Projeto de Bloco: Engenharia de Softwares Escaláveis [25E3_5]

TP01

Aluno: André Augusto Ferrarez

Prof°: Leonardo Silva da Gloria

25/08/2025

SUMÁRIO

Link
https://github.com/AndreFerrarez/transacoes
TP01

1. Implementação de uma Aplicação com Spring Boot

- Design em Camadas:

A aplicação demonstra uma clara separação em camadas. A classe TransacaoController atua como a camada de controle, a TransacaoRepository como a camada de acesso a dados, e a estrutura aponta para a existência de uma TransacaoService para a lógica de negócio, atendendo ao requisito.

- Auto Configuração do Spring Boot:

O projeto faz uso efetivo da auto configuração do Spring Boot. As configurações no arquivo application.properties são suficientes para que o framework configure automaticamente o banco de dados H2, o JPA, e o servidor web embutido, simplificando o desenvolvimento.

2. Gerenciamento de Dependências

- Gerenciamento Eficaz de Dependências:

O projeto gerencia suas dependências de forma eficaz através do uso dos starters do Spring Boot, como spring-boot-starter-data-jpa e spring-boot-starter-web. Este mecanismo simplifica a gestão de bibliotecas e suas compatibilidades.

3. Desenvolvimento de APIs REST

- Implementação de APIs REST:

A competência é demonstrada na classe TransacaoController, que utiliza anotações padrão do Spring MVC como @RestController, @GetMapping e @PostMapping para construir e expor uma API REST funcional para as operações de transação.

4. Práticas de Código Limpo e Manutenção

- Aplicação de Princípios de Código Limpo e SOLID: O código está organizado e aplica princípios de design, como a separação de responsabilidades entre as camadas. A estrutura atual é limpa e de fácil manutenção. Como ponto de melhoria, sugere-se a utilização de DTOs (Data Transfer Objects) para desacoplar a API da entidade de banco de dados e a implementação de validações nos dados de entrada.

5. Modelagem de Domínios

- Modelagem de Domínios e Conceitos de DDD: A modelagem de domínio foi aplicada corretamente para o escopo do projeto. A classe Transacao representa de forma fiel a entidade de negócio central do sistema, com atributos que refletem diretamente o domínio do problema (uma carteira de transações).