



Instituto Federal do Piauí – IFPI

**ANTONIO HITTALO RAMYRES PAULO RODRIGUES MACEDO
BENTO KAUÊ DE SOUSA LIMA
JOÃO MANUEL DA SILVA PAULO**

Test Driven Development (TDD) no Desenvolvimento de Software

PEDRO II – PI

2025

1. Introdução ao TDD

O Test Driven Development (TDD) é uma metodologia de desenvolvimento de software que enfatiza a escrita de testes automatizados antes da implementação do código-fonte. Trata-se de uma metodologia ágil onde o desenvolvedor escreve um teste automatizado que falha antes mesmo de escrever o código de produção, desse modo é garantido que o código implementado atenda a uma necessidade específica, resultando em um código limpo, seguro e flexível, tudo isso através de seu ciclo principal: Red → Green → Refactor:

Red: Escrevendo um teste que inicialmente falha, pois o código ainda não foi implementado.

Green: Implemente o mínimo de código necessário para que o teste passe.

Refactor: Melhore o código mantendo os testes passando, garantindo qualidade e manutenção sem comprometer a funcionalidade.

A aplicação do TDD promove código mais confiável, legível e fácil de manter, pois cada funcionalidade é validada por testes desde o início do desenvolvimento até qualquer fase final do projeto em desenvolvimento.

2. Aplicação do TDD no projeto

O projeto consistiu no desenvolvimento de um módulo de cálculo e validação de operações básicas, implementado na classe.

2.1 Métodos implementados

Foram criados os seguintes métodos:

somar(a, b): Retorna a soma de dois números.

subtrair(a, b): Retorna a subtração de dois números.

multiplicar(a, b): Retorna a multiplicação de dois números.

dividir(a, b): Retorna o quociente da divisão, lançando exceção ao tentar dividir por zero.

isPar(numero): Retorna verdadeiro se o número for par, falso caso contrário.

validarNumeroPositivo(numero): Retorna verdadeiro para números positivos, considerando zero como positivo, e falso para números negativos.

2.2 Testes unitários

Seguindo a metodologia TDD, todos os testes foram escritos antes da implementação. Foram cobertas todas as situações descritas na atividade, conforme citado abaixo:

somar(a, b): soma de dois positivos, soma de positivo e negativo, soma de zeros.

subtrair(a, b): subtração correta, resultado negativo, valores iguais.

multiplicar(a, b): multiplicação de dois positivos, multiplicação com zero, multiplicação com negativos.

dividir(a, b): divisão correta, divisão por zero, divisão com números decimais.

isPar(numero): números pares, ímpares e zero.

validarNumeroPositivo(numero): números positivos, negativos e zero.

Todos os testes foram executados com sucesso, não sendo observadas dificuldades na implementação. O ciclo Red → Green → Refactor foi seguido à risca, garantindo confiabilidade e organização do código o que facilitou o serviço.

3. Dificuldades encontradas

Durante a atividade, não foram observadas dificuldades significativas, uma vez que a escrita dos testes e a implementação dos métodos seguiram de forma direta o ciclo do TDD e informações passadas no enunciado do trabalho.

A decisão de considerar zero como número positivo foi mantida até o presente momento do desenvolvimento deste relatório, garantindo consistência com as regras matemáticas básicas e a lógica dos testes.

4. Benefícios percebidos

- Garantia de funcionalidades corretas desde o início, evitando erros silenciosos durante todo o desenvolvimento do trabalho.
- Código mais organizado e modular, facilitando manutenção futura.
- Maior confiança na implementação, já que cada método possui testes que validam seu comportamento durante todo o processo.
- Aprendizado prático da importância de escrever testes antes do código, reforçando boas práticas de engenharia de software

5. Conclusão

O projeto permitiu compreender e aplicar a metodologia TDD, desenvolvendo um módulo de calculadora de forma incremental e testada. A experiência demonstrou que testes unitários escritos antes da implementação promovem código confiável, seguro e de fácil manutenção, confirmando os princípios do TDD como uma prática essencial no desenvolvimento de software moderno.