

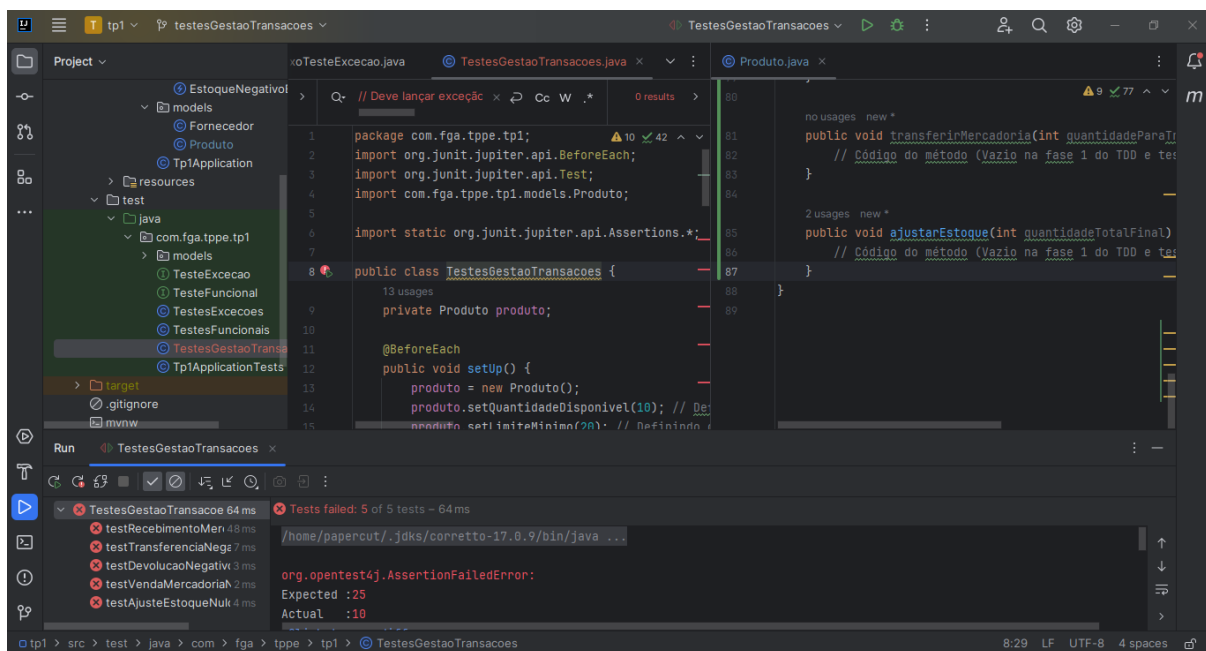
Gestão de transações - Processo de TDD

Funcionalidade: Gestão de transações

Membro do grupo: Eric Chagas de Oliveira - 180119508

Github: Eric-chagas

1. Inicialmente foram desenvolvidos os métodos de teste, que validavam se ao tentar inserir valores negativos para cada uma das transações (com exceção do ajuste de estoque), era lançada uma exceção de `IllegalArgumentException`. Com os testes desenvolvidos os métodos que estavam sendo testados foram adicionados à classe produto, sem conteúdo para que falhassem intencionalmente, e o resultado foi o seguinte:



The screenshot shows an IDE with three open files: `TesteExeccao.java`, `TestesGestaoTransacoes.java`, and `Produto.java`. The `TestesGestaoTransacoes.java` file contains the following code:

```
package com.fga.tppe.tp1;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import com.fga.tppe.tp1.models.Produto;

import static org.junit.jupiter.api.Assertions.*;

public class TestesGestaoTransacoes {
    private Produto produto;

    @BeforeEach
    public void setUp() {
        produto = new Produto();
        produto.setQuantidadeDisponivel(10); // Deve lançar exceção
        produto.setLimiteMinimo(20); // Definindo
    }
}
```

The `Produto.java` file contains the following code:

```
no usages new *
public void transferirMercadoria(int quantidadeParaTr
    // Código do método (Vazio na fase 1 do TDD e tes
}

2 usages new *
public void ajustarEstoque(int quantidadeTotalFinal)
    // Código do método (Vazio na fase 1 do TDD e tes
}
```

The Run tab shows the test results for `TestesGestaoTransacoes`. The tests failed, with the following error message:

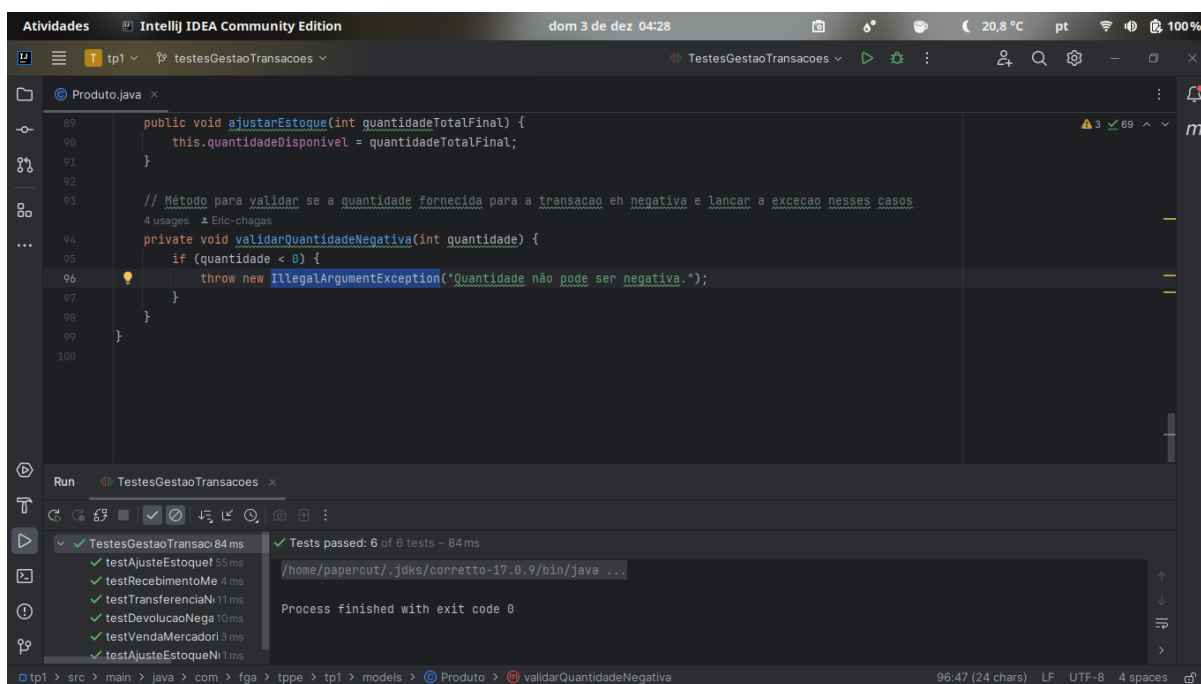
```
org.opentest4j.AssertionFailedError:
Expected :25
Actual   :10
```

Todos os testes falharam como esperado, inclusive o de recebimento de mercadoria e de ajuste no estoque, isso por que as funcionalidades não existe, e apesar de não se esperar que os testes deles lancem exceções, o resultado correto da execução deles também não podia ser esperado até então.

2. Após isso, vale notar que alguns erros foram corrigidos, como a classe `transferirMercadoria` que não era utilizada em nenhum lugar por uma falha em

chamar o nome correto do método durante os testes, bem como os métodos de teste com duas chamadas para o método sendo testado sem necessidade.

3. Após essas correções, foram desenvolvidos os conteúdos dos métodos na classe “produto”, que inicialmente consistiam de implementações simples com fluxo de condição if-else, para realizar as operações com os valores passados e lançar exceção caso necessário.
4. Os testes todos passaram com sucesso, além de haver sido adicionado um novo teste de ajuste de estoque, para validar que o método funciona normalmente mesmo com valores negativos, um dos requisitos da aplicação.



```
89 public void ajustarEstoque(int quantidadeTotalFinal) {
90     this.quantidadeDisponivel = quantidadeTotalFinal;
91 }
92
93 // Método para validar se a quantidade fornecida para a transacao eh negativa e lancar a excecao nesses casos
94 // 4 usages Eric-chagas
95 private void validarQuantidadeNegativa(int quantidade) {
96     if (quantidade < 0) {
97         throw new IllegalArgumentException("Quantidade não pode ser negativa.");
98     }
99 }
100
```

Run TestesGestaoTransacoes

Tests passed: 6 of 6 tests - 84 ms

- testAjusteEstoque 55 ms
- testRecebimentoMe 4 ms
- testTransferenciaN 11 ms
- testDevolucaoNega 10 ms
- testVendaMercadori 3 ms
- testAjusteEstoqueN 1 ms

Process finished with exit code 0

Testes passaram.

5. Após isso, foram refatoradas as classes para que houvesse mais modularização e o código ficasse mais legível. Nessa etapa, as validações individuais de quantidade negativa, que haviam em cada um dos métodos, bem como o lançamento da exceção, tornaram-se um novo método `validarQuantidadeNegativa`, que é utilizado por todos os outros.

Todos os commits realizados nesse processo estão acessíveis no github do Grupo <https://github.com/DouglasMonteles/tp1/>, e para essa funcionalidade em específico, foram todos de minha autoria (Eric Chagas de Oliveira). Os commits foram esses abaixo:

Commits

testesGestaoTr...

Commits on Dec 3, 2023

Refatorando o código dos métodos de transações com mais modularidade ...	21c8cc0
Eric-chagas committed 1 minute ago	
Adicionando conteúdo dos métodos que fazem os testes passarem	079f3e6
Eric-chagas committed 15 minutes ago	
Corrigindo teste de ajuste de estoque e adicionando caso de teste de ...	d971c44
Eric-chagas committed 16 minutes ago	
Ajustando as chamadas do transferir e corrigindo os asserts de except...	c6598d9
Eric-chagas committed 24 minutes ago	
Adicionando os métodos de teste e os métodos de gestão de transferenc...	8bba37a
Eric-chagas committed 32 minutes ago	