Desenvolvimento Dirigido a Testes

Prof. Msc. Carlos Augusto Mar

FIC TDD - 2021.1

Projeto Aranouá

TDD (Test Driven Development)

- Técnica de desenvolvimento de software, criada por Kent Beck, que se relaciona com os conceitos de verificação e validação baseada em curto ciclo de repetições;
- Ciclo: desenvolvimento de um caso de teste automatizado para uma melhoria ou nova funcionalidade, implementação do código a ser validado pelo teste e por fim a refatoração do código para padrões aceitáveis;
- Está relacionado aos conceitos de XP (Extreme Programming) que foi criado em 1999.

Requisitos

1. Criar testes automatizados que definem requisitos em código antes de se implementar a aplicação;
2. Os testes contém assertivas que podem ser falsas ou verdadeiras e determinam se o teste falha ou não;
3. Os testes são executados de forma contínua conforme o desenvolvimento de cada funcionalidade;
4. Utilização de frameworks de automação de testes como o Junit, por exemplo.

Modelo de teste

- Devem ser automatizados, precisarem de somente um comando para que sejam executados por completo sem necessidade de intervenção manual, seguir determinados princípios.
- Princípios **FIRST.**

Princípios FIRST

- F (Fast) — Rápidos: a execução de um teste não pode ser demorada;
- I (Isolated) — Isolados: testes unitários são isolados e não podem ser afetados — nem afetar — a execução de outros;
- R (Repeatable) — Reproduzíveis: eles não devem depender de dados do ambiente/instância, sendo consistentes ao fornecer os mesmos resultados para as mesmas entradas;
- S (Self-verifying) — Auto Verificáveis: um teste deve conter todas as informações necessárias para reproduzir uma resposta simples: passou ou falhou;
- T (Timely) — Oportuno: eles devem cobrir todos os cenários possíveis e serem testados para exceções, argumentos inválidos, etc.

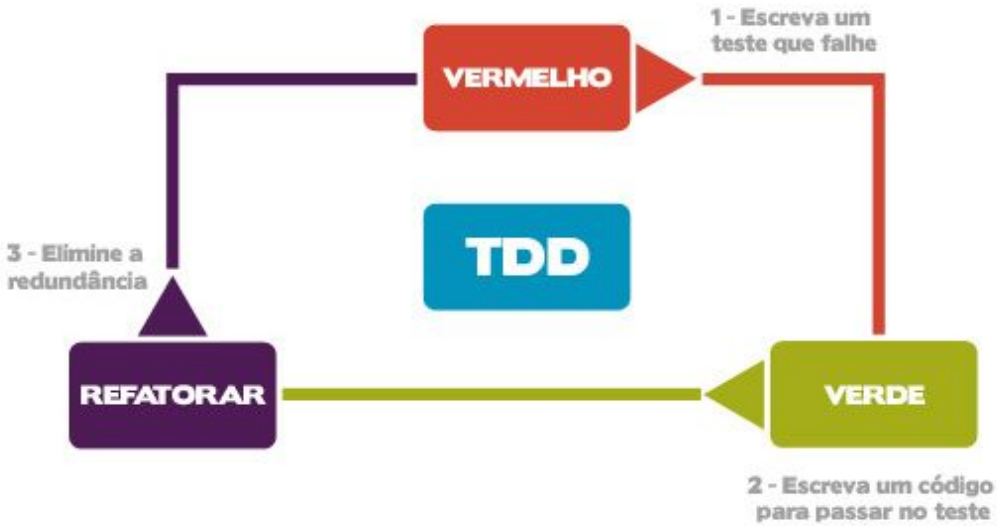
Benefícios

1. Código com maior qualidade, coeso e pouco acoplado;
2. Aumento da confiança da equipe de desenvolvimento no código escrito e seu funcionamento;
3. *Feedback* rápido sobre possíveis quebras de código;
4. Testes de regressão automatizados e mais rápidos;
5. Cobertura do código implementado pelos testes desenvolvidos anteriormente;
6. Redução de custos;
7. Manutenibilidade.

Ciclo de Desenvolvimento

1. Adicionar um teste;
2. Executar o teste e ver ele falhar;
3. Escrever o código que é validado pelo teste;
4. Executar o teste e ver ele passar;
5. Refatorar o código (melhorar o código escrito);
6. Voltar ao passo 1.

Ciclo TDD



Fonte: DevMedia

Exemplo: Carrinho de compras

Vermelho

```
@Test
void testAddGetTotal() {
  Book b1 = new Book("book1", 10, "1");
  Book b2 = new Book("book2", 20, "2");
  ShoppingCart cart = new ShoppingCart();
  cart.add(b1);
  cart.add(b2);
  assertEquals(30.0,cart.getTotal());
}
```

Fonte: Engenharia de Software Moderna

```
public class Book {
  public String title;
  public double price;
  public String isbn;

  public Book(String title, double price, String isbn) {
    this.title = title;
    this.price = price;
    this.isbn = isbn;
  }

}

public class ShoppingCart {

  public ShoppingCart() {}

  public void add(Book b) {}

  double getTotal() {
    return 0.0;
  }
}
```

Ainda vermelho, mas pelo menos compilando

Implementação ainda provisória

Fonte: Engenharia de Software Moderna

Primeiro Verde

```
public class ShoppingCart {
  public ShoppingCart() {}
  public void add(Book b) {}
  double getTotal() {
    return 30.0;
  }
}
```

Fonte: Engenharia de Software Moderna

```
public class ShoppingCart {

  private ArrayList<Book> items;

  private double total;

  public ShoppingCart() {
    items = new ArrayList<Book>();
    total = 0.0;
  }

  public void add(Book b) {
    items.add(b);
    total += b.price();
  }

  double getTotal() {
    return total;
  }

}
```

Agora um verde real

Fonte: Engenharia de Software Moderna

Amarelo: podemos refatorar e melhorar o código?

```
public class Book {
  public String title;
  public double price;
  public String isbn;

  public Book(String title, double price, String isbn) {
    this.title = title;
    this.price = price;
    this.isbn = isbn;
  }

}
```

Por exemplo, encapsular esses campos

Fonte: Engenharia de Software Moderna

FIM.