

Plano de Testes e Casos de Testes

Empresa NutriVitta

Aluno: Marlon Mendonça

Plano de Testes

1. Introdução

A finalidade do *Plano de Teste* é reunir todas as informações necessárias para o planejamento de testes referentes a uma iteração específica.

Neste plano serão testadas algumas funcionalidades do sistema de Calculadora IMC da Empresa NutriVitta, visando avaliar se elas se comportam de acordo com o esperado.

2. Objetivos

Este documento tem por objetivo criar um plano de testes para o sistema de Calculadora IMC da empresa NutriVitta, cobrindo as seguintes atividades:

- Executar testes de unidade e integração;
- Verificar o comportamento da funcionalidade de cálculo do IMC;
- Verificar o comportamento da funcionalidade de classificação do IMC de acordo com a tabela Abeso;
- Identificar possíveis erros no uso da aplicação;
- Sugerir a implementação de melhorias de acordo com os resultados dos testes

3. Escopo Geral

Os seguintes itens serão testados:

- O cálculo do IMC.
- A classificação do resultado de acordo com os parâmetros estabelecidos na tabela da Abeso.

4. Requisitos e Estratégia de Testes

Os testes verificarão as funcionalidades da aplicação, certificando-se de que elas apresentam o comportamento esperado e/ou identificando possíveis erros do sistema, através da simulação com dados fictícios. Neste caso, testaremos as funcionalidades de cálculo e classificação.

5. Caso de Testes

Os testes serão escritos na linguagem C#. Os elementos testados são descritos a seguir:

- **Caso de Testes 1:** Calcular o IMC do usuário

Procedimento: Utilizar o cálculo $IMC = peso / (altura \times altura)$ com os dados fictícios de peso = 79,0 kg e altura = 1,72 m

Resultados: o resultado esperado será 26,7

- **Caso de Testes 2:** Classificar o resultado do cálculo de IMC de acordo com os valores da tabela Abeso

Procedimento: com o resultado do cálculo de IMC verificar em qual faixa da tabela ele se encontra e retornar os valores correspondentes.

Resultados: Retorna a mensagem “Sobrepeso”

6. Desenho de Testes

Calculo.cs

```
namespace CalculoIMC
{
    public static class Calculos
    {
        public static double CalcularIMC(double peso_kg, double altura_cm)
        {
            return peso_kg / (altura_cm * altura_cm);
        }

        public static string ClassificarIMC(double IMC)
        {
            var Classificacao = "";

            if (IMC < 18.5)
            {
                Classificacao = "Abaixo do peso";
            }

            else if (IMC >= 18.5 && IMC <= 24.9)
            {
                Classificacao = "Peso Normal";
            }

            else if (IMC >= 25.0 && IMC <= 29.9)
```

```
        {
            Classificacao = "Sobrepeso";
        }
        else if (IMC >= 30.0 && IMC <= 34.9)
        {
            Classificacao = "Obesidade Grau I";
        }
        else if (IMC >= 35.0 && IMC <= 39.9)
        {
            Classificacao = "Obesidade Grau II";
        }
        else
        {
            Classificacao = "Obesidade Grau III";
        }

        return Classificacao;
    }
}
```

```
UnitTest1.cs
using CalculoIMC;

namespace TestProject1
{
    public class UnitTest1
    {
        [Fact]
        public void CalcularIMC()
        {
            //Arrange
```

```
        double peso_kg = 79.0;

        double altura_cm = 1.72;

        string total = "26,7";

        //Act

        var resultado = Calculos.CalcularIMC(peso_kg,
altura_cm).ToString("#.#");

        //Assert

        Assert.Equal(total, resultado);
    }

    [Fact]
    public void ClassificarIMC()
    {
        //Arrange

        double IMC = 26.7;

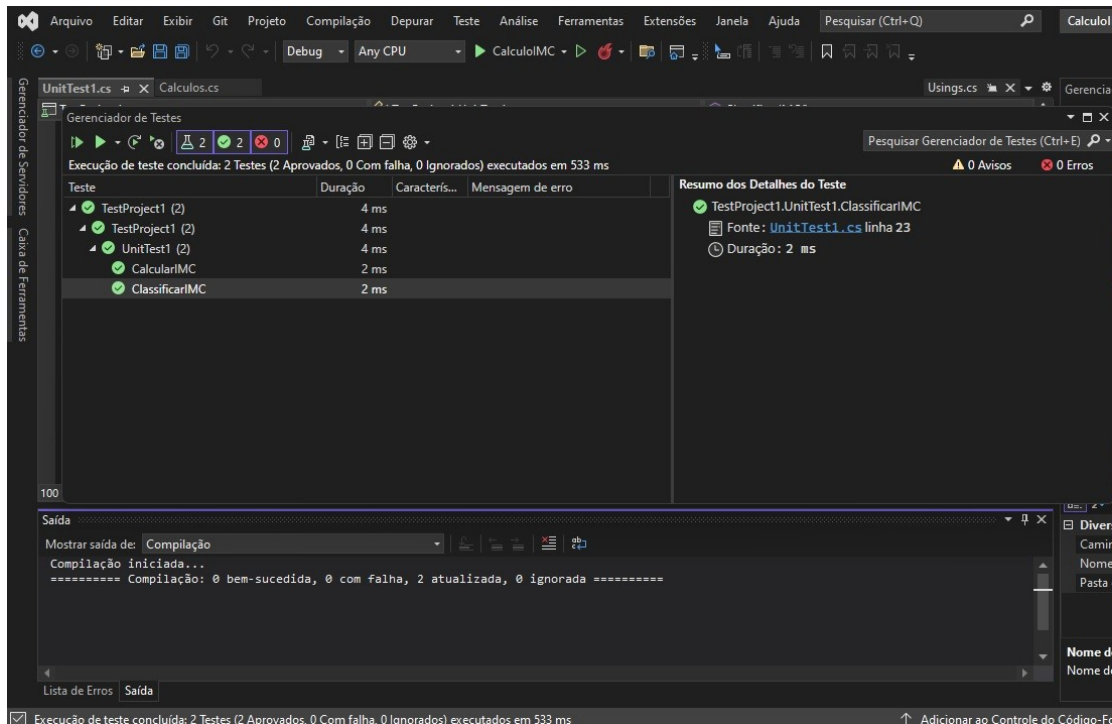
        string result = "Sobrepeso";

        //Act

        var resultado = Calculos.ClassificarIMC(IMC);

        //Assert

        Assert.Equal(result, resultado);
    }
}
```



7. Relatório de Testes

- **Caso de Testes 1:** Calcular o IMC do usuário
Resultado: Passou
- **Caso de Testes 2:** Classificar o resultado do cálculo de IMC de acordo com os valores da tabela Abeso
Resultado: Passou

8. Resumo de Testes:

Os testes foram elaborados pela aluna Mayara Ramos Saldanha. Foram executados dois (02) casos de teste e ambos tiveram o comportamento esperado, retornando erros apenas quando previstos. Todos os testes passaram.

9. Ferramentas

São necessários os seguintes elementos de software básicos no ambiente de teste deste *Plano de Teste*:

- SO Windows 10;
- Microsoft Visual Studio 2022;
- Xunit;
- Linguagem C#;

10. Equipe

Marlon Trindade Lopes Mendonça

11. Normas Utilizadas

- IEE 829-2008
- ISO/IEC 25010:2017

12. Cronograma

- Planejamento dos testes: 22 /06/2022;
- Execução: 09/06/2022;
- Avaliação: 09/06/2022;

13. Referências

Microsoft Visual Studio. Disponível em: <https://visualstudio.microsoft.com/pt-br/downloads/>. Acesso em: 02 de Jun. de 2022.

ISO/IEC 25010:2011. Disponível em: <https://www.iso.org/standard/35733.html>. Acesso em: 02 de Jun. de 2022.

IEEE 829-2008. Disponível em: <https://www.ieee.org/>. Acesso em: 02 de Jun. de 2022.

Xunit. Disponível em: <https://xunit.net/>. Acesso em: 02 de Jun. de 2022.