

Gustavo de Oliveira Lima

## Linguagem de Programação III

### 1. Domínio e decisões de design:

“Book rating” é uma aplicação com o objetivo de fazer um gerenciamento de Livros (como para uma biblioteca, livraria, ou outros domínios de negócio que precisem).

Na fase atual, a aplicação realiza apenas operações simples, como:

Transformar os objetos do negócio para o formato JSON

#### 1.1. O que e porquê foi utilizado:

Java Reflection – Funcionalidade que permite inspecionar e manipular classes, métodos, atributos e construtores em tempo de execução, mesmo sem conhecer os elementos da classe/estejam privados (como é o caso dos atributos dos livros).

Jackson Databind – Biblioteca que permite que possamos converter os atributos dos objetos para o formato JSON, útil para simplificar o processo de serialização, deixando nosso código mais útil e eficiente.

JUnit – Para realização de testes unitários automatizados, com cenários possíveis que podem acontecer, útil para testar e observar rapidamente o funcionamento do nosso código.

### 2. Riscos:

#### 2.1. Utilização do Java Reflections:

**Redução de desempenho:** A chamada com Reflections é mais lenta do que a chamadas diretas para acesso a atributos ou métodos, pois o Java precisa realizar verificações adicionais enquanto o programa está rodando (enquanto em chamadas diretas, essas informações adicionais já são conhecidas);

**Segurança interna dos objetos:** Com esse recurso violamos o pilar “Encapsulamento” da POO, pois temos acesso a métodos e atributos de um objeto que não deveriam ser acessíveis.

**Como violamos?** Com a utilização da função `setAccessible(True)`

### **3. Como foram mitigados:**

**Em relação ao desempenho,** podemos mitigar ao reduzir o escopo de utilização do Reflection, utilizando apenas quando for necessário, e fora de trechos que sejam executados com frequência.

Em um contexto limitado, como busca de dados de um livro, é mais aceitável.

**Com relação a segurança:** Assim como no ponto acima, utilização apenas quando necessário - como nosso objetivo com o mapeamento para as DTOs é justamente evitar a propagação de dados sensíveis (como um campo senha dos usuários), já evitamos a invasão de dados que com certeza não devem ser acessados.

Podemos também utilizar ferramentas alternativas ao Reflection, como o próprio Jackson Databind, MapStruct, que conseguem manipular os objetos sem a utilização do Reflection.