

```
CalculadoraIMC.cs X ImcTestes.cs
C# CalculadoraIMCNutriVitta CalculadoraIMC

1 using System;
2
3 namespace CalculadoraIMCNutriVitta
4 {
5     2 referências
    public class CalculadoraIMC
6     {
7         10 referências
        public double CalcularIMC(double peso, double altura)
8         {
9             try
10            {
11                double imc = peso / (altura * altura);
12                return imc;
13            }
14            catch (DivideByZeroException e)
15            {
16                Console.WriteLine("Erro ao calcular IMC: " + e.Message);
17                throw e;
18            }
19            catch (Exception e)
20            {
21                Console.WriteLine("Erro desconhecido ao calcular IMC: " + e.Message);
22                throw e;
23            }
24        }
25
26        1 referência
        public string ClassificarIMC(double imc)
27        {
28            if (imc < 18.5)
29            {
30                return "Abaixo do Peso";
31            }
32            else if (imc < 25)
33            {
34                return "Peso Normal";
35            }
36            else if (imc < 30)
37            {
38                return "Sobrepeso";
39            }
40            else if (imc < 35)
41            {
42                return "Obesidade Grau I";
43            }
44            else if (imc < 40)
45            {
46                return "Obesidade Grau II";
47            }
48            else
49            {
50                return "Obesidade Grau III";
51            }
52        }
53    }
54 }
```

69 % Não foi encontrado nenhum problema

PowerShell do Desenvolvedor Lista de Erros Saída

[TestMethod]

0 referências

public void TesteAlturaZero()

{

    peso = 70;

    altura = 0;

    imcCalculado = calculadora.CalcularIMC(peso, altura);

    Assert.ThrowsException<System.DivideByZeroException>(() => calculadora.CalcularIMC(peso, altura));

}

[TestMethod]

0 referências

public void TestePesoInvalido()

{

    peso = -70;

    altura = 1.75;

    imcCalculado = calculadora.CalcularIMC(peso, altura);

    Assert.ThrowsException<System.Exception>(() => calculadora.CalcularIMC(peso, altura));

}

}

```
}
```

```
[TestMethod]
```

```
0 referências
```

```
public void TesteIMCAbaixoDoPeso()
```

```
{
```

```
    peso = 50;
```

```
    altura = 1.65;
```

```
    imcEsperado = 18.37;
```

```
    imcClassificacaoEsperada = "Abaixo do Peso";
```

```
    imcCalculado = calculadora.CalcularIMC(peso, altura);
```

```
    Assert.AreEqual(imcEsperado, imcCalculado, 0.01);
```

```
    Assert.AreEqual(imcClassificacaoEsperada, imcClassificacaoCalculada);
```

```
}
```

```
1  using Microsoft.VisualStudio.TestTools.UnitTesting;
2
3
4  namespace CalculadoraIMCNutriVitta
5  {
6
7      [TestClass]
8      public class ImcTestes
9      {
10         CalculadoraIMC calculadora = new CalculadoraIMC();
11         double peso = 0.0;
12         double altura = 0.0;
13         double imcEsperado = 0.0;
14         double imcCalculado = 0.0;
15         string imcClassificacaoCalculada;
16         string imcClassificacaoEsperada;
17
18         [TestMethod]
19         public void TesteIMCPesoNormal()
20         {
21
22             peso = 70;
23             altura = 1.75;
24             imcEsperado = 22.86;
25             imcClassificacaoEsperada = "Peso Normal";
26
27             imcCalculado = calculadora.CalcularIMC(peso, altura);
28             imcClassificacaoCalculada = calculadora.ClassificarIMC(imcCalculado);
29
30             Assert.AreEqual(imcEsperado, imcCalculado, 0.01);
31             Assert.AreEqual(imcClassificacaoEsperada, imcClassificacaoCalculada);
32         }
33     }
```

[TestMethod]

0 referências

public void TesteIMCSobrepeso()

{

    peso = 85;

    altura = 1.75;

    imcEsperado = 27.76;

    imcClassificacaoEsperada = "Sobrepeso";

    imcCalculado = calculadora.CalcularIMC(peso, altura);

    Assert.AreEqual(imcEsperado, imcCalculado, 0.01);

    Assert.AreEqual(imcClassificacaoEsperada, imcClassificacaoCalculada);

}

```
[TestMethod]
```

```
0 referências
```

```
public void TesteIMCObesidadeGrauI()
```

```
{
```

```
    peso = 100;
```

```
    altura = 1.75;
```

```
    imcEsperado = 32.65;
```

```
    imcClassificacaoEsperada = "Obesidade Grau I";
```

```
    imcCalculado = calculadora.CalcularIMC(peso, altura);
```

```
    Assert.AreEqual(imcEsperado, imcCalculado, 0.01);
```

```
    Assert.AreEqual(imcClassificacaoEsperada, imcClassificacaoCalculada);
```

```
}
```

```
[TestMethod]
```

```
0 referências
```

```
public void TesteIMCObesidadeGrauII()
```

```
{  
    peso = 120;  
    altura = 1.75;  
    imcEsperado = 39.18;  
    imcClassificacaoEsperada = "Obesidade Grau II";  
  
    imcCalculado = calculadora.CalcularIMC(peso, altura);  
  
    Assert.AreEqual(imcEsperado, imcCalculado, 0.01);  
    Assert.AreEqual(imcClassificacaoEsperada, imcClassificacaoCalculada);  
}
```



[TestMethod]

0 referências

public void TesteIMCObesidadeGrauIII()

{

    peso = 150;

    altura = 1.75;

    imcEsperado = 48.98;

    imcClassificacaoEsperada = "Obesidade Grau III";

    imcCalculado = calculadora.CalcularIMC(peso, altura);

    Assert.AreEqual(imcEsperado, imcCalculado, 0.01);

    Assert.AreEqual(imcClassificacaoEsperada, imcClassificacaoCalculada);

}

[TestMethod]