Regras de négocios

Course	Spring Boot 3
Confidence	Not Confident
Next Review	@8 de agosto de 2023
Last Edited	@7 de agosto de 2023 10:41
Curso	Spring Boot 3: Documente teste e prepare uma API para o deploy

Topicos

- Classes de validação
- Aplicando princípios SOLID
- Para saber mais: princípios SOLID
- Testando o agendamento
- Boas práticas de programação
- Faça como eu fiz: validando cancelamento de consultas
- O que aprendemos?

Para saber mais: princípios SOLID

SOLID é uma sigla que representa cinco princípios de programação:

- **S**ingle Responsibility Principle (Princípio da Responsabilidade Única)
- **O**pen-Closed Principle (Princípio Aberto-Fechado)
- **L**iskov Substitution Principle (Princípio da Substituição de Liskov)

- **I**nterface Segregation Principle (Princípio da Segregação de Interface)
- **D**ependency Inversion Principle (Princípio da Inversão de Dependência)

Cada princípio representa uma boa prática de programação, que quando aplicadas facilita muito a sua manutenção e extensão. Tais princípios foram criados por Robert Martin, conhecido como *Uncle Bob*, em seu artigo **Design Principles and Design Patterns**.

Estes dois episódios do podcast Hipsters.Tech foram dedicados ao tema SOLID:

- Hipsters #129 - Práticas de Orientação a Objetos
- Hipsters #219 - SOLID: Código bom e bonito

Boas práticas de programação

Ao longo do curso, implementamos validações e regras de negócio utilizando padrões de projeto, bem como seguindo os princípios da Orientação a Objetos.

Considerando o que aprendemos, por que é importante seguir boas práticas de programação, como a utilização de Design Patterns, princípios SOLID e princípios da Orientação a Objetos nos códigos da aplicação?

- Para melhorar a manutenção do código da aplicação.
  - Boas práticas de programação tornam o código mais fácil de entender, e, principalmente, de realizar manutenção.
- Para melhorar a segurança da aplicação.
  - Não necessariamente a utilização de boas práticas vai influenciar na segurança da aplicação.
- Para melhorar a performance da aplicação.
  - Não necessariamente a utilização de boas práticas vai influenciar na performance da aplicação.

O que aprendemos?

Nesta aula, você aprendeu como:

- Isolar os códigos de validações de regras de negócio em classes separadas, utilizando nelas a anotação `@Component` do Spring;
- Finalizar a implementação do algoritmo de agendamento de consultas;
- Utilizar os **princípios SOLID** para deixar o código da funcionalidade de agendamento de consultas mais fácil de entender, evoluir e testar.
