

ENGENHARIA DE SOFTWARE II

Atividade 01 – Testes Unitários com JUnit, SOLID e TDD

Data de entrega e apresentação: 19-11-2019

Importante: *Este trabalho terá peso de 50% na primeira nota*

Cenário: Uma aplicação responsável por fazer um controle de empréstimo de livros.

Para tanto, precisamos de uma classe **Livro**.

A classe **Livro** possui os atributos `autor`, `titulo`, `isEmprestado` e `isReservado`. Estes dois últimos são do tipo boolean. Também possui um atributo chamado `historico` que é uma lista de Empréstimos.

Classe `Emprestimo`: Cada empréstimo da lista contém informações como:

1. quem foi o *Usuário* que pegou o livro emprestado,
2. qual foi a data do empréstimo (`dataEmprestimo`),
3. data prevista para devolução (`dataPrevista`). A data prevista é definida pelo sistema, e sempre será 7 dias após a data do empréstimo.
4. qual foi a data da devolução (`dataDevolucao`).
5. O livro 'locado' pelo usuário.
 - a. (Cada empréstimo estará associado a um único livro)
6. O valor pago. Fica definido o valor fixo de R\$ 5,00. Caso a devolução ocorra com atraso será aplicada seguinte regra: R\$0,40 por dia de atraso, limitada a 60% do valor do aluguel.

Esta classe também possui o método `consultarEmprestimosPorUsuario()`, para o qual é passada uma instância da classe `Usuario`.

```
public class Usuario {  
    private String nome;  
    private String matricula;
```

OBS: *Um usuário poderá simultaneamente ter até 2 livros emprestados.*

Esta aplicação tem o objetivo apenas de criar elementos com as estruturas necessárias para exemplificar o uso de testes unitários.

Nota 01: Você deverá realizar decisões de projeto na organização das classes, nomenclatura, visibilidade e definição de métodos, criação de classes de serviço e de teste.

Nota 02: Programe com os princípios SOLID em mente.

Nota 03: Tente desenvolver o código com a prática do TDD.

A bateria de testes deve possuir no mínimo os seguintes cenários de teste:

- Realizar Empréstimo em livro que não esteja reservado.
- Tentativa de empréstimo em livro que já possui uma reserva.
- Garantir que a data prevista esteja correta, após a locação de um livro.
- Testa um usuário sem empréstimo.
- Usuário com 1 (um) empréstimo.
- Usuário com 2 (dois) empréstimos
- Tentativa de realizar um 3º empréstimo para o mesmo usuário.

Para os próximos testes deverão ser checados pelo menos, os atributo isEmprestado e o valorPago, com ou sem multa.

- Testar uma devolução antes da data prevista
- Testar a devolução na data prevista
- Testar a devolução 1 dia após a data prevista
- Testar a devolução 30 dias após a data prevista

Links úteis:

<http://blog.caelum.com.br/conheca-a-nova-api-de-datas-do-java-8/>

<http://www.caelum.com.br/apostila-java-testes-xml-design-patterns/testes-automatizados/#3-10-exercicios-novos-testes>