



Pontifícia Universidade Católica de Minas Gerais
Bacharelado em Engenharia de Software – Unidade Praça da
Liberdade
Laboratório de Programação Modular - Noite
Prof. Luiz Henrique da Costa Silva – Semestre 2º/2023
Projeto prático 1 – Grupo 6 - Miguel Castro, João Paulo Sales, Matheus
Avelar, Ryann Victor, Cristiano Junior.

Aplicação do Modelo TDD para Testar a Classe Equipamentos

Visão Geral da Classe Equipamento

A classe Equipamento representa um equipamento disponível para aluguel. Ela possui os seguintes atributos:

- id: Identificador único do equipamento.
- descricao: O nome ou uma breve descrição do Equipamentos.
- tipo: O tipo ou categoria do equipamento.
- valor: O valor diário do aluguel do equipamento.
- quantidadeDisponível: A quantidade disponível desse equipamento para aluguel.

1. Vermelho - Escrevendo Testes

Nesta fase, escrevemos testes que descrevem o comportamento esperado da classe Equipamento. Para cada método e comportamento relevante, foi criado um teste que inicialmente falha (ou seja, está "vermelho").

Exemplo de teste inicial:

```
@Test
public void testGetId() {
    // Configuração do teste
    Equipamento equipamento = new Equipamento(1, "Equipamento de Teste", "Tipo de Teste", 100.0, 10);

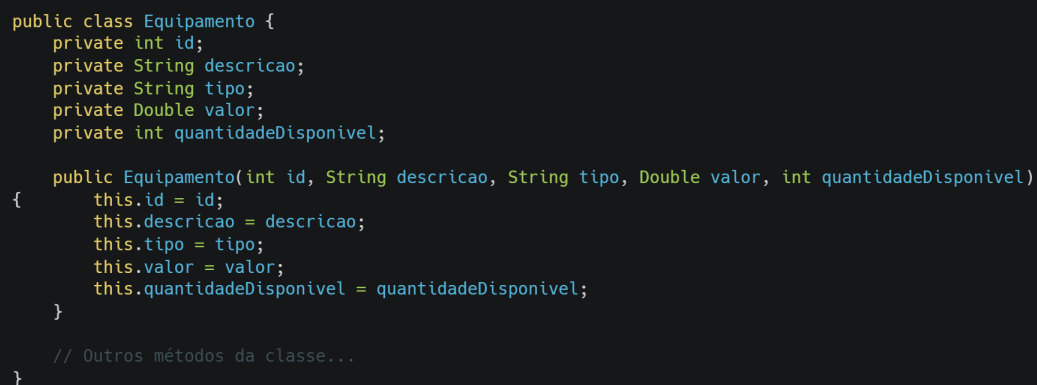
    // Execução do teste
    int id = equipamento.getId();

    // Verificação do resultado
    assertEquals(1, id); // Esperamos que o ID seja igual a 1
}
```

2. Verde - Implementação do Código

Nesta fase, implementamos a classe Equipamento e seus métodos para fazer o teste passar (ou seja, ficar "verde"). O código é escrito de maneira a satisfazer os requisitos do teste.

Exemplo de implementação inicial:



```
public class Equipamento {
    private int id;
    private String descricao;
    private String tipo;
    private Double valor;
    private int quantidadeDisponivel;

    public Equipamento(int id, String descricao, String tipo, Double valor, int quantidadeDisponivel)
    {
        this.id = id;
        this.descricao = descricao;
        this.tipo = tipo;
        this.valor = valor;
        this.quantidadeDisponivel = quantidadeDisponivel;
    }

    // Outros métodos da classe...
}
```

3. Refactor (Refatorar) - Melhorando o Código

Após passar no teste inicial, podemos refatorar o código para melhorar a qualidade e a eficiência. Durante essa fase, não alteramos o comportamento do código, apenas o organizamos e o otimizamos.

Adição de Mais Testes

Além do teste de getId, devemos adicionar mais testes para cobrir todos os métodos e comportamentos da classe Equipamento. Por exemplo, testes para os métodos getDescricao, getTipo, getValor, getQuantidadeDisponivel e validaQuantidadeDisponivel. Cada teste deve verificar se o método retorna o valor esperado com base na configuração inicial.

Testes de Limites

Também é importante criar testes que abordem cenários de limite, como tentar alugar mais equipamentos do que estão disponíveis

(validaQuantidaDisponivel) ou verificar se os métodos funcionam corretamente com valores mínimos ou máximos.

Testes de Exceção

Se a classe Equipamento lançar exceções em determinadas situações, é essencial criar testes para capturar essas exceções e garantir que elas sejam tratadas corretamente.

Conclusão

A aplicação do TDD para testar a classe Equipamento é uma abordagem eficaz para garantir que a classe funcione conforme o esperado e que possíveis problemas sejam detectados precocemente no processo de desenvolvimento. Ao seguir o ciclo "Red-Green-Refactor" e adicionar testes abrangentes, podemos ter confiança na qualidade e na robustez da classe Equipamento. Esse processo também facilita a manutenção e a evolução do código no futuro.