Relatório de Produção Individual – Hugo Santos

Projeto: Sistema de Gerenciamento de Livraria

1. Atribuição de Cargo e Tarefas

- **Cargo Atribuído:** Desenvolvedor Backend.
- **Tarefas Atribuídas a Priori:** Fui primariamente responsável pelo planejamento e desenvolvimento inicial do **Sistema 2 (API RESTful)**, conforme a arquitetura do projeto. Minhas responsabilidades iniciais incluíam:
    - Modelagem e implementação da API usando Spring Boot e MongoDB.
    - Criação das funcionalidades de CRUD para as entidades principais do sistema.
    - Planejamento da integração da API com o Middleware.
    - Criação da documentação inicial e dos testes unitários para garantir a qualidade do código.
- **O que foi exercido na prática:** Na prática, exerci plenamente o papel de Desenvolvedor Backend. Fui responsável por todo o ciclo de vida da API nesta primeira etapa, desde a concepção da arquitetura, passando pela codificação das funcionalidades, até a documentação e os testes. As principais atividades executadas foram:
    - Implementação completa do CRUD para a entidade `Livro`.
    - Desenvolvimento da funcionalidade de sub-recurso para adicionar `Resenhas` a um livro.
    - Criação da documentação técnica do projeto (`README.md`) e de código (`JavaDoc`).
    - Elaboração e implementação de testes unitários para as camadas de `Repository` e `Controller`.

---

2. Contribuição de Acordo com a Atribuição

- **O que foi cumprido:** Todas as tarefas atribuídas para a etapa inicial do backend foram cumpridas com sucesso. Entreguei uma API funcional, documentada e testada, servindo como uma base sólida para as próximas etapas do projeto.
- **Lista dos 3 commits mais relevantes:**
    - **`CRUD de livro`**
    - **`endpoint de adicionar Resenha`**
    - **`Testes unitários`**
- **Documentos mais relevantes:**
    - **`README.md`:** Documento completo detalhando a arquitetura da API, as tecnologias utilizadas, e um guia passo a passo de como instalar, executar e testar todos os endpoints.

    - **Documentação JavaDoc:** Todas as classes principais (`Livro`, `Resenha`, `LivroController`, `LivroRepository`) foram documentadas seguindo o padrão JavaDoc, explicando o propósito de cada classe, método e os parâmetros esperados.
- **Principais dificuldades:** A principal dificuldade foi conceitual: definir a melhor abordagem para a sincronização de dados entre os sistemas. A análise entre usar Redis (Pub/Sub) vs. JMS/ActiveMQ, e a decisão de fazer o middleware chamar a API em vez de acessar o banco de dados diretamente, exigiu pesquisa e ponderação sobre os prós e contras de cada arquitetura em termos de acoplamento e manutenibilidade.

---

3. Contribuição Além do Atribuído

Além da implementação do código, busquei contribuir ativamente para o desenho da arquitetura do sistema como um todo. Propus e detalhei o fluxo de sincronização ODM → ORM, onde a API notifica o middleware sobre alterações (como a criação de uma resenha) para que o banco de dados da aplicação desktop seja atualizado.

Essa discussão ajudou a solidificar um padrão de comunicação desacoplado e assíncrono, que tornará o sistema mais resiliente e fácil de escalar no futuro. Também ajudei a definir um contrato claro para a API (através da documentação e dos testes), o que facilitará o trabalho da equipe que for consumir esses dados.

---

4. Considerações Gerais

- **O que aprendi:** Esta etapa do projeto foi uma excelente oportunidade para aprofundar meus conhecimentos práticos em diversas áreas:
    - **Desenvolvimento Backend:** Aplicação prática do Spring Boot para criar APIs RESTful robustas.
    - **Banco de Dados NoSQL:** Modelagem de dados com MongoDB, especialmente o uso de documentos embutidos.
    - **Arquitetura de Software:** Compreensão profunda sobre os benefícios de sistemas desacoplados, comunicação assíncrona com message brokers e os padrões de teste para cada camada da aplicação.
    - **Qualidade de Código:** A importância da documentação (JavaDoc) e dos testes unitários (JUnit e Mockito) para a manutenção e evolução de um projeto.
- **Trabalhos futuros pendentes:** Para as próximas etapas, a API está preparada para as seguintes evoluções:
    - Implementação do `Publisher` de mensagens para o ActiveMQ, ativando a sincronização da API para o sistema desktop.
- **Conclusões:** A primeira etapa do desenvolvimento do backend foi concluída com sucesso, resultando em uma base de código estável, bem-documentada e testada. A arquitetura definida se provou flexível e adequada para os requisitos do projeto,

deixando o caminho preparado para as integrações e novas funcionalidades da Etapa 2.