



# QUALIDADE E TESTE DE SOFTWARE

**Prof Augusto / E EU MESMO**

# TDD

Test Driven Development ou Desenvolvimento Orientado a Testes.

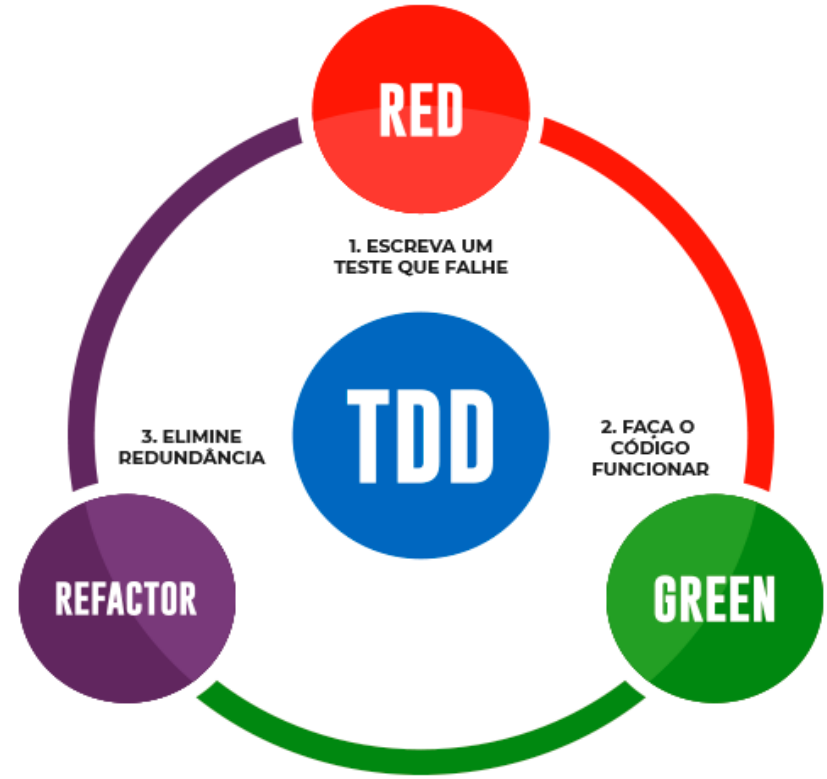
O conceito é bem simples, os testes são desenvolvidos antes de escrevermos o código de produção.

# TDD

Para que o TDD dê certo é necessário trabalhar rigorosamente com um ciclo contínuo de ações sem pular nenhuma fase.

# TDD

O ciclo de desenvolvimento é chamado de *Red, Green and Refactor* e é composto das seguintes fases.



# TDD

## 1. Novo teste:

Fase que iremos escrever o código que irá testar o resultado da nossa função, esta parte é muito importante pois é a base do nosso teste.

É importante ressaltar que é necessário especificar muito bem qual a entrada e a saída do código a ser testado.

Se você não consegue entender o que seu código vai fazer e retornar, você não poderá fazer um **teste assertivo**.

No começo não é muito intuitivo, mas com a prática aprendemos a escrever e a prever melhor os resultados de nossas funções.

# TDD

## 2. Teste falhando:

Por ainda não existir uma função o seu teste não vai funcionar (*se funcionar tem algo de errado aí*).

Com o teste falhando temos um objetivo a alcançar, fazer o teste funcionar através da nossa funcionalidade.

# TDD

## 3. Criar funcionalidade:

Nessa fase iremos escrever o nosso código da maneira mais simples possível.

O nosso objetivo é unicamente a aprovação nos testes.

Neste momento não é necessário se preocupar com as boas práticas, design patterns ou coisas do tipo, o código só deve funcionar e passar pelo teste sem quebrar outros testes.



# TDD

## 4. Teste passando:

Sinal verde! O código que você escreveu passou pelo teste e não quebrou **outros testes**.

É muito importante garantir que o seu código além de passar pelo teste desenvolvido não está quebrando os testes que já existem.

# TDD

## 5. Refatorar:

É um processo de modificar um sistema de software para melhorar a estrutura interna do código sem alterar seu comportamento externo.

Achou que não ia precisar mais se preocupar com padrões e boas práticas né? Achou errado!

# TDD

## Montando um código C#:

### **PARA NOTA ENTREGAR EM DUPLA:**

Cria um código em C# de leilão simples, onde contenha os arquivos com suas respectivas classe e orientações AVALIAR O MENOR LEILÃO E MAIOR LEILÃO:

**Usuario.cs**

**Lance.cs**

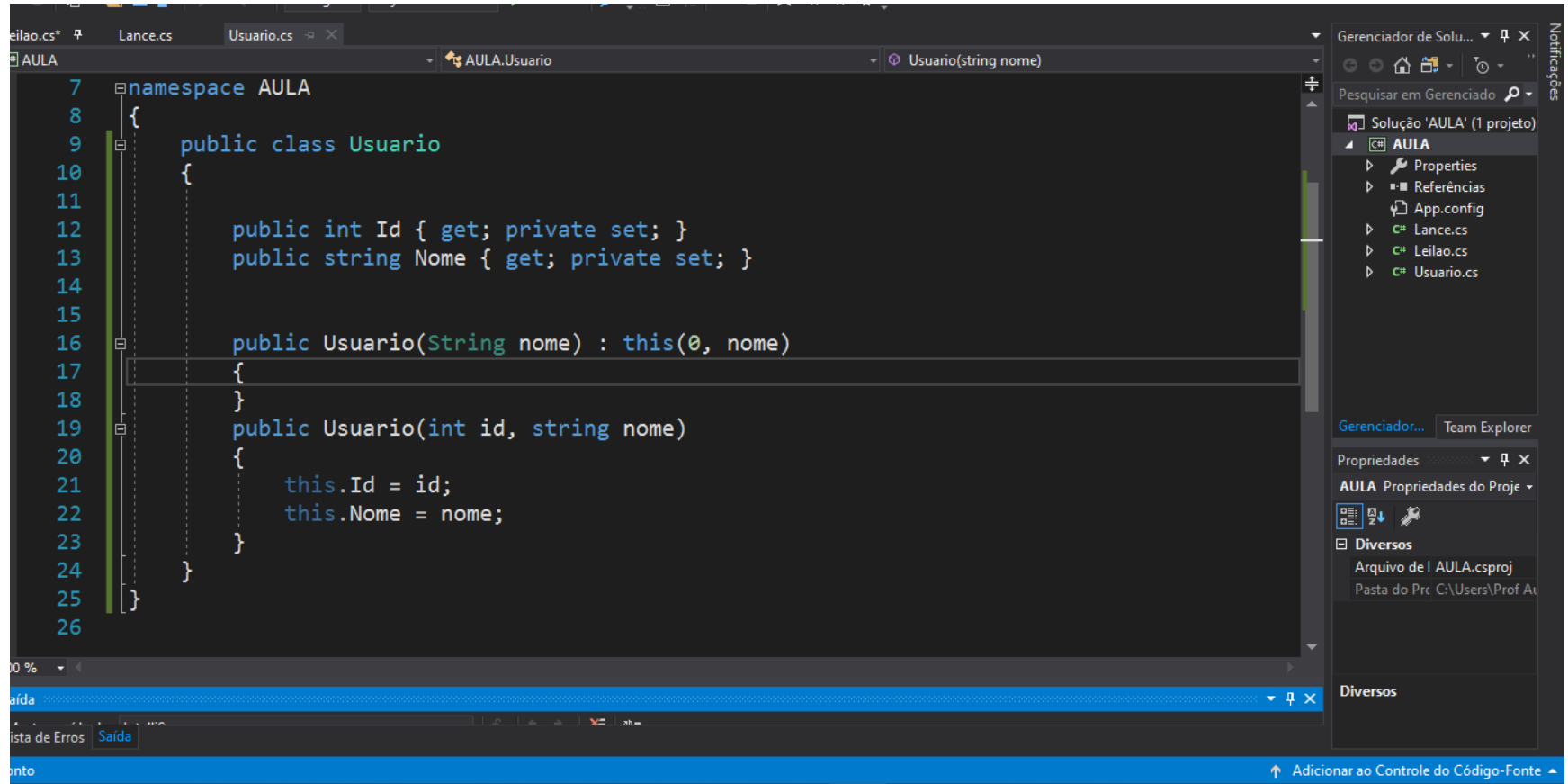
**Leilao.cs**

Depois criar as classes de testes:

**avaliador.cs** (ele vai ter que varrer todos os lances que derem, e depois vão ter que armazenar o maior e o menor valor)

**TesteDoAvaliador.cs** vai criar os usuários e alimentar o sistema com os valores.

# Usuario.cs



```
7 namespace AULA
8 {
9     public class Usuario
10     {
11
12         public int Id { get; private set; }
13         public string Nome { get; private set; }
14
15
16         public Usuario(String nome) : this(0, nome)
17         {
18         }
19         public Usuario(int id, string nome)
20         {
21             this.Id = id;
22             this.Nome = nome;
23         }
24     }
25 }
26
```

Gerenciador de Solu... Team Explorer

Pesquisar em Gerenciado

Solução 'AULA' (1 projeto)

- AULA
  - Properties
  - Referências
  - App.config
  - Lance.cs
  - Leilao.cs
  - Usuario.cs

Propriedades

AULA Propriedades do Proje

Diversos

- Arquivo de I AULA.csproj
- Pasta do Pro C:\Users\Prof At

Diversos

Adicionar ao Controle do Código-Fonte

# Lance.cs

The image shows a screenshot of the Microsoft Visual Studio IDE. The main editor window displays the code for `Lance.cs` within a project named `AULA`. The code is as follows:

```
6
7 namespace AULA
8 {
9     class Lance
10    {
11        public Usuario Usuario { get; private set; }
12        public double Valor { get; private set; }
13
14        public Lance(Usuario usuario, double valor)
15        {
16            this.Usuario = usuario;
17            this.Valor = valor;
18        }
19    }
20 }
21
22
23
```

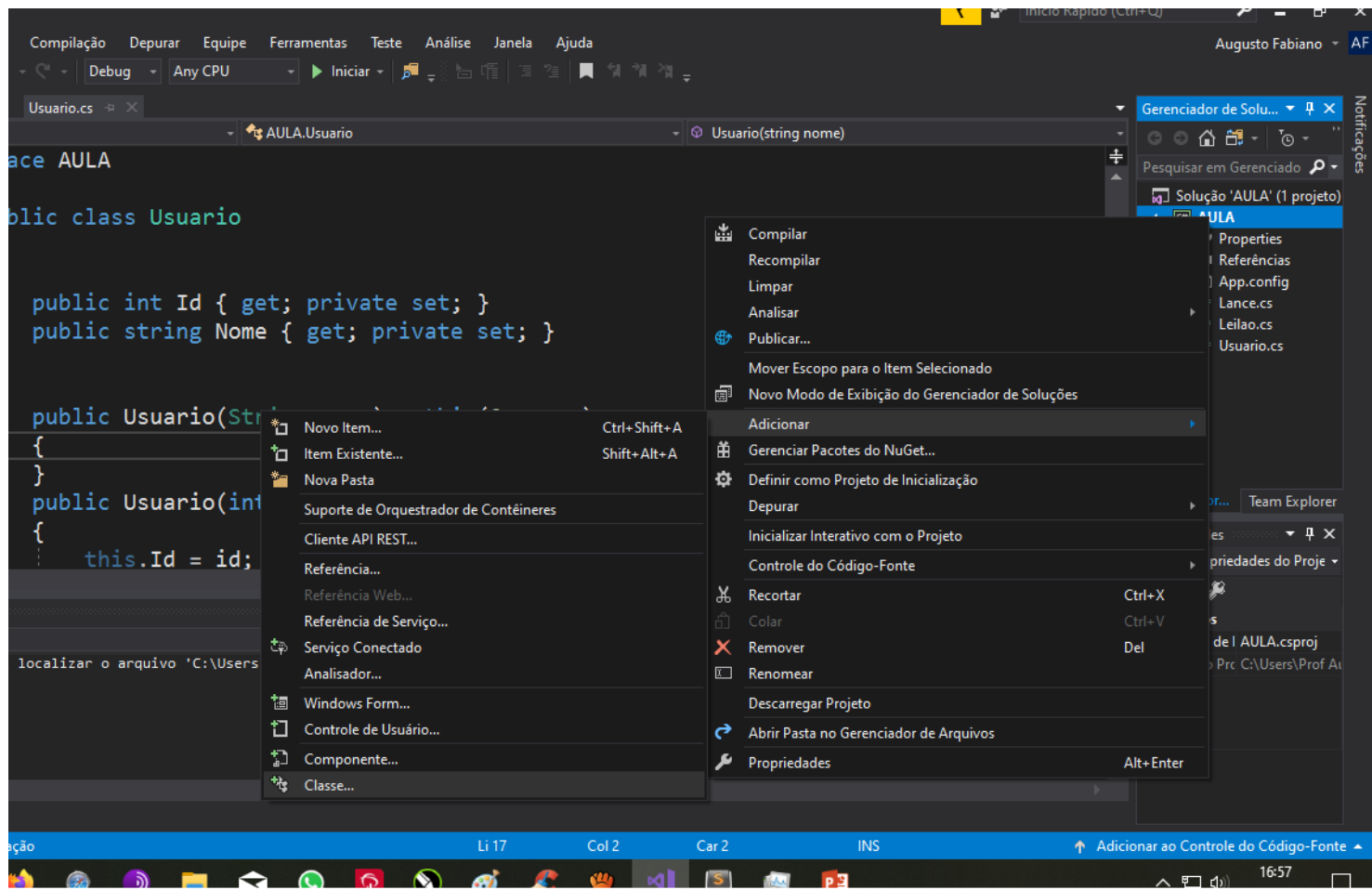
The interface includes the following components:

- Top Menu Bar:** Arquivo, Editar, Exibir, Projeto, Compilação, Depurar, Equipe, Ferramentas, Teste, Análise, Janela, Ajuda.
- Top Toolbar:** Includes icons for file operations, a search bar with "Início Rápido (Ctrl+Q)", and window management icons.
- Left Sidebar:** Gerenciador de Servidores.
- Right Sidebar:**
  - Solução 'AULA' (1 projeto):** AULA, Properties, Referências, App.config, Lance.cs, Leilao.cs, Usuario.cs.
  - Team Explorer:** Gerenciador...
  - Propriedades:** Properties window.
- Bottom Panel:** 100% zoom, Saída (Output) window, and Lista de Erros (Error List) window.
- Taskbar:** Windows taskbar at the bottom with various application icons.

# Leilao.cs

```
7
8 namespace AULA
9 {
10     public class Leilao
11     {
12
13         public string Descricao { get; set; }
14         public IList<Lance> Lances { get; set; }
15
16         public Leilao(string descricao)
17         {
18             this.Descricao = descricao;
19             this.Lances = new List<Lance>();
20         }
21
22         public void Propoe(Lance lance)
23         {
24             Lances.Add(lance);
25         }
26     }
27 }
```

# Criando outras classes



# Avaliador.cs

```
7 namespace AULA
8 {
9     public class Avaliador
10    {
11        private double maiorDeTodos = Double.MinValue;
12        private double menorDeTodos = Double.MaxValue;
13
14        public void Avalia(Leilao leilao)
15        {
16            foreach (var lance in leilao.Lances)
17            {
18                if (lance.Valor > maiorDeTodos)
19                {
20                    maiorDeTodos = lance.Valor;
21                }
22                else if (lance.Valor < menorDeTodos)
23                {
24                    menorDeTodos = lance.Valor;
25                }
26            }
27        }
28    }
29 }
```



# TesteDoAvaliador.cs

The image shows a Visual Studio IDE with the following components:

- Top Bar:** Includes icons for file operations, a dropdown menu with 'Debug' and 'Any CPU', a 'Iniciar' button, and other utility icons.
- Tab Bar:** Displays open files: 'Leilao.cs\*', 'TesteDoAvaliador.cs' (active), 'Avaliador.cs\*', 'Lance.cs\*', and 'Usuario.cs'.
- Editor:** Contains the C# code for 'AULA.TesteDoAvaliador'. The code is as follows:

```
10 {  
11     static void Main(string[] args)  
12     {  
13         Usuario davi = new Usuario("Davi"); //criar os usuarios  
14         Usuario jonatasflash = new Usuario("JonatasFlash");  
15         Usuario isaque = new Usuario("Isaque");  
16  
17         Leilao leilao = new Leilao("Cubo do matheus"); // alimentar com os valores  
18         leilao.Propoe(new Lance(davi, 280.0));  
19         leilao.Propoe(new Lance(jonatasflash, 400.0)); //existe um erro aqui  
20         leilao.Propoe(new Lance(isaque, 500.0));  
21         Avaliador leiloeiro = new Avaliador();  
22         leiloeiro.Avalia(leilao);  
23  
24         Console.WriteLine(leiloeiro.MaiorLance); //mostra no console da tela os valores  
25         Console.WriteLine(leiloeiro.MenorLance);  
26  
27         Console.ReadKey();  
28     }  
29 }
```
- Right Side:** Includes the 'Gerenciador de Soluções' (Solution Explorer) showing the project structure for 'AULA', and the 'Propriedades' (Properties) window.
- Left Side:** A vertical sidebar with the text 'Gerenciador de Servidores'.

# TesteDoAvaliador.cs

The image shows a Visual Studio IDE with the following components:

- Top Bar:** Includes icons for file operations, a dropdown menu with 'Debug' and 'Any CPU', a 'Iniciar' button, and other utility icons.
- Tab Bar:** Displays open files: 'Leilao.cs\*', 'TesteDoAvaliador.cs' (active), 'Avaliador.cs\*', 'Lance.cs\*', and 'Usuario.cs'.
- Editor:** Contains the C# code for 'AULA.TesteDoAvaliador'. The code is as follows:

```
10 {  
11     static void Main(string[] args)  
12     {  
13         Usuario davi = new Usuario("Davi"); //criar os usuarios  
14         Usuario jonatasflash = new Usuario("JonatasFlash");  
15         Usuario isaque = new Usuario("Isaque");  
16  
17         Leilao leilao = new Leilao("Cubo do matheus"); // alimentar com os valores  
18         leilao.Propoe(new Lance(davi, 280.0));  
19         leilao.Propoe(new Lance(jonatasflash, 400.0)); //existe um erro aqui  
20         leilao.Propoe(new Lance(isaque, 500.0));  
21         Avaliador leiloeiro = new Avaliador();  
22         leiloeiro.Avalia(leilao);  
23  
24         Console.WriteLine(leiloeiro.MaiorLance); //mostra no console da tela os valores  
25         Console.WriteLine(leiloeiro.MenorLance);  
26  
27         Console.ReadKey();  
28     }  
29 }
```
- Right Side:**
  - Solution Explorer:** Shows the project structure for 'Solução \'AULA\' (1 projeto)' with a folder 'AULA' containing files: 'Properties', 'Referências', 'App.config', 'Avaliador.cs', 'Lance.cs', 'Leilao.cs', 'TesteDoAvaliador.cs', and 'Usuario.cs'.
  - Properties Window:** Currently empty, with tabs for 'Gerenciador de Soluções' and 'Team Explorer'.

**O ponto é a Máquina testar**

**Porque é chato!**

**Demorado**

**E caro**

**O ponto é a Máquina testar**

**Se forem perceber já escrevemos 3/2  
de um teste automatizado...**

**O ponto é a Máquina testar**

**CENÁRIO...**

**AÇÃO...**

**VALIDAÇÃO...**