

FluentAssertions (<https://fluentassertions.com/introduction>)

Principais métodos e funcionalidades oferecidos pelo FluentAssertions :

Igualdade: `actual.Should().Be(expected);`

Diferença: `actual.Should().NotBe(expected);`

Verificar se é Nulo ou Não Nulo:

```
actual.Should().BeNull();           // Verifica se é nulo
actual.Should().NotBeNull();        // Verifica se não é nulo
```

Verificar Tipo:

```
actual.Should().BeOfType<ExpectedType>();
// Verifica se é do tipo esperado
```

Comparação com Precisão:

```
actual.Should().BeInRange(low, high);
// Verifica se o valor está dentro de um intervalo
```

Coleções:

```
collection.Should().Contain(expectedItem);
// Verifica se a coleção contém o item esperado
```

```
collection.Should().NotContain(unexpectedItem);
// Verifica se a coleção não contém o item inesperado
```

```
collection.Should().HaveCount(expectedCount);
// Verifica se a coleção tem o tamanho esperado
```

```
collection.Should().OnlyContain(item => item.Condition);
// Verifica se todos os itens atendem a uma condição
```

String:

```
actualString.Should().StartWith("prefixo");
// Verifica se a string começa com o prefixo
```

```
actualString.Should().EndWith("sufixo");
// Verifica se a string termina com o sufixo
```

```
actualString.Should().Contain("substring");
// Verifica se a string contém uma substring
```

```
actualString.Should().Match("regex pattern");
// Verifica se a string corresponde a um padrão de expressão regular
```

Exceções:

```
Action act = () => SomeMethodThrowingException();
act.Should().Throw<ExpectedExceptionType>();
// Verifica se uma exceção do tipo esperado é lançada
```