Aula 31

Prof: Henrique Augusto Maltauro

Codificar Back-end de Aplicações Web

NUnit

Como muitas outras ferramentas do .Net, o NUnit possui alguns atributos de configuração, que serão utilizados para definir os processos de testes.

Existem mais de 40 atributos de configuração no NUnit. Todos eles podem ser verificados na documentação do NUnit.

https://docs.nunit.org

NUnit

- [TestFixture]
- [OneTimeSetUp]
- [OneTimeTearDown]
- [Test]
- [SetUp]
- [TearDown]
- [Order]
- [Random]
- [Values]

NUnit

- [TestFixture]

O atributo de configuração [TestFixture] irá demarcar uma classe como um dispositivo de teste, classe essa que deverá possuir os métodos de testes, os quais irão realizar os processos de testes.

NUnit

- [TestFixture]

```
[TestFixture]
public class ClasseDeTeste
{
    //Bloco de código da classe
}
```

NUnit

- [OneTimeSetUp]

O atributo de configuração [OneTimeSetUp] irá demarcar um método de uma classe demarcada com [TestFixture], como um método que será executado uma única vez antes de iniciar os testes daquela classe.

NUnit

- [OneTimeSetUp]

```
[TestFixture]
public class ClasseDeTeste
{
    [OneTimeSetUp]
    public void OneTimeInit()
    {
        //Bloco de código do método
    }
}
```

NUnit

- [OneTimeTearDown]

O atributo de configuração [OneTimeTearDown] irá demarcar um método de uma classe demarcada com [TestFixture], como um método que será executado uma única vez depois de finalizar os testes daquela classe.

NUnit

- [OneTimeTearDown]

```
[TestFixture]
public class ClasseDeTeste
{
    [OneTimeTearDown]
    public void OneTimeClean()
    {
        //Bloco de código do método
    }
}
```

NUnit

- [Test]

O atributo de configuração [Test] irá demarcar um método de uma classe demarcada com [TestFixture], como método de teste, o qual irá realizar os processos de testes.

NUnit

- [Test]

```
[TestFixture]
public class ClasseDeTeste
{
    [Test]
    public void MetodoDeTeste()
    {
        //Bloco de código do método
    }
}
```

NUnit

- [SetUp]

O atributo de configuração [SetUp] irá demarcar um método de uma classe demarcada com [TestFixture], como um método que será executado antes do teste, toda vez que um teste daquela classe for executado.

NUnit

- [SetUp]

```
[TestFixture]
public class ClasseDeTeste
{
    [SetUp]
    public void Init()
    {
        //Bloco de código do método
    }
}
```

NUnit

- [TearDown]

O atributo de configuração [TearDown] irá demarcar um método de uma classe demarcada com [TestFixture], como um método que será executado depois do teste, toda vez que um teste daquela classe for executado.

NUnit

- [TearDown]

```
[TestFixture]
public class ClasseDeTeste
{
    [TearDown]
    public void Clean()
    {
        //Bloco de código do método
    }
}
```

NUnit

- [Order]

O atributo de configuração [Order] irá demarcar um método demarcado com [Test], determinando a ordem de execução dos métodos de testes.

NUnit

- [Order]

```
[TestFixture]
public class ClasseDeTeste
{
    [Test, Order(1)]
    public void MetodoDeTeste1()
    {
        //Bloco de código do método
    }

    [Test, Order(2)]
    public void MetodoDeTeste2()
    {
        //Bloco de código do método
    }
}
```

NUnit

- [Values]

O atributo de configuração [Values] irá demarcar os parâmetros de um método demarcado com [Test], determinando uma sequência de valores bem definidos, que serão utilizados para executar os testes.

NUnit

- [Values]

```
[TestFixture]
public class ClasseDeTeste
{
    [Test]
    public void MetodoDeTeste([Values(3, 6, 9)] int numero)
    {
        //Bloco de código do método
    }
}
```

NUnit

- [Values]

```
[TestFixture]
public class ClasseDeTeste
{
    [Test]
    public void MetodoDeTeste([Values(3, 6, 9)] int numero,
                              [Values("ABC", "DEF")] string texto)
    {
        //Bloco de código do método
    }
}
```