



## 05. Mais recursos

📅 Date	@07/11/2022
📁 Categoria	Java
📖 Curso	TDD e Java: testes automatizados com JUnit

### Tópicos

- Projeto da aula anterior
- Organizando o código de teste
- Outras anotações do JUnit
- Como testar métodos privados?
- O que testar na aplicação?
- Escolhendo classes para serem testadas
- Faça como eu fiz
- Projeto final do curso
- O que aprendemos?
- Conclusão

### Organizando o código de teste

Nessa aula foi apresentado algumas anotações,

`@BeforeEach`, `@AfterEach`, `@BeforeAll` e `@AfterAll`.

`@BeforeEach`

- Executado antes de cada teste
- Não precisa ser static

#### @AfterEach

- Executado depois de cada teste
- Não precisa ser static

#### @BeforeAll

- Executado depois de todos os testes
- Precisa ser static

#### @AfterAll

- Executado antes de todos os teste
- Precisa ser static

---

## Escolhendo classes para serem testadas

Vimos que em uma aplicação é possível termos centenas, ou até mesmo milhares de classes que implementam as regras de negócio dela. Quais dessas classes são consideradas boas candidatas para testes automatizados? Apenas as classes que possuem regras de negócio e que possivelmente sofrerão bastante modificações.

---

## O que aprendemos?

Nessa aula, você aprendeu:

- Como organizar o código de uma classe de teste
- A utilizar as anotações `@BeforeEach`, `@AfterEach`, `@BeforeAll` e `AfterAll` do JUnit
- Que não se deve testar métodos privados
- Avaliar quais classes devem ser testadas em uma aplicação

