

# Testes de mutação com Stryker.NET

## Um pouco sobre a biblioteca

Stryker.NET é uma biblioteca que oferece testes de mutação para .NET, ele permite que você teste seus testes unitários inserindo bugs temporariamente, é importante lembrar que a biblioteca só pode ser usado para aplicações .net 5 e adiante

## Sobre os testes

O teste de mutação ou testes mutantes são feitos para avaliarem a bateria de testes do seu código. O conceito é modificar uma parte do seu código, implementando bugs ou "mutantes" fazendo com que seus testes quebrem e encontrando possíveis falhas ou erros. Seus testes são executados para cada mutante. Se seus testes falharem, o mutante sobrevive. Se seus testes passarem, o mutante é morto. Quanto maior a porcentagem de mutantes mortos, mais eficazes serão seus testes.

Os estados dos mutantes:

- Killed: Quando pelo menos um teste falhou enquanto este mutante estava ativo, o mutante é morto
- Survived: Quando todos os testes passaram enquanto este mutante estava ativo, o mutante sobreviveu.
- No Coverage: O mutante não é coberto por um de seus testes e sobreviveu como resultado.

## Iniciando o projeto

Neste exemplo utilizaremos o xunity para nossos testes, você poderá estar utilizando este [Template](https://github.com/Eliandro-Freitas/TemplateMutationTests.git) com as classes e os testes já implementadas para adiantar o processo ou clonando o projeto diretamente pelo link <https://github.com/Eliandro-Freitas/TemplateMutationTests.git>

Com o template aberto teremos uma classe chamada **Toy** com um método **RecomendadeAge** Temos uma situação que esse brinquedo é recomendado para crianças entre 2 e 10 anos

```
public class Toy
{
    public bool RecomendadeAge(int age)
    {
        if (age >= 2 && age < 10)
            return true;

        return false;
    }
}
```

Apenas com esses dois métodos teremos testados as duas condicionais e os testes irão passar, mas a nossa demanda não será atendida, precisamos que a idade seja entre 2 e 10 anos e aqui está até 9 anos, é apenas um sinal de '=' que as vezes passa despercebido que poderá trazer um BUG na hora de rodar seu código em produção

```
public class ToyTests
{
    [Fact]
    public void ShouldReturnTrueWhenAgeIsRecomendada()
    {
        var result = new Toy().RecomendadaAge(4);
        result.Should().BeTrue();
    }

    [Fact]
    public void ShouldReturnFalseWhenAgeIsNotRecomendada()
    {
        var result = new Toy().RecomendadaAge(15);
        result.Should().BeFalse();
    }
}
```

## Agora vamos adicionar a biblioteca

Com o template em mãos ou no seu projeto de preferência, entre no package manager console do visual studio ou entre na pasta com o cmd ou powerShell onde seu projeto principal foi criado e execute o comando

```
dotnet tool install -g dotnet-stryker
```

## E rodar o Stryker

No windows explorer navegue até onde está salvo o arquivo csproj do seu projeto de testes e digite o comando `dotnet Stryker` e de enter para executar

## Você verá uma janela como essa

```

  /- - - - - \
 /  C  A  N  \
/   N  E  T   \
 \  <  >  \
  - - - - - \

Version: 1.3.1

[21:26:44 INF] Identifying project to mutate.
A new version of Stryker.NET (1.4.0) is available. Please consider upgrading using 'dotnet tool update -g dotnet-stryker'

A preview version of Stryker.NET (1.4.0) is available.
If you would like to try out this preview version you can install it with 'dotnet tool update -g dotnet-stryker --version 1.4.0'
Since this is a preview feature things might not work as expected! Please report any findings on GitHub!

[21:26:46 INF] The project C:\Estudos\TemplateTestesMutacao\Template_Testes_de_mutacao\TemplateMutationTests\Template\Template.csproj will be mutated.
[21:26:48 INF] Analysis complete.
[21:26:48 INF] Building test project C:\Estudos\TemplateTestesMutacao\Template_Testes_de_mutacao\TemplateMutationTests\Tests\Tests.csproj (1/1)
[21:27:02 INF] Total number of tests found: 2.
[21:27:02 INF] Initial testrun started.
[21:27:05 INF] 9 mutants created
[21:27:05 INF] Capture mutant coverage using 'CoverageBasedTest' mode.
[21:27:06 INF] 9 total mutants will be tested

100,00% | Testing mutant 9 / 9 | K 7 | S 2 | T 0 | ~0m 00s |
00:00:08

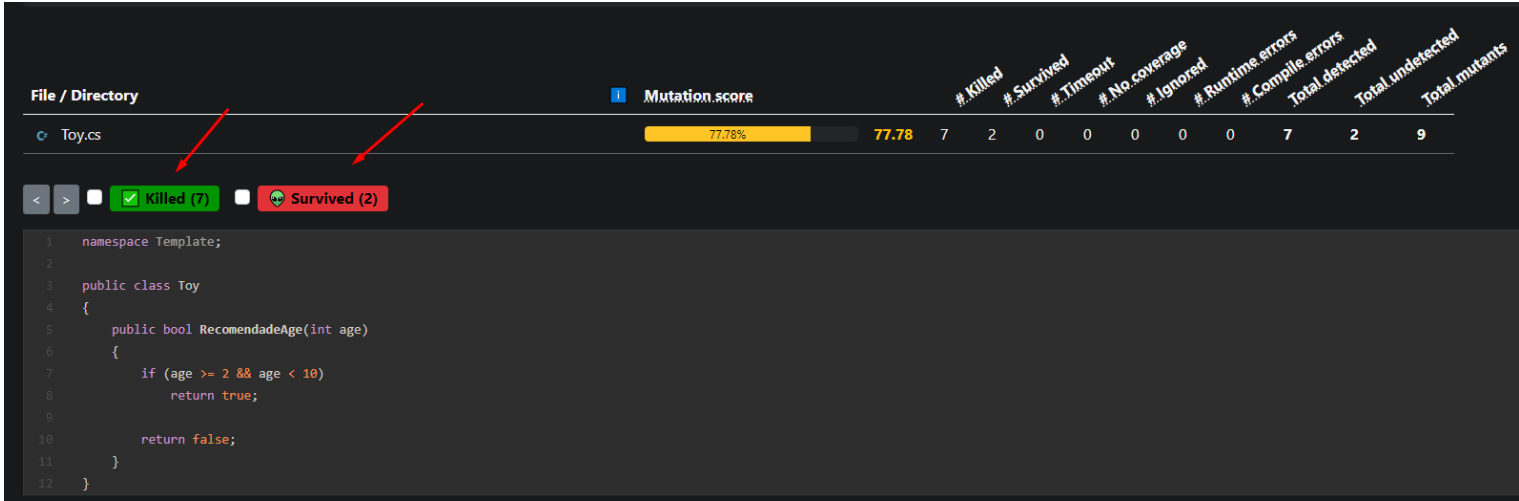
Killed: 7
Survived: 2
Timeout: 0
Hint: by passing "--open-report or -o" the report will open automatically once Stryker is done.
Your html report has been generated at:
file:///C:/Estudos/TemplateTestesMutacao/Template_Testes_de_mutacao/TemplateMutationTests/Tests/StrykerOutput/2022-03-02.21-26-44/reports/mutation-report.html
You can open it in your browser of choice.
[21:27:15 INF] Time Elapsed 00:00:31.1574393
[21:27:15 INF] The final mutation score is 77.78 %
PS C:\Estudos\TemplateTestesMutacao\Template_Testes_de_mutacao\TemplateMutationTests\Tests>
```

Copie a URL gerada e cole no seu browser de preferência para abrir o relatório

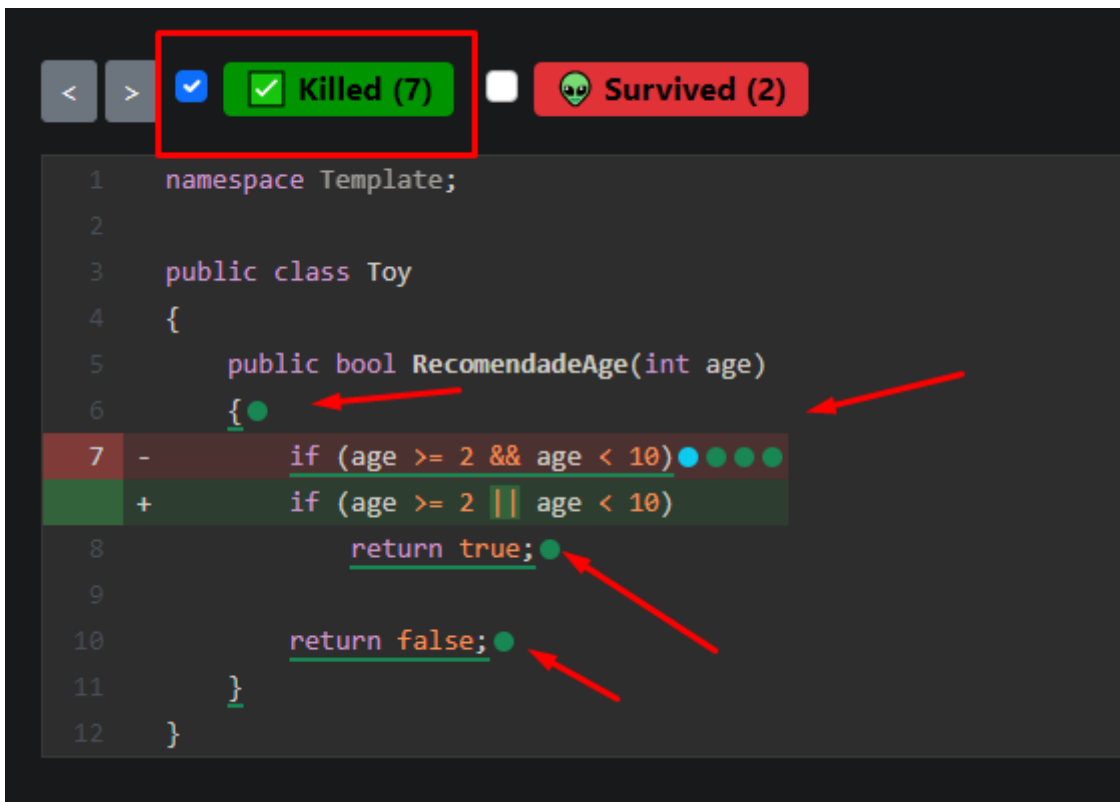
```

Survived: 2
Timeout: 0
Hint: by passing "--open-report or -o" the report will open automatically once Stryker is done.
Your html report has been generated at:
file:///C:/Estudos/TemplateTestesMutacao/Template_Testes_de_mutacao/TemplateMutationTests/Tests/StrykerOutput/2022-03-02.21-37-43/reports/mutation-report.html
You can open it in your browser of choice.
[21:38:08 INF] Time Elapsed 00:00:24.4023989
[21:38:08 INF] The final mutation score is 77.78 %
PS C:\Estudos\TemplateTestesMutacao\Template_Testes_de_mutacao\TemplateMutationTests\Tests>
```

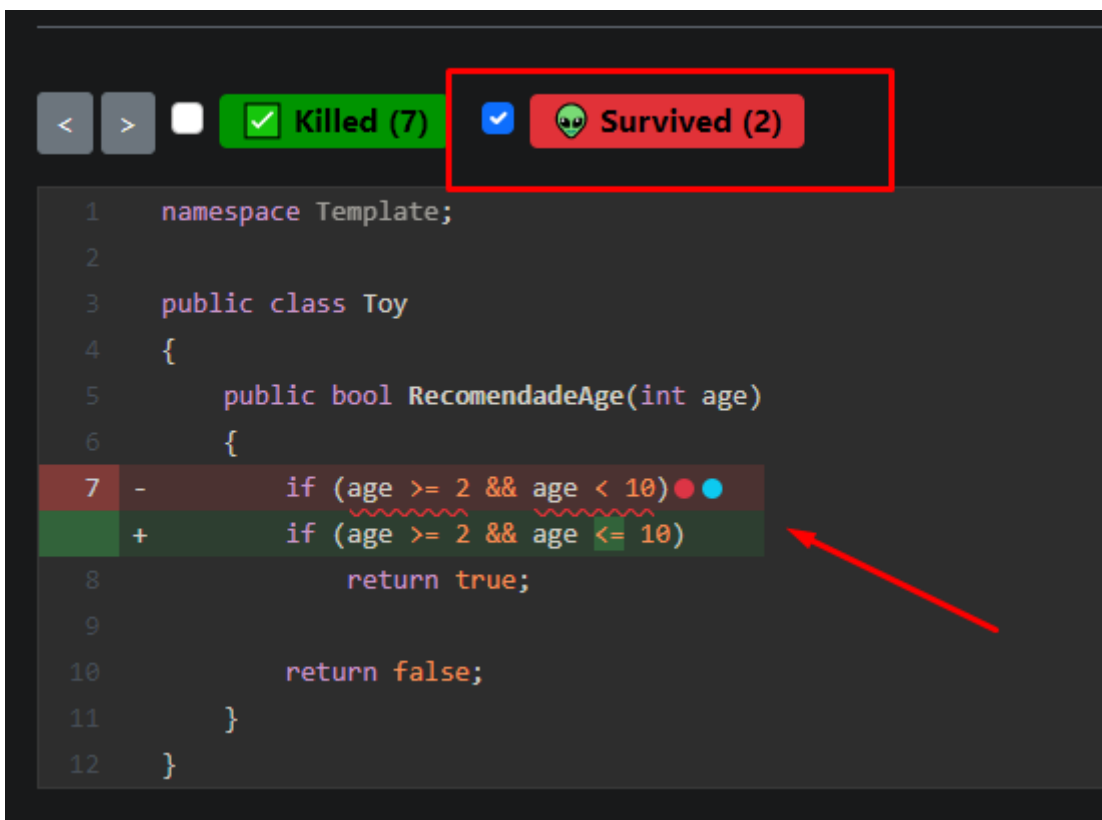
Com o browser aberto teremos nosso relatório mostrando quantos mutantes foram criados, sobreviveram ou morreram com as mutações criadas e também o nosso **score** , o mais importante para nós são a quantidade de mutantes que sobreviveram aos testes, eles são indicativos de testes que estão faltando



Ao selecionar a box de killed poderemos ver todas as alterações que foram feitas no código e que passaram nos nossos testes, cada bolinha verde mostrará uma alteração feita



Selecionando a box de Survived será mostrando todas as alterações feitas e que fizeram os testes falharem e mostrando possíveis alterações para melhorar o código, mas fique atento, nem todas alterações recomendadas atendem nossas necessidades, temos que saber analisar e usar o que for necessário para nossa regra de negócio, assim



Tendo todos esses dados, iremos ajustar nosso método e melhorar os testes Vamos adicionar mais uma condicional no método para que seja aceito a idade de 10 anos também

```
public bool RecomendadeAge(int age)
{
    if (age >= 2 && age <= 10)
        return true;

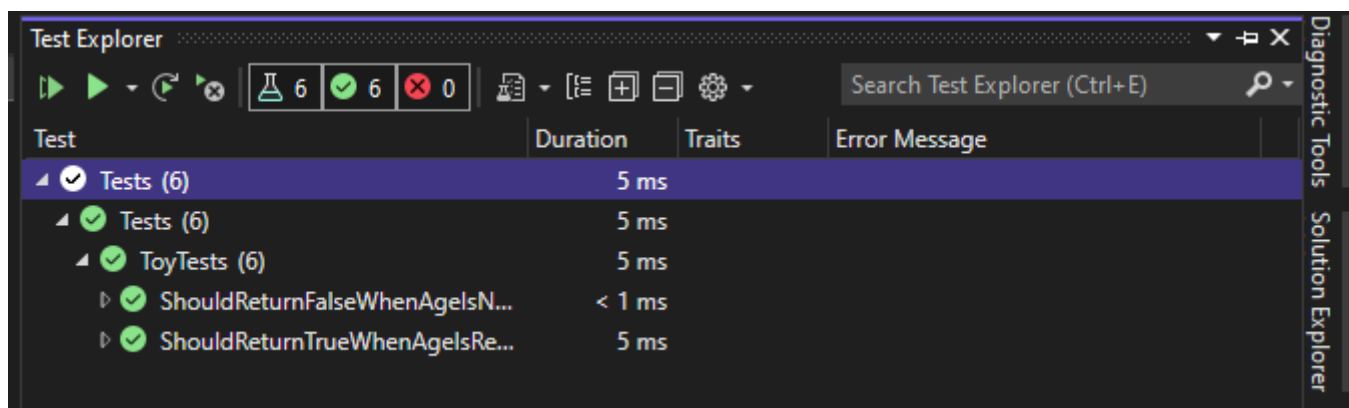
    return false;
}
```

E vamos adicionar alguns parametros nos nossos testes para ele ficar mais dinâmico e cobrir uma quantidade maior de possibilidade

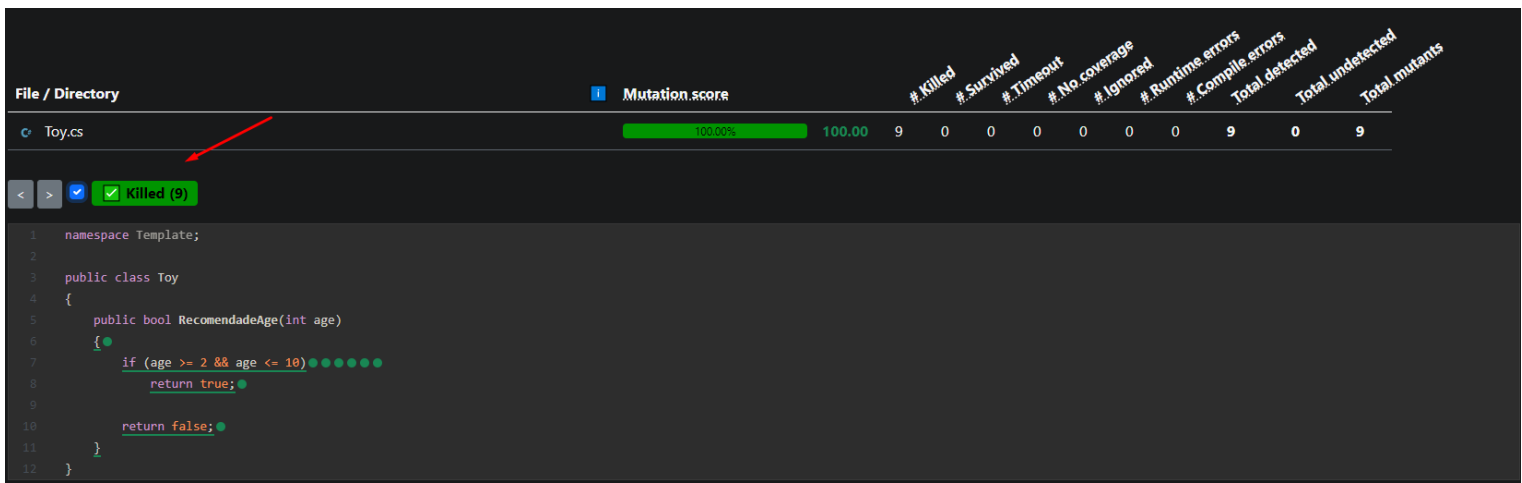
```
[Theory]
[InlineData(2)]
[InlineData(4)]
[InlineData(10)]
public void ShouldReturnTrueWhenAgeIsRecomendade(int age)
{
    var result = new Toy().RecomendadeAge(age);
    result.Should().BeTrue();
}

[Theory]
[InlineData(1)]
[InlineData(11)]
[InlineData(15)]
public void ShouldReturnFalseWhenAgeIsNotRecomendade(int age)
{
    var result = new Toy().RecomendadeAge(age);
    result.Should().BeFalse();
}
```

Vamos rodar nossos testes e ver se todos passam



Os testes continuam passando, agora vamos novamente executar o comando **dotnet Stryker** cmd ou PowerShell na pasta onde está nosso projeto de testes, em seguida copiar a URL gerada e abrir no browser para podermos usar o relatório. Com ele aberto já teremos nosso relatório, mostrando todas as alterações feitas em nosso método e a quantidade que mutantes de sobreviveram ou morreram aos testes



Na imagem acima ja podemos ver que nenhum mutante sobreviveu, todos nossos testes passaram e nosso método está de acordo com nossa regra de negócio

## Considerações finais

Nossos códigos nem sempre vão estar cobertos totalmente pelos testes, os testes de mutação podem nos ajudar a criar novos cenários e a ajudar o programador a prevenir possíveis bugs, aqui tivemos um exemplo simples, apenas para mostrar a biblioteca e falar um pouco sobre os testes de mutação, mas com projetos reais a quantidade de combinações criadas pode crescer muito, com isso criando vários cenários e ajudando o programadores a melhorar os testes, vale lembrar que nem todas as alterações recomendadas quando um mutante sobrevive são as melhores para nossa regra de negócio, devemos saber analisar para utilizar a melhor recomendação

Referências [Stryker.NET](#)

**Espero que tenham gostado e que eu tenha conseguido compartilhar um pouco de conhecimento**