

PLANO DE TESTE DO APLICATIVO PARA O CÁLCULO DO IMC DA NUTRIVITA

Preparado por:

Daniel Monteiro

10/10/2022

TABELA DE CONTEÚDOS:

- 1 INTRODUÇÃO
- 2 OBJETIVOS
- 3 ESCOPO
- 4 REQUISITOS DE TESTE
- 5 ESTRATÉGIA DE TESTE
- 6 REQUISITOS DE HARDWARE E SOFTWARE
- 7 EQUIPE
- 8 CRONOGRAMA
- 9 DESENHO DO TESTE

1. INTRODUÇÃO

Este documento planeja o teste do aplicativo para o cálculo do IMC da NutriVitta.

2. OBJETIVOS

O objetivo do teste é verificar se o cálculo do IMC e a sua correspondente classificação, conforme a tabela da Abeso, estão ocorrendo de modo correto.

3. ESCOPO

O aplicativo é composto pelas seguintes regras pré-estabelecidas:

- Realização do cálculo do IMC;
- Classificação do IMC de acordo com a tabela da Abeso.

O Índice de Massa Corporal (IMC) é reconhecido como padrão internacional para avaliar o grau de obesidade. O IMC é calculado dividindo o peso (em kg) pela altura ao quadrado (em metros), organizado conforme a tabela:

Quadro 1 – Classificação do IMC (2017).

Categoria	IMC
Abaixo do peso	Abaixo de 18,5
Peso normal	18,5 - 24,9
Sobrepeso	25,0 - 29,9
Obesidade Grau I	30,0 - 34,9
Obesidade Grau II	35,0 - 39,9
Obesidade Grau III	40,0 e acima

Fonte: Abeso

4. REQUISITOS DE TESTE

O requisito a ser testado é o sistema de cálculo e classificação do IMC. Nesse sentido, serão adotados os seguintes casos:

Casos de Teste

a) Caso 1

No aplicativo, insere-se a altura 1,70 metros e o peso 100 kg. O resultado da divisão entre o peso e o quadrado da altura deve ser igual a 34,60. A classificação correspondente, conforme a tabela da Abeso, deve ser Obesidade Grau I.

b) Caso 2

No aplicativo, insere-se a altura 1,80 metros e o peso 90 kg. O resultado da divisão entre

o peso e o quadrado da altura deve ser igual a 27,77. A classificação correspondente, conforme a tabela da Abeso, deve ser Sobrepeso.

c) Caso 3

No aplicativo, insere-se a altura 1,60 metros e o peso 60 kg. O resultado da divisão entre o peso e o quadrado da altura deve ser igual a 23,43. A classificação correspondente, conforme a tabela da Abeso, deve ser Peso normal.

5. ESTRATÉGIA DE TESTE

Será adotado, para cada caso em questão, o Teste unitário. Tal teste será conduzido pelo setor responsável, que terá a tarefa de verificar se o aplicativo está funcionando corretamente e de reportar eventuais problemas.

6. REQUISITOS DE HARDWARE E SOFTWARE

Desktop ou notebook com sistema operacional Windows e Visual Studio 2019.

7. EQUIPE

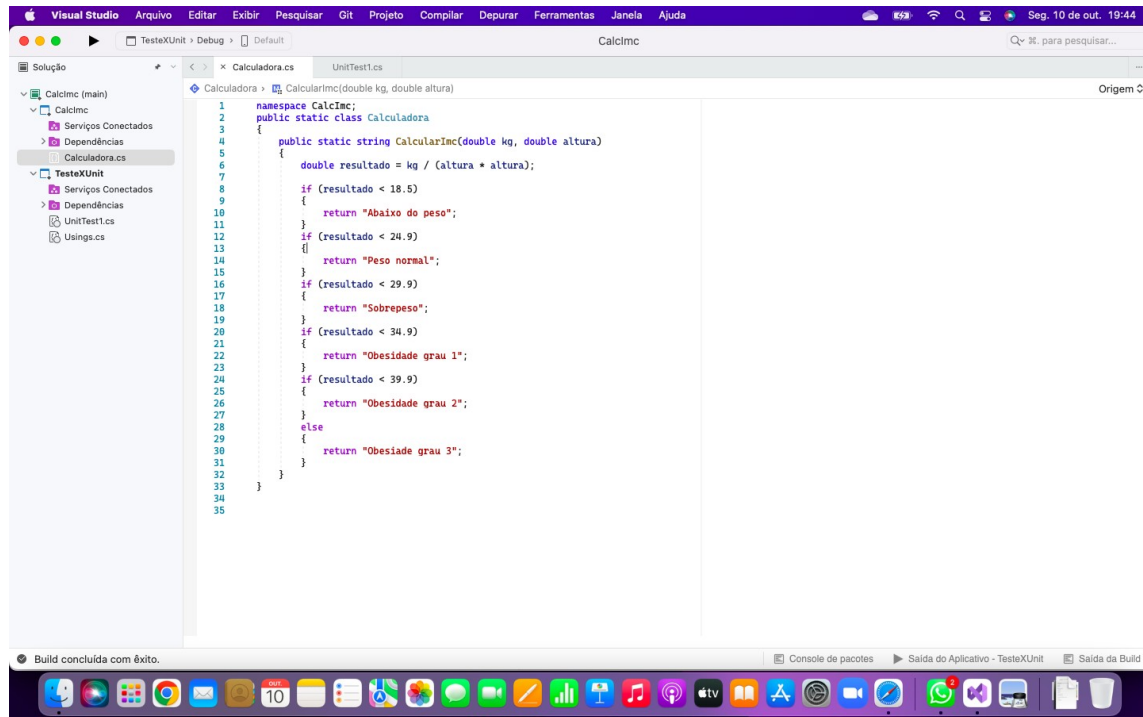
Setor responsável pelos testes.

8. CRONOGRAMA

TAREFA	COMEÇAR	ACABAR
Desenho do teste	05/10/2022	06/10/2022
Execução do teste do Caso 1	07/10/2022	08/10/2022
Execução do teste do Caso 2	07/10/2022	08/10/2022
Execução do teste do Caso 3	07/10/2022	08/10/2022
Elaboração de documentos e resultados	09/10/2022	10/10/2022

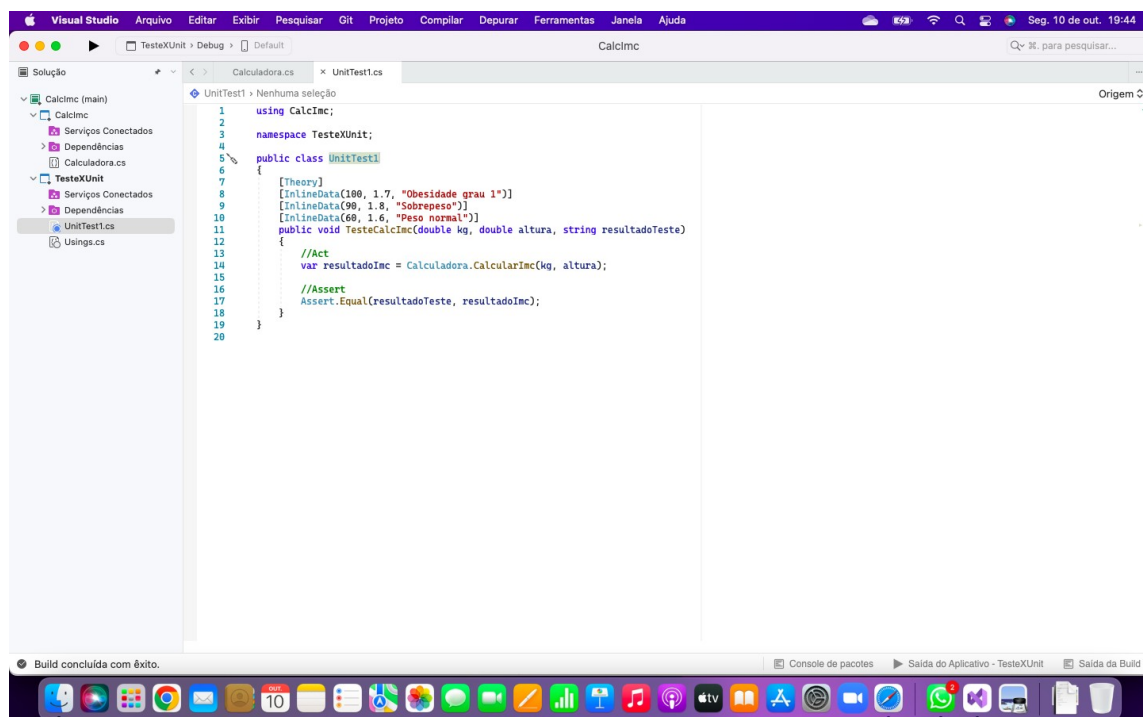
9. DESENHO DO TESTE

A codificação para testar a aplicação, com base nos casos de teste listados, utilizando a linguagem C#, é a seguinte (também disponível em <https://github.com/danieldmonteiro/senai-uc15/tree/main/CalcImc>):



The screenshot shows the Visual Studio IDE with the 'Calculadora.cs' file open. The code implements the `CalcImc` namespace and the `Calculadora` class. The `CalcularImc` method takes weight (`kg`) and height (`altura`) as parameters and returns a string representing the BMI category. The logic uses a series of if-else statements to categorize the BMI value into 'Abaixo do peso', 'Peso normal', 'Sobrepeso', 'Obesidade grau 1', 'Obesidade grau 2', or 'Obesidade grau 3'.

```
1 namespace CalcImc;
2 public static class Calculadora
3 {
4     public static string CalcularImc(double kg, double altura)
5     {
6         double resultado = kg / (altura * altura);
7
8         if (resultado < 18.5)
9         {
10             return "Abaixo do peso";
11         }
12         if (resultado < 24.9)
13         {
14             return "Peso normal";
15         }
16         if (resultado < 29.9)
17         {
18             return "Sobrepeso";
19         }
20         if (resultado < 34.9)
21         {
22             return "Obesidade grau 1";
23         }
24         if (resultado < 39.9)
25         {
26             return "Obesidade grau 2";
27         }
28         else
29         {
30             return "Obesidade grau 3";
31         }
32     }
33 }
34
35
```



The screenshot shows the Visual Studio IDE with the 'UnitTest1.cs' file open. The code implements the `UnitTest1` class using the `TesteXUnit` namespace. It includes three data-driven test cases using the `[Theory]` attribute and `[InlineData]` attributes. The `TesteCalcImc` method calls the `Calculadora.CalcularImc` method and asserts that the result matches the expected value using `Assert.Equal`.

```
1 using CalcImc;
2
3 namespace TesteXUnit;
4
5 public class UnitTest1
6 {
7     [Theory]
8     [InlineData(100, 1.7, "Obesidade grau 1")]
9     [InlineData(90, 1.8, "Sobrepeso")]
10    [InlineData(60, 1.6, "Peso normal")]
11    public void TesteCalcImc(double kg, double altura, string resultadoTeste)
12    {
13        //Act
14        var resultadoImc = Calculadora.CalcularImc(kg, altura);
15
16        //Assert
17        Assert.Equal(resultadoTeste, resultadoImc);
18    }
19
20 }
```