

Relatório de Produção Individual Final - Hugo Santos

Projeto: Sistema de Gerenciamento de Livraria (Versão Final Integrada)

Período: Julho de 2025

1. Visão Geral das Atribuições

Cargo Inicial: Desenvolvedor Backend.

Escopo Prático Exercido: Embora a responsabilidade primária tenha sido o desenvolvimento da API RESTful, o escopo do trabalho foi significativamente expandido para abranger a arquitetura de integração, o suporte ativo aos outros módulos (Desktop e Integrador) e a completa automação do ambiente de desenvolvimento e execução (DevOps). Na prática, atuei como um **Arquiteto de Soluções e Desenvolvedor Full-Stack da Infraestrutura**, garantindo não apenas que a API funcionasse, mas que todo o ecossistema fosse coeso, resiliente e fácil de operar.

2. Contribuições Detalhadas

2.1. Desenvolvimento do Backend (API RESTful)

Conforme o planejamento inicial, fui responsável por todo o ciclo de vida da API, entregando um serviço robusto e funcional.

- **Implementação do CRUD:** Desenvolvi todos os endpoints REST para as operações de Criar, Ler, Atualizar e Deletar (CRUD) para a entidade Livro.
- **Gerenciamento de Sub-recursos:** Criei o endpoint específico (POST /api/livros/{id}/resenhas) para gerenciar a adição de resenhas como um documento embutido, uma decisão de modelagem que otimizou a performance de leitura.
- **Testes Unitários:** Garanti a qualidade e a estabilidade da API através da implementação de testes unitários para as camadas de Controller e Repository, validando a lógica de negócio e o acesso aos dados.

2.2. Participação Ativa na Integração e Refatoração (Além do Atribuído)

A fase de integração exigiu uma colaboração intensa e a depuração de problemas complexos que iam além do escopo original da API.

- **Diagnóstico e Correção do Fluxo de Mensagens:** Identifiquei e corriji um bug crítico no fluxo **Desktop -> API**, onde os autores não eram sincronizados. A solução envolveu a refatoração do LivroController da aplicação Desktop para garantir que a publicação da mensagem JMS ocorresse somente após a persistência das associações, garantindo a integridade dos dados enviados.
- **Resolução de Loops de Mensagens (Eco):** Solucionei um problema de "eco" onde mensagens circulavam infinitamente entre os sistemas. Implementei um padrão de cabeçalho de origem (X-Message-Source), fazendo com que o Integrador marcasse

suas requisições e a API ignorasse a republicação de mensagens originadas pelo processo de sincronização, quebrando o loop.

- **Correção de Erros de Conexão:** Depurei e corriji as configurações de conexão em todos os três componentes (Desktop, API, Integrador) após a mudança para o ambiente Docker, ajustando os hostnames de localhost para os nomes dos serviços (activemq, livraria-api) e mapeando as portas corretamente para a comunicação entre o host e os containers.

2.3. Automação e Orquestração (DevOps)

Para transformar o projeto de um conjunto de aplicações separadas em um sistema coeso e de fácil execução, liderei a iniciativa de containerização e automação.

- **Dockerização dos Serviços:** Criei os Dockerfiles para a **LivrariaAPI** e para o **Integrador**, definindo o ambiente de execução e empacotando as aplicações em imagens de container portáteis e consistentes.
- **Orquestração com Docker Compose:** Elaborei o arquivo docker-compose.yml que define e orquestra toda a infraestrutura de backend. Este arquivo gerencia:
 - Os serviços de **MongoDB** e **ActiveMQ**.
 - A construção e execução dos containers da **API** e do **Integrador**.
 - A configuração de uma **rede virtual (livraria-net)** para a comunicação interna entre os containers.
 - O **mapeamento de volumes** para garantir a persistência dos dados do MongoDB e a acessibilidade do banco de dados do Integrador (integrador_dados.sqlite) diretamente da máquina host.
- **Criação de Scripts de Execução:** Desenvolvi os scripts start.sh (para Linux/macOS) e start.bat (para Windows), que automatizam todo o ciclo de vida do projeto com um único comando: compilação de todos os módulos, inicialização da infraestrutura Docker e execução da aplicação Desktop.

2.4. Documentação Técnica

Fui responsável por criar a documentação central do projeto, garantindo que o conhecimento fosse acessível e que o projeto fosse fácil de entender e manter.

- **Documentação Técnica Abrangente:** Elaborei o documento livrariaDoc.pdf, detalhando a arquitetura completa, os fluxos de sincronização, a estrutura de cada módulo, diagramas UML e um guia completo de instalação e execução.
- **README.md Principal:** Criei o README.md na raiz do projeto, servindo como um guia de início rápido e um portal de entrada para a documentação mais detalhada.

3. Principais Dificuldades e Soluções

- **Conflitos de Rede e Portas:** A transição para o Docker trouxe desafios como conflitos de portas (port is already allocated) e falhas de comunicação (localhost vs. nomes de serviço). Estes problemas foram resolvidos através do mapeamento de portas customizadas no docker-compose.yml e da correção das URLs de conexão nos códigos da API, Desktop e Integrador.

- **Lógica de Sincronização Assíncrona:** Depurar a ordem das operações e as condições de corrida no fluxo de mensagens exigiu uma análise cuidadosa dos logs de todos os componentes para entender o timing exato dos eventos e garantir que os dados fossem persistidos antes de serem publicados.

4. Considerações Gerais e Aprendizados

Esta etapa final do projeto foi extremamente enriquecedora, solidificando conhecimentos em áreas cruciais para o desenvolvimento de software moderno:

- **DevOps e Containerização:** Aplicação prática de Docker e Docker Compose para criar ambientes de desenvolvimento e produção replicáveis, portáteis e isolados.
- **Arquitetura de Microsserviços e Integração:** Compreensão aprofundada dos desafios e padrões de comunicação em sistemas distribuídos, como a prevenção de loops de mensagens e a gestão de redes de containers.
- **Resolução de Problemas Complexos:** A necessidade de depurar problemas que atravessam múltiplos sistemas (Desktop, Fila, Integrador, API, Banco de Dados) aprimorou significativamente minhas habilidades de diagnóstico.

5. Conclusão

Concluo minha participação no projeto com a entrega de um ecossistema de software completo, funcional e robusto. Além de cumprir com as responsabilidades originais de desenvolvimento backend, a liderança nas frentes de DevOps e a participação ativa na resolução de problemas de integração foram fundamentais para transformar um conjunto de componentes individuais em um sistema coeso e profissional. O resultado é um projeto estável, bem documentado e, acima de tudo, fácil de executar e manter.