



**Universidade Federal do Ceará**  
**Centro de Ciências/Departamento de Computação**  
**Código da Disciplina:** CK0236  
**Professor:** Ismayle de Sousa Santos

**Aula 08**

# **Técnica de Programação II**

## **Desenvolvimento Guiado Por Testes**

---



**qpg4p5x**



**ismaylesantos@great.ufc.br**



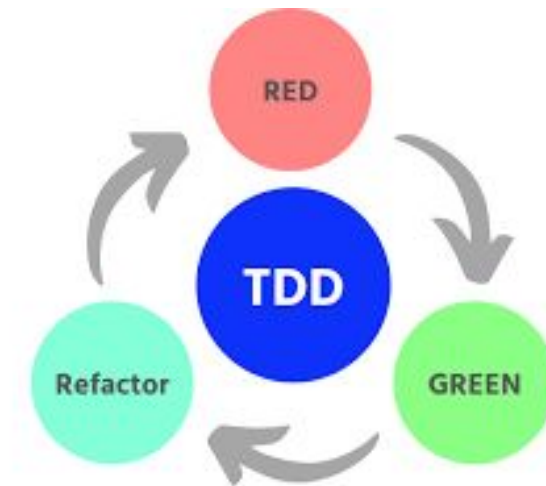
**@IsmayleSantos**

# Agenda

- TDD
    - O que é?
    - Exemplo
    - Padrões de TDD
  - BDD
  - TDD vs ATDD vs BDD
-

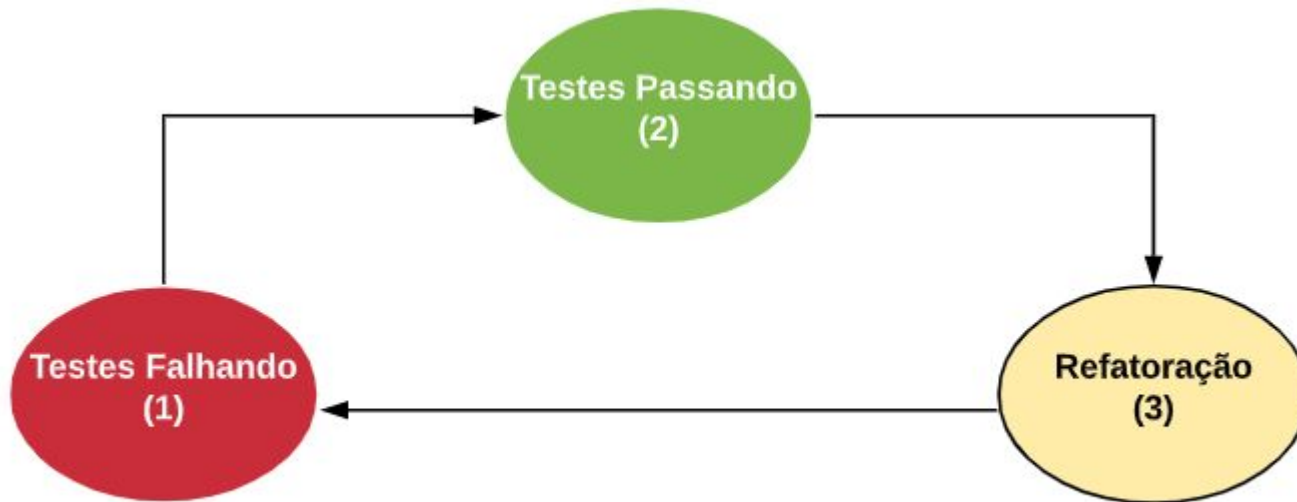
# Desenvolvimento Dirigido por Testes

- É uma das práticas de programação propostas por Extreme Programming (XP)
- A ideia base é escrever os testes antes do código



# Desenvolvimento Dirigido por Testes

- Ciclo do TDD



Ciclos de TDD

# Desenvolvimento Dirigido por Testes

- TDD foi proposto para:
  - TDD ajuda a evitar que os desenvolvedores esqueçam de escrever testes
  - TDD favorece a escrita de código com alta testabilidade
  - TDD é uma prática relacionada não apenas com testes, mas também com a melhoria do design de um sistema

# Desenvolvimento Dirigido por Testes

- Ciclo do TDD

Book tem “preço” como atributo  
**ShoppingCart** deve ter métodos para  
adicionar no carrinho, retornar o valor  
total e remover do carrinho

```
@Test
void testGetTotalVazio() {
    ShoppingCart cart = new ShoppingCart();
    assertEquals(0.0, cart.getTotal());
}

@Test
void testAddOneBookGetTotal() {
    ShoppingCart cart = new ShoppingCart();
    cart.add(book1);
    assertEquals(45.0, cart.getTotal());
}

@Test
void testAddTwoBooksGetTotal() {
    ShoppingCart cart = new ShoppingCart();
    cart.add(book1);
    cart.add(book2);
    assertEquals(70.0, cart.getTotal());
}

@Test
void testRemoveBookCartEmpty() {
    ShoppingCart cart = new ShoppingCart();
    assertThrows(NullPointerException.class, cart.removeBook(book1));
}

@Test
void testRemoveBookGetTotal() {
    ShoppingCart cart = new ShoppingCart();
    cart.add(book1);
    cart.add(book2);
    cart.removeBook(book1);
    assertEquals(25.0, cart.getTotal());
}
```

---

# Desenvolvimento Dirigido por Testes

- Ciclo do TDD




Testes Falhando  
(1)

```
public class ShoppingCart {  
  
    public boolean[] getTotal() {  
        // TODO Auto-generated method stub  
        return null;  
    }  
  
    public Executable removeBook(Book book1) {  
        // TODO Auto-generated method stub  
        return null;  
    }  
  
    public void add(Book book2) {  
        // TODO Auto-generated method stub  
    }  
  
}
```

---

# Desenvolvimento Dirigido por Testes

- Ciclo do TDD



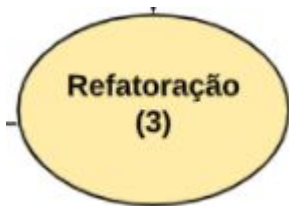
Testes Passando  
(2)

```
public class ShoppingCart {  
  
    private ArrayList<Book> items = new ArrayList<Book>();  
  
    public double getTotal() {  
        if (items.size() == 0)  
            return 0.0;  
        double total = 0;  
        for (Book book : items) {  
            total += book.getPreco();  
        }  
        return total;  
    }  
  
    public void removeBook(Book book1) {  
        if (items.size() == 0)  
            throw new NullPointerException();  
        else  
            items.remove(book1);  
    }  
  
    public void add(Book book2) {  
        items.add(book2);  
    }  
}
```



# Desenvolvimento Dirigido por Testes

- Ciclo do TDD



```
public class ShoppingCart {  
  
    private ArrayList<Book> items; //changed  
  
    public ShoppingCart() {  
        items = new ArrayList<>(); //changed  
    }  
  
    public double getTotal() {  
        if (items.isEmpty()) //changed  
            return 0.0;  
  
        double total = 0;  
        for (Book book : items) {  
            total += book.getPreco();  
        }  
        return total;  
    }  
  
    public void removeBook(Book book) { //changed  
        if (items.isEmpty()) //changed  
            throw new NullPointerException("Lista já estava vazia"); //changed  
        else  
            items.remove(book);  
    }  
  
    public void add(Book newBook) { //changed  
        items.add(newBook);  
    }  
}
```

# Desenvolvimento Dirigido por Testes

- Algumas observações
  - Ao criar os testes você pode inclusive não ter nem a classe que está sendo testada criada
  - O passo a passo pode variar
    - Pouco código, retesta.. Pouco código retesta
    - Estratégias
      - Fake it (retorna constantes e gradualmente vai substituindo)
      - Implementação Direta
  - Na refatoração
    - Eliminar dependência entre código e os testes
    - “Se dependência é o problema, duplicação é o sintoma”

# Padrões para TDD

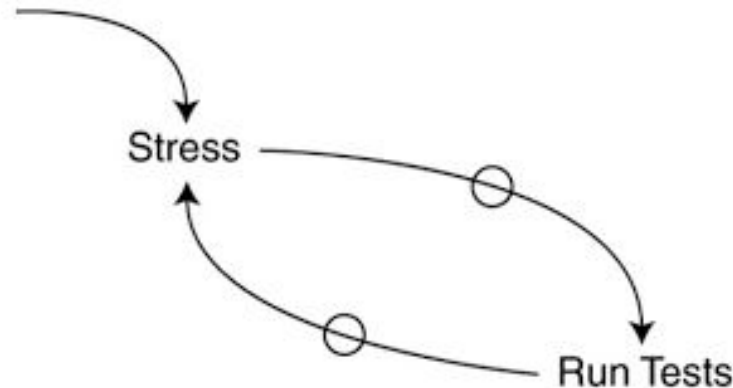
- Testes isolados (Isolated Test)
  - A execução de um teste não deve afetar outro
  - A ordem de execução dos testes não deve importar

# Padrões para TDD

- **Lista de Teste (Test List)**
  - **Antes de começar, escreva uma lista de todos os testes que você vai precisar implementar**
  - **Exemplos**
    - **Cada operação que você precisa implementar**
    - **Listar refatorações que serão necessárias**

# Padrões para TDD

- **Teste Primeiro (Test First)**
  - Escreva os testes antes de escrever o código que será testado
  - Testar primeiro implica em reduzir o estresse!



# Padrões para TDD

- **Assert First**
  - **Tente escrever os asserts primeiro!**
    - Qual é a resposta correta?
    - Como verificar?

# Padrões para TDD

- **Test Data**
  - Use dados que tornem o teste fácil de ler e seguir
  - Nunca use a mesma constante para significar mais de uma coisa
  - Uma opção é usar Dados do mundo real

# Padrões para TDD

- **Evident Data**
  - Você não está escrevendo testes só para um computador
  - Tente deixar o relacionamento entre resultados reais e esperados claros
    - Exemplo: deixando a fórmula de cálculo explícita



# Agora vamos falar do BDD ..



# Desenvolvimento Guiado por comportamento

- BDD (Behavior Driven Development)
    - Idealizado por Dan North
    - “Evolução do TDD”
      - os testes ainda orientam o desenvolvimento
    - Usar exemplos para descrever o comportamento de uma aplicação ou unidades de código
    - Automatizar os exemplos para prover um feedback rápido e testes de regressão
-

# Desenvolvimento Guiado por comportamento

- Vantagens do BDD
    - Melhora a Comunicação entre a equipe
    - Todos os envolvidos podem especificar mais cenários
    - Documentação do sistema
    - Incentiva o reuso de código de testes
-

## Ferramentas - BDD

specflow 

Cucumber 

 *jbehave*

---

# Desenvolvimento Guiado por comportamento

- Exemplo de User Story

*Funcionalidade : [Nome]*

*Para [ Valor ao Negócio ] Eu, como [ Papel ] Desejo poder realizar [ Funcionalidade ]*

*Cenário : [ Nome ]*

*Dado que [ Estado inicial do sistema ] Quando [ Ação a ser realizada no sistema ] Então [ Coisas que o sistema deve fazer após a ação do Quando ]*

# Desenvolvimento Guiado por comportamento

- Exemplo

**Feature:** Calculator

In order to avoid silly mistakes

As a math idiot

I want to be told the sum of two numbers

@mytag

**Scenario:** Add two numbers

**Given** I have entered 50 into the calculator

**And** I have also entered 70 into the calculator

**When** I press add

**Then** the result should be 120 on the screen

---

# Desenvolvimento Guiado por comportamento

- Exemplo

Step 1

```
[Given(@"I have entered (.*) into the calculator")]  
public void (GivenIHaveEnteredIntoTheCalculator(int number)
```

Step 2

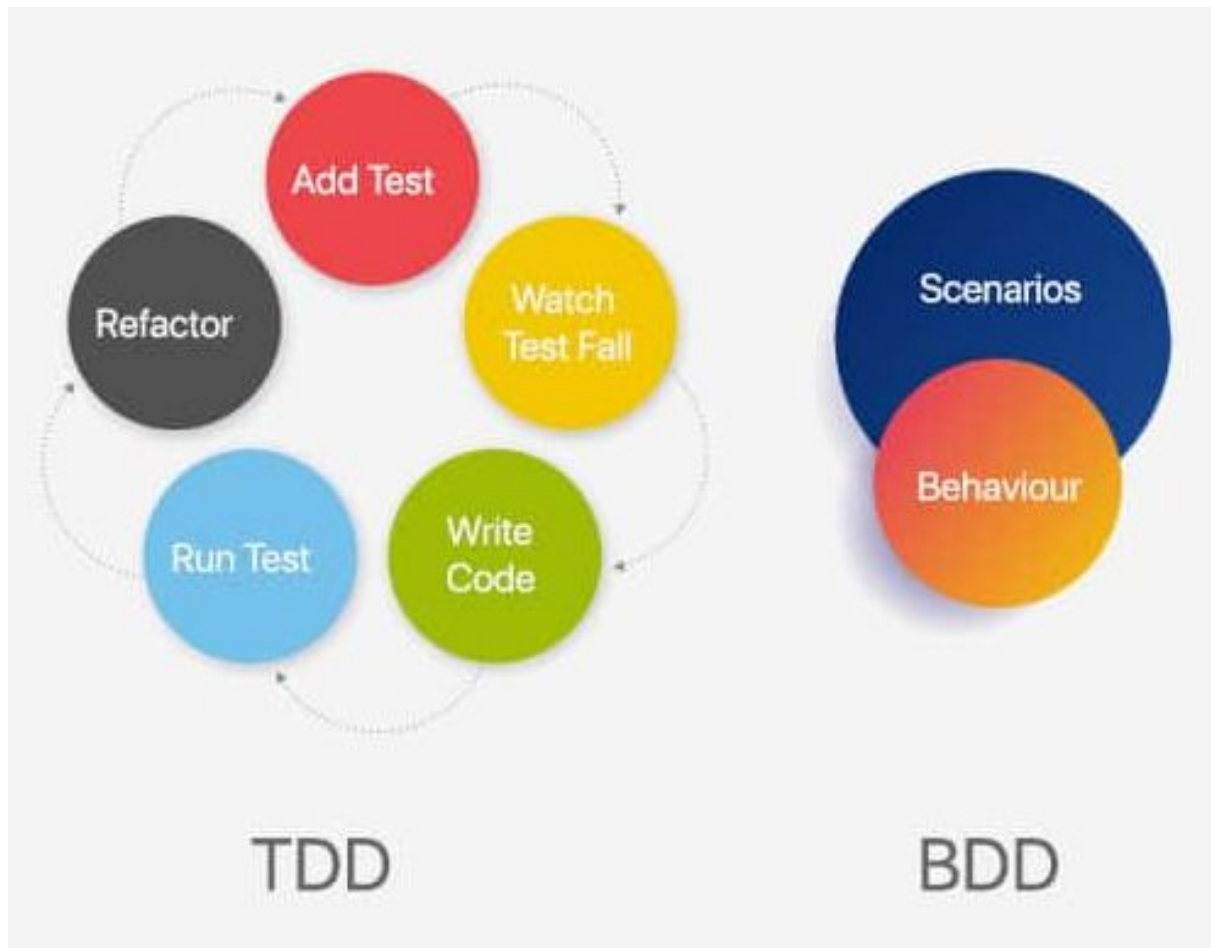
```
public void (GivenIHaveEnteredIntoTheCalculator(int number)  
{  
    calculator.FirstNumber = number;  
}
```

Step 3

```
public int FirstNumber {get; set; }
```

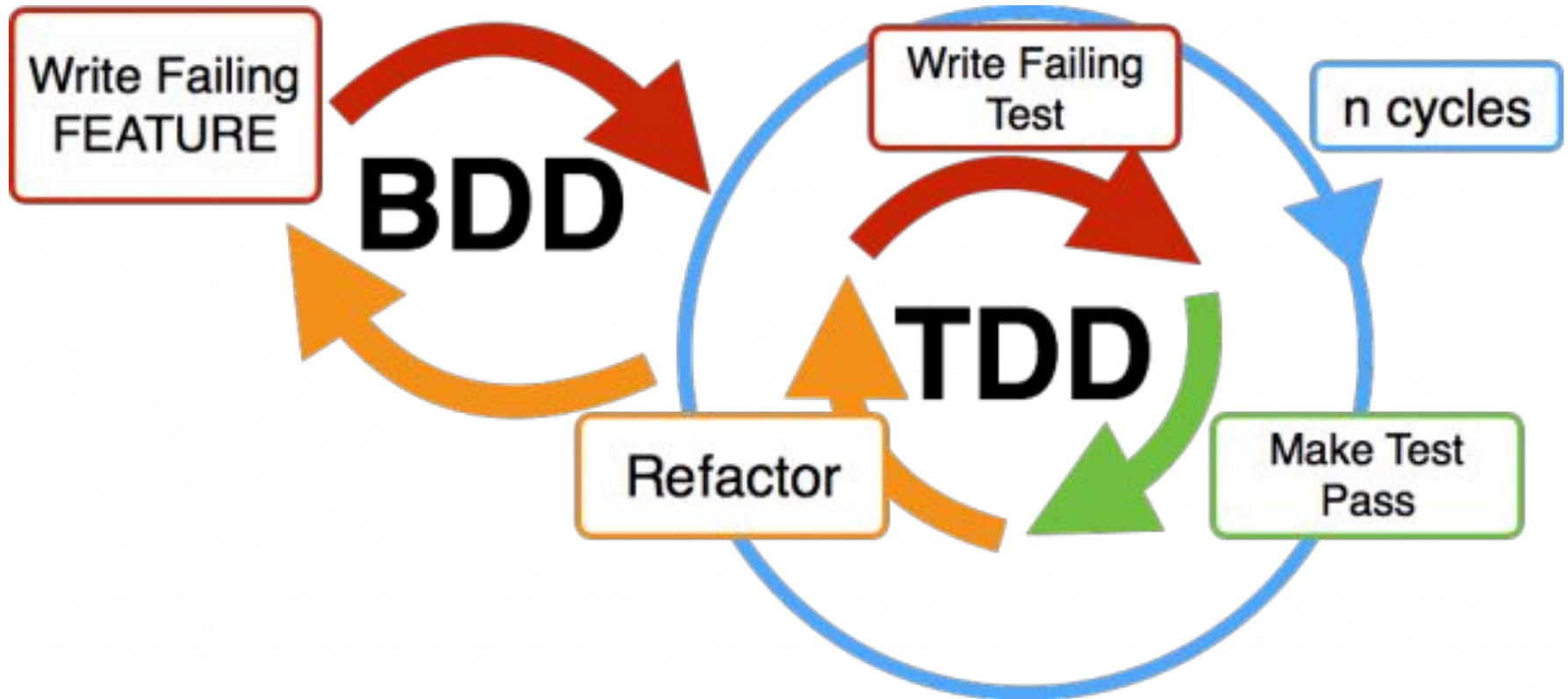
---

# BDD e TDD





# BDD e TDD



# Desenvolvimento orientado a testes de aceitação (ATDD)

- Apoia na obtenção de requisitos de forma colaborativa
- Criação de Testes antes do Código
- Exemplos concretos e testes automatizados para especificar os requisitos
  - “Especificações executáveis”
- Teste de aceitação
  - Focado no ponto de vista do usuário



# TDD vs BDD vs ATDD

	<b>TDD</b>	<b>BDD</b>	<b>ATDD</b>
Foco	Implementação de uma funcionalidade  Testes Unitários	Comportamento do sistema (de uma feature)  Entendimento dos Requisitos	Foca na captura dos requisitos  Escrita de Testes de Aceitação
Participantes	Desenvolvedor	Desenvolver, Clientes, QAs	Desenvolvedor, Cliente, QAs

# *Obrigado!*

## *Por hoje é só pessoal...*

# **Dúvidas?**



qpg4p5x



ismaylesantos@great.ufc.br



@IsmayleSantos

---