



TDD

▼ O que é TDD

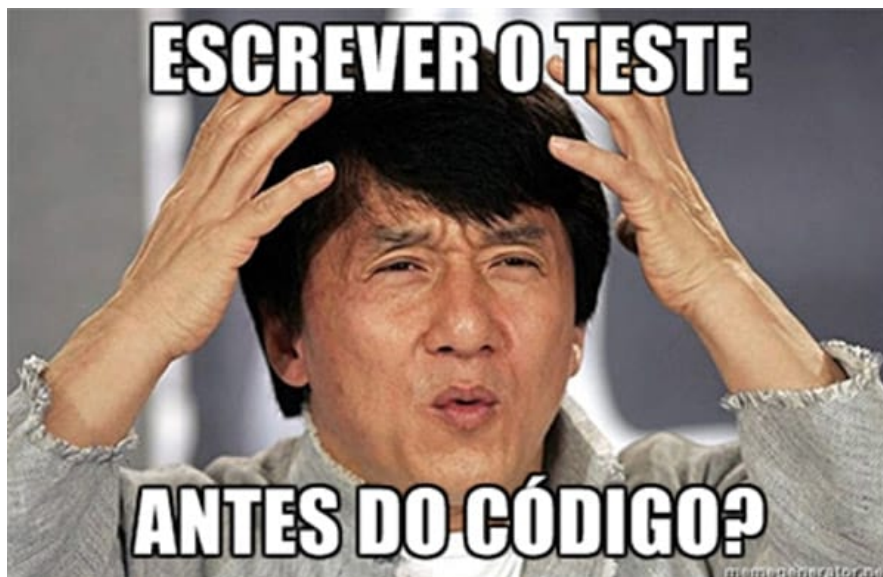
▼ Origem

Surgiu por volta de 1990, onde foi introduzida inicialmente pelo XP (Extreme Programming), por Kent Beck.

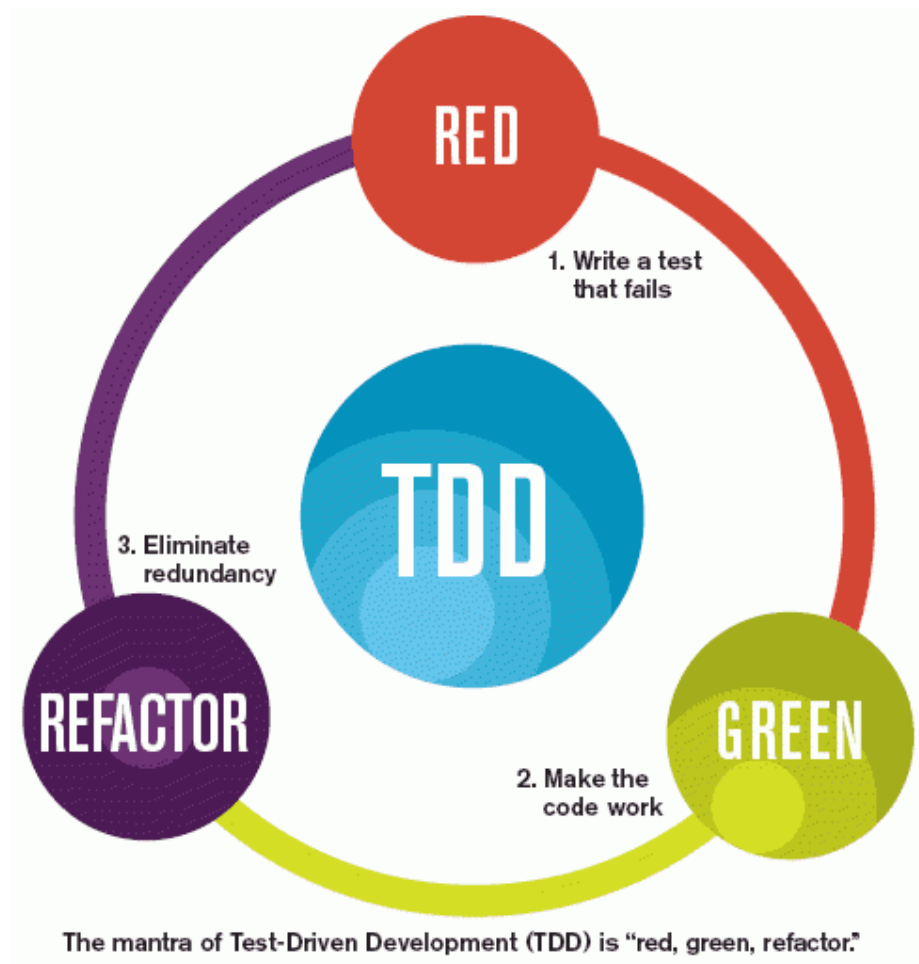
O intuito era de encorajar o desenvolvimento de códigos simples, para que pudessem ser checados e validados com uma técnica igualmente simples.

▼ Conceito

Test Driven Development - Metodologia a qual prega que os testes devem ser escritos antes da codificação propriamente dita, diferente do modelo tradicional.



▼ Etapas



▼ Bibliotecas

▼ .Net

- MSTest
- xUnit
- NUnit

▼ JavaScript

- Jest
- Mocha + Chai

▼ Java

- JUnit

▼ PHP

- PHPUnit

▼ Exemplos

▼ Boas Práticas

- ▼ Estrutura de Testes AAA e GWT

Padrões comuns de escrever testes de unidade, e que facilitam a leitura e compreensão do código com 3 passos.

- **Given/Arrange** - Definir um estado conhecido.
- **When/Act** - Rodar a aplicação
- **Then/Assert** - Certificar-se de que o novo estado é o estado esperado.

▼ Simples

▼ Step 1 - Red

```
[Fact]
public void TestsTemperatureConversor()
{
    //Given/Arrange
    double fahrenheit = 32;
    double celsius = 0;

    //When/Act
    double calculateVal= TemperatureConverter.FahrenheitToCelsius(fahrenheit);

    //Then/Assert
    Assert.Equal(celsius, calculateVal);
}
```

▼ Step 2 - Green

```
public static class TemperatureConverter
{
    public static double FahrenheitToCelsius(double temperature)
    {
        throw new NotImplementedException();
    }
}
```


▼ Step 3 - Refactor

```
public static class TemperatureConverter
{
    public static double FahrenheitToCelsius(double temperature)
    {
        return Math.Round((temperature - 32) / 1.8, 2);
    }
}
```

▼ Prática

▼ GitHub Link

GitHub - JVidalN/tdd-presentation: Assets for KT (Knowledge Transfer) presentation about TDD
Assets for KT (Knowledge Transfer) presentation about TDD - GitHub - JVidalN/tdd-presentation: Assets for
KT (Knowledge Transfer) presentation about TDD

 <https://github.com/JVidalN/tdd-presentation>

JVidalN/**tdd-
presentation**

Assets for KT (Knowledge Transfer) presentation
about TDD

 1 Contributor  0 Issues  0 Stars  0 Forks

▼ Projeto em .Net

1. Create a Blank Solution
2. Add xUnit Project
 - a. Create tests
3. Add Class library project
 - a. Create class

▼ Projeto em JavaScript

1. Project setup

`yarn install` or `yarn add jest`

2. Change package.json

```
"scripts": {  
  "test": "jest"  
}
```

3. Create tests
4. Create function
5. Execute tests

`yarn tests`

▼ Prós e Contras

▼ Prós

- Capacidade de reduzir pequenos bugs
- Código mais limpo e simples, eliminando códigos desnecessários
- Confiabilidade do código
- Favorece o processo de documentação
- Facilita o refactoring

▼ Contra

- Curva de aprendizado
- Os testes se tornam parte da manutenção do projeto
- Torna o projeto "caro"

▼ Dúvidas

