

# Relatório de Produção Individual – Aplicação Desktop

**Aluno:** Daired Almeida Cruz

## 1. Atribuição de Cargo e Tarefas

Minha atribuição no projeto foi codificar a Aplicação Desktop. A priori, minhas responsabilidades incluíam a concepção e implementação completa do cliente desktop para o sistema de gestão da biblioteca.

Na prática, minhas tarefas foram:

- **Arquitetura do Software:** Definir e implementar a arquitetura MVC (Model-View-Controller) para garantir a separação de responsabilidades e a manutenibilidade do código.
- **Desenvolvimento da Interface:** Criar a interface gráfica (GUI) utilizando JavaFX, estruturando a navegação em abas para as diferentes funcionalidades do sistema.
- **Implementação da Persistência de Dados:** Configurar e gerenciar o banco de dados SQLite e implementar a camada de acesso a dados utilizando o framework ORMLite para o mapeamento objeto-relacional.
- **Desenvolvimento das Funcionalidades Core:** Implementar todas as operações de CRUD (Create, Read, Update, Delete) para as entidades principais do sistema, como Livro, Autor, Usuário, etc.
- **Qualidade e Documentação:** Garantir o funcionamento do código através de testes unitários com JUnit e elaborar a documentação técnica inicial do projeto com JavaDoc e o arquivo README.md.

## 2. Contribuição de Acordo com a Atribuição

Cumpri integralmente as tarefas que me foram atribuídas, entregando uma aplicação desktop funcional e bem estruturada.

- **O que foi cumprido:** Todos os tópicos da seção anterior.
- **Entregas mais relevantes:**
  - **README:** O arquivo README.md, que detalha a arquitetura, tecnologias e o guia de execução do projeto.
    - <https://github.com/SPD-BES-2025-3/grupo8/blob/main/biblioteca-desktop-app/README.md>
  - **Testes:** Implementação dos testes JUnit de todos os CRUDs testando todas as operações e atributos.
    - <https://github.com/SPD-BES-2025-3/grupo8/tree/main/biblioteca-desktop-app/src/test/java/model>
  - **JXML:** Criação das telas de interface de todos os CRUDs, apesar de parecidos, foi o que mais demandou tempo.
    - <https://github.com/SPD-BES-2025-3/grupo8/tree/main/biblioteca-desktop-app/src/main/resources/view>
- **Principais dificuldades:**

- A maior dificuldade foi criação das interfaces, era a parte que menos tinha domínio no Java, o restante por ter sempre trabalhado muito com Java tive uma boa agilidade. A criação inicial dos UMLs também foi um ponto de maior gasto de tempo, pois além da demora para formular as classes e operações, eu e o meu colega Hugo demoramos um pouco para chegar em um consenso do que ser implementado e colocado e o que poderia ser descartado para deixar o projeto mais enxuto.

### 3. Contribuição Além do Atribuído

- Dividimos o projeto em 2 parte o meu desktop com interface e o do Hugo com a API e integrador, então acabamos não ajudando na codificação um do outro. Houve maior contato durante a criação das bases do projeto, definição de funções e a própria modelagem UML. Provavelmente no futuro terei que ajudá-lo na criação do integrador para auxiliá-lo a entender os objetos orm e como manipulá-los nas conversões do Integrador.

### 4. Considerações Gerais

- **O que aprendi:**
  - **Técnicas:** Aprofundei meus conhecimentos em arquitetura MVC, no uso de frameworks como JavaFX e ORMLite, e na importância dos testes unitários (JUnit) para garantir a qualidade do software.
  - **Conceituais:** A implementação do repositório e do controlador abstrato melhorou meu entendimento sobre os princípios de reutilização de código, herança e programação genérica. Assim deixei a aplicação bem desacoplada para tentar ao maximo deixa-la mais facil de adicionar novas funcionalidades e a posteriori a integra-la com a aparte da API via o Integrador.
- **Trabalhos futuros pendentes:**
  - Sistema de Login e divisão de visão e funcionalidade entre diferentes tipos de Usuario.
  - Fazer a integração dessa minha implementação desktop com a parte da APIRest via Integrador.
  - Refinamento e ajustes das interfaces, para ajustar as operações de empréstimo e resenha junto as infos vindas da API.
- **Conclusão:** Concluo que a base da aplicação desktop foi estabelecida e por ter feito o uso de padrões de projeto e boas práticas, o código ficou bem desacoplado, fácil manutenção e com boa legibilidade/organização.