

Resumo de Testes

NutriVitta

Responsável: Thiago Leonardo R. de Freitas

Instituição: SENAI / SP – EAD

Santo André / SP

2022

Sumário

1. Introdução.....	3
2. Objetivo.....	3
3. Cálculos de IMC e Classificação de IMC	3
4. Resultado de Teste.....	3
5. Desenhos de Teste.....	4
5.1 NutriVitta - CalculadoraImc.cs.....	4
5.2 TesteXnitImc – UnitTest1.cs	5
6. Ferramentas e Sistemas.....	5

1. Introdução

A finalidade desse relatório de testes é reunir os resultados dos testes efetuados na calculadora de IMC.

2. Objetivo

Descrever e apresentar os resultados dos testes para cálculos de índice de massa corpórea (IMC), contemplando:

- Cálculo de IMC;
- Classificação de IMC.

3. Cálculos de IMC e Classificação de IMC

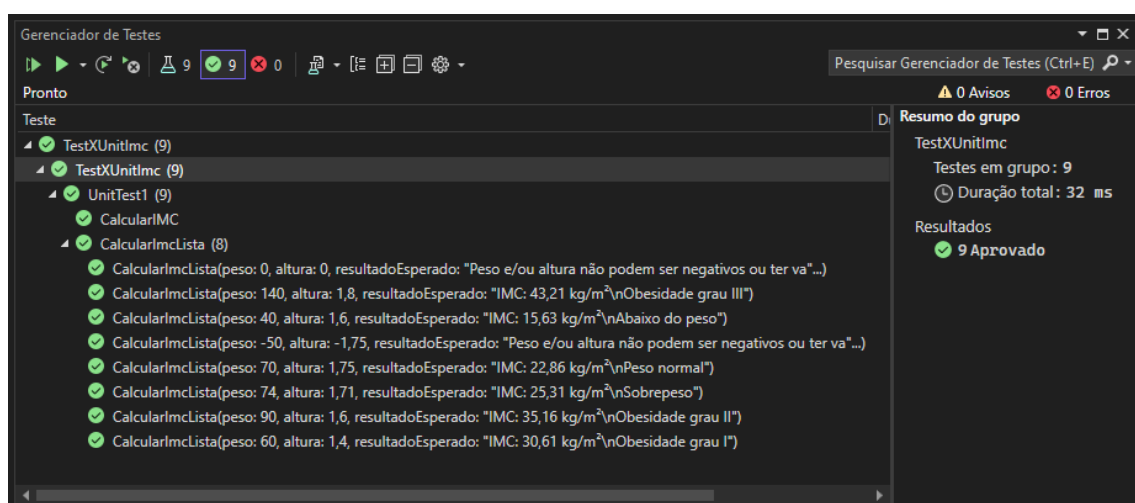
Verificação se os dados foram inseridos corretamente. O sistema não deve aceitar números negativos, número zero e números elevados fora do comum.

Validar se a função executa os cálculos de forma correta considerando a métrica $IMC = peso / (altura \times altura)$. Após o resultado, verificar a faixa e retornar à avaliação final.

Todos os testes foram executados assertivamente, vide tópico 4.

4. Resultado de Teste

Todos os testes foram executados assertivamente:



5. Desenhos de Teste

Os testes unitários e de função foram feitos da seguinte forma:

5.1 NutriVitta - CalculadoraImc.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace NutriVitta
{
    public static class CalculadoraImc
    {
        public static string ClassificacaoIMC(double peso, double altura)
        {
            if (peso <= 0 || altura <= 0)
            {
                return "Peso e/ou altura não podem ser negativos ou ter valor igual a zero.";
            }

            double imc = peso / (altura * altura);
            string classificacao = "";

            if (imc < 18.5)
            {
                classificacao = "Abaixo do peso";
            }
            else if (imc < 25)
            {
                classificacao = "Peso normal";
            }
            else if (imc < 30)
            {
                classificacao = "Sobrepeso";
            }
            else if (imc < 35)
            {
                classificacao = "Obesidade grau I";
            }
            else if (imc < 40)
            {
                classificacao = "Obesidade grau II";
            }
            else if (imc >= 40)
            {
                classificacao = "Obesidade grau III";
            }

            return $"IMC: {imc:0.00} kg/m²\n{classificacao}";
        }
    }
}
```

5.2 TesteXnitImc – UnitTest1.cs

```
using NutriVitta;
using System;
using Xunit;

namespace TestXUnitImc
{
    public class UnitTest1
    {
        [Fact]
        public void CalcularIMC()
        {
            //Arrange
            double peso = 0;
            double altura = 0;
            string resultadoEsperado = "Peso e/ou altura não podem ser negativos ou ter valor igual a zero.";
            //Act
            var resultadoObtido = CalculadoraImc.ClassificacaoIMC(peso, altura);

            //Assert
            Assert.Equal(resultadoEsperado, resultadoObtido);
        }

        [Theory]
        [InlineData(-50, -1.75, "Peso e/ou altura não podem ser negativos ou ter valor igual a zero.")]
        [InlineData(0, 0, "Peso e/ou altura não podem ser negativos ou ter valor igual a zero.")]
        [InlineData(40, 1.60, "IMC: 15,63 kg/m²\nAbaixo do peso")]
        [InlineData(70, 1.75, "IMC: 22,86 kg/m²\nPeso normal")]
        [InlineData(74, 1.71, "IMC: 25,31 kg/m²\nSobrepeso")]
        [InlineData(60, 1.40, "IMC: 30,61 kg/m²\nObesidade grau I")]
        [InlineData(90, 1.60, "IMC: 35,16 kg/m²\nObesidade grau II")]
        [InlineData(140, 1.80, "IMC: 43,21 kg/m²\nObesidade grau III")]
        public void CalcularImcLista(double peso, double altura, string resultadoEsperado)
        {
            var resultadoObtido = CalculadoraImc.ClassificacaoIMC(peso, altura);

            Assert.Equal(resultadoEsperado, resultadoObtido);
        }
    }
}
```

6. Ferramentas e Sistemas

Serão empregadas as seguintes ferramentas:

Sistema Operacional	Windows 10
Planejamento	Microsoft
Desenho e execução	Visual Studio
	xUnit