

Planejamento de Teste

NutriVitta –
Calculadora de IMC

Histórico da Revisão

Data	Versão	Descrição	Autor
25/08/2022	1.0	--	Igor Nunes

Índice:

1.	Resumo.....	4
2.	Introdução.....	4
2.1	Introdução.....	4
2.2	Objetivos.....	4
3.	Escopo Geral.....	5
3.1	Elementos que serão testados.....	5
3.2	Requisitos de testes.....	5
3.3	Casos de teste.....	6
3.3.1	Teste do método QuadradoAltura.....	6
3.3.2	Teste do método CalcularIMC.....	7
4.	Estratégia de Testes.....	9
4.1	Técnica.....	9
4.2	Explicação Detalhada.....	9
4.3	Recursos Utilizados.....	9
4.3.1	Ferramentas.....	10
4.3.2	Sistema.....	10
5.	Desenho de Testes.....	10
6.	Cronograma das Atividades.....	13
7.	Relatório de Testes.....	14
7.1	Teste de Unidade.....	14
7.2	Teste de Integração.....	15
7.3	Responsáveis.....	16
8.	Normas e Referências Utilizadas.....	17

1. Resumo

Esse documento tem por objetivo apresentar o planejamento de teste para determinados elementos do código-fonte, que serão devidamente apresentados e descritos nas seguintes seções do projeto da Calculadora de IMC, um sistema em desenvolvimento da empresa NutriVitta que visa o cálculo do IMC (índice de massa corporal) para a avaliação do grau de obesidade de um indivíduo.

2. Introdução

2.1 Introdução:

A NutriVitta é uma empresa atuante no ramo da nutrição prestando atendimento clínico para seus clientes e tem como objetivo, para poder garantir a melhor realização dessa tarefa, o desenvolvimento de uma calculadora de IMC, uma aplicação que vise o cálculo do IMC. Com a intenção de produzir um sistema com a menor possibilidade de apresentação de erros, foram traçadas rotas de testes para as funcionalidades da aplicação ao longo de seu desenvolvimento, que serão apresentadas e discutidas por meio deste documento.

2.2 Objetivos:

Esse documento de Plano de Teste compõe-se dos seguintes objetivos:

- Identificar informações de projeto existentes e os componentes de software que devem ser testados;
- Listar os Requisitos a testar recomendados;
- Recomendar e descrever as estratégias de teste a serem empregadas;
- Identificar os recursos necessários e prover uma estimativa dos esforços de teste;
- Listar os elementos resultantes do projeto de testes.

3. Escopo Geral

Os testes realizados na aplicação da Calculadora de IMC da empresa NutriVitta, serão dos tipos de teste de unidade e de integração sobre os métodos existentes no sistema, verificando se as contas realizadas pelos usuários apresentarão resultados satisfatórios e corretos, seguindo a classificação de obesidade que pode ser observado no quadro a seguir:

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

3.1 Elementos que serão testados:

- Método QuadradoAltura;
- Método CalcularIMC.

3.2 Requisitos de testes:

Será considerado como aprovado sem erros o(s) seguinte(s) fluxo(s) de teste(s):

- Valores pré-determinados serão passados para o método e a conta deverá ser realizada retornando o resultado que siga a classificação do Quadro 1, sem apresentação de erro qualquer.

3.3 Casos de teste:

A seguir, serão apresentados os casos de teste, sendo demonstrado o fluxo básico e os resultados esperados:

3.3.1 Teste do método QuadradoAltura:

Método TesteQuadradoAltura:

1. Fluxo:
 - a. Entrada de dados: altura (1,80) e resultado esperado (3,24);
 - b. Método QuadradoAltura faz a conta e retorna o resultado real;
 - c. O resultado esperado é comparado com o resultado real;
2. Resultados esperados:
 - a. Caso o resultado esperado seja igual ao resultado real, o teste será aprovado;
 - b. Caso o resultado esperado seja diferente do resultado real, o teste falhará.

Método TesteQuadradoAlturaLista:

1. Fluxo:
 - a. Entrada de conjunto de dados: alturas e resultados esperados;
 - b. O primeiro conjunto de dados é testado: altura (1,70) e resultado esperado (2,89);
 - c. Método QuadradoAltura faz a conta e retorna o resultado real;
 - d. O resultado esperado é comparado com o resultado real;
 - e. O segundo conjunto de dados é testado: altura (1,60) e resultado esperado (2,56);
 - f. Método QuadradoAltura faz a conta e retorna o resultado real;
 - g. O resultado esperado é comparado com o resultado real;
 - h. O terceiro conjunto de dados é testado: altura (1,90) e resultado esperado (3,61);
 - i. Método QuadradoAltura faz a conta e retorna o resultado real;
 - j. O resultado esperado é comparado com o resultado real;
2. Resultados esperados:
 - a. Caso o resultado esperado seja igual ao resultado real para cada conjunto de dados, o teste será aprovado;

- b. Caso o resultado esperado seja diferente do resultado real para cada conjunto de dados, o teste falhará.

3.3.2 Teste do método CalcularIMC:

Método TesteCalcularIMC:

1. Fluxo:
 - a. Entrada de dados: altura em metros (1,80), peso em quilogramas (70,0) e resultado esperado (Categoria: Peso normal);
 - b. Método CalcularIMC chama o método QuadradoAltura e faz as contas necessárias e retorna o resultado real em formato de string;
 - c. É feita uma comparação do resultado esperado com o resultado real, para ver se o real contém o esperado;
2. Resultados esperados:
 - a. Caso o resultado esperado esteja contido na string do resultado real, o teste será aprovado;
 - b. Caso o resultado esperado não esteja contido na string do resultado real, o teste falhará.

Método TesteCalcularIMCLista:

1. Fluxo:
 - a. Entrada de conjunto de dados: altura (em metros), peso (em quilogramas) e resultado esperado;
 - b. O primeiro conjunto de dados é testado: altura (1,80), peso (50,0) e resultado esperado (Categoria: Abaixo do peso);
 - c. Método CalcularIMC chama o método QuadradoAltura e faz as contas necessárias e retorna o resultado real em formato de string;
 - d. É feita uma comparação do resultado esperado com o resultado real, para ver se o real contém o esperado;
 - e. O segundo conjunto de dados é testado: altura (1,80), peso (90,0) e resultado esperado (Categoria: Sobrepeso);
 - f. Método CalcularIMC chama o método QuadradoAltura e faz as contas necessárias e retorna o resultado real em formato de string;

- g. É feita uma comparação do resultado esperado com o resultado real, para ver se o real contém o esperado;
- h. O terceiro conjunto de dados é testado: altura (1,80), peso (110,0) e resultado esperado (Categoria: Obesidade Grau I);
- i. Método CalcularIMC chama o método QuadradoAltura e faz as contas necessárias e retorna o resultado real em formato de string;
- j. É feita uma comparação do resultado esperado com o resultado real, para ver se o real contém o esperado;
- k. O quarto conjunto de dados é testado: altura (1,80), peso (120,0) e resultado esperado (Categoria: Obesidade Grau II);
- l. Método CalcularIMC chama o método QuadradoAltura e faz as contas necessárias e retorna o resultado real em formato de string;
- m. É feita uma comparação do resultado esperado com o resultado real, para ver se o real contém o esperado;
- n. O quinto conjunto de dados é testado: altura (1,80), peso (140,0) e resultado esperado (Categoria: Obesidade Grau III);
- o. Método CalcularIMC chama o método QuadradoAltura e faz as contas necessárias e retorna o resultado real em formato de string;
- p. É feita uma comparação do resultado esperado com o resultado real, para ver se o real contém o esperado;
- q. O sexto conjunto de dados é testado: altura (0), peso (120,0) e resultado esperado (Altura inválida!);
- r. Método CalcularIMC chama o método QuadradoAltura e faz as contas necessárias e retorna o resultado real em formato de string;
- s. É feita uma comparação do resultado esperado com o resultado real, para ver se o real contém o esperado;
- t. O primeiro conjunto de dados é testado: altura (-1,80), peso (120,0) e resultado esperado (Opa! Ocorreu um erro!);
- u. Método CalcularIMC chama o método QuadradoAltura e faz as contas necessárias e retorna o resultado real em formato de string;
- v. É feita uma comparação do resultado esperado com o resultado real, para ver se o real contém o esperado;

2. Resultados esperados:

- a. Caso o resultado esperado esteja contido na string do resultado real para cada conjunto de dados, o teste será aprovado;
- b. Caso o resultado esperado não esteja contido na string do resultado real para cada conjunto de dados, o teste falhará.

4. Estratégia de Testes

4.1 Técnica:

Será utilizada a técnica estrutural (caixa-branca) para a avaliação e validação do código fonte da aplicação e dos métodos escolhidos. Serão analisados o fluxo e a conclusão do teste para verificar se ocorreu/ocorrerá ou não alguma anormalidade durante o processo com o auxílio de um projeto de automação na linguagem C# e outras ferramentas que simularão a entrada de dados para o sistema pelo usuário final.

4.2 Explicação detalhada:

Serão passados valores pré-determinados como parâmetros para os métodos existente da classe Calculadora e os valores esperados para cada operação (ou outros tipos de dados para confirmação, como um verdadeiro ou falso). Realizada a conta com os parâmetros de entrada, o resultado retornado pelo método será comparado com o valor esperado. Caso o valor esperado e o resultado retornado sejam diferentes, o teste falhará. Caso ambos sejam iguais, o teste terá sucesso.

4.3 Recursos utilizados:

Essa seção tem como objetivo apresentar as ferramentas e tecnologias utilizadas na realização do planejamento, execução e avaliação do teste.

4.3.1 Ferramentas:

As seguintes ferramentas são utilizadas nesse projeto:

	Ferramenta	Proprietário
Planejamento de Teste	Microsoft Office Word	Microsoft
Desenho de Teste	Visual Studio 2022	Microsoft
Execução de Teste	.NET 6.0	Microsoft

4.3.2 Sistema:

A tabela a seguir mostra as condições do computador utilizado e alguns complementos necessários para a realização do projeto:

Sistema	Complementos
Sistema operacional Windows 10 com processador Intel Core i7, memória RAM de 16GB e 520GB de armazenamento SSD.	Além das ferramentas citadas na seção anterior, foram utilizados os projetos de testes MSTest e o XUnit para a preparação, execução e verificação dos testes.

5. Desenho de Testes

Nesta seção será apresentado o código desenvolvido para a realização dos testes. O primeiro código fará referência ao teste do método QuadradoAltura, com a utilização com projeto de teste MSTest:

```

using CalculadoraIMCPFS;
using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace CalculadoraIMCMSTest
{
    [TestClass]
    public class TesteCalculadoraIMCMSTest
    {
        [TestMethod]
        public void TesteQuadradoAltura()
        {
            // Arrange - Preparação
            double altura = 1.80;
            double resultadoEsperado = 3.24;

            // Act - Ação
            var resultado = Calculadora.QuadradoAltura(altura);

            // Assert - Verificação
            Assert.AreEqual(resultadoEsperado, resultado);
        }

        [DataTestMethod]
        [DataRow(1.7, 2.89)]
        [DataRow(1.6, 2.56)]
        [DataRow(1.9, 3.61)]
        public void TesteQuadradoAlturaLista(double altura,
double resultadoEsperado)
        {
            // Act - Ação
            var resultado = Calculadora.QuadradoAltura(altura);

            // Assert - Verificação
            Assert.AreEqual(resultadoEsperado, resultado);
        }
    }
}

```

O próximo código fará referência ao teste do método CalcularIMC, com a utilização do projeto de teste XUnit:

```
using CalculadoraIMCPFS;
using Xunit;

namespace CalculadoraIMCUnit
{
    public class TesteCalculadoraIMCUnit
    {
        [Fact]
        public void TesteCalcularIMC()
        {
            // Arrange - Preparação
            double altura = 1.80;
            double peso = 70.0;
            string resultadoEsperado = "Categoria: Peso normal";

            // Act - Ação
            var resultado = Calculadora.CalcularIMC(altura,
            peso);

            // Assert - Verificação
            Assert.Contains(resultadoEsperado, resultado);
        }

        [Theory]
        [InlineData(1.8, 50.0, "Categoria: Abaixo do peso")]
        [InlineData(1.8, 90.0, "Categoria: Sobrepeso")]
        [InlineData(1.8, 110.0, "Categoria: Obesidade Grau I")]
        [InlineData(1.8, 120.0, "Categoria: Obesidade Grau II")]
        [InlineData(1.8, 140.0, "Categoria: Obesidade Grau III")]
        [InlineData(0, 120.0, "Altura inválida!")]
        [InlineData(-1.8, 120.0, "Opa! Ocorreu um erro!")]
        public void TesteCalcularIMCLista(double altura, double
            peso, string resultadoEsperado)
        {
            // Act - Ação
            var resultado = Calculadora.CalcularIMC(altura,
            peso);

            // Assert - Verificação
            Assert.Contains(resultadoEsperado, resultado);
        }
    }
}
```

6. Cronograma das atividades

A tabela a seguir mostra o cronograma para realização das atividades:

Tarefa	Data de Início	Data de Término
Planejamento de Teste	15/02/2022	17/02/2022
Documentos de requisitos de revisão	17/02/2022	17/02/2022
Preparação do ambiente de testes	18/02/2022	18/02/2022
Desenvolver o Desenho de Teste	19/02/2022	19/02/2022
Executar o Teste	20/02/2022	20/02/2022
Avaliar Teste	20/02/2022	21/02/2022

7. Relatórios de testes

Nesta seção serão apresentados os relatórios dos resultados dos testes de unidade e de integração, sendo este último realizado na aplicação Chapter Web API.

7.1 Testes de unidades

- Testes do projeto MStest:

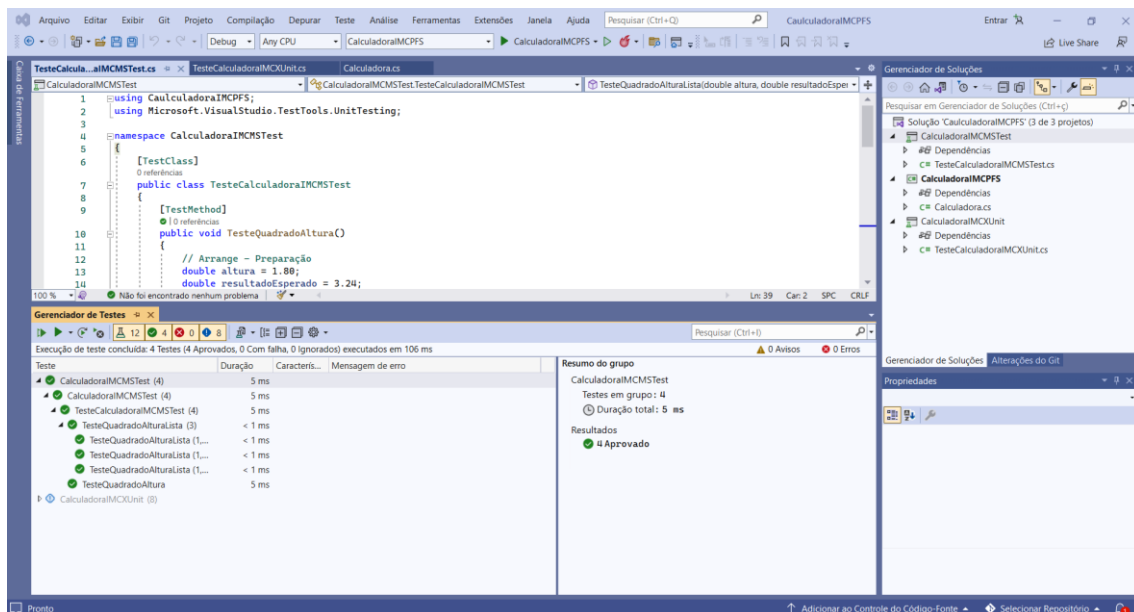
Testes em grupo: 4;

Duração da execução do teste: 5ms;

Duração total: 3 min;

Resultados: 4 Aprovados, 0 Com Falhas, 0 Ignorados;

Descrição: Os testes foram realizados com sucesso, sem apresentar qualquer tipo de erro. Não foram necessários testes adicionais.



- Testes do projeto Xunit:

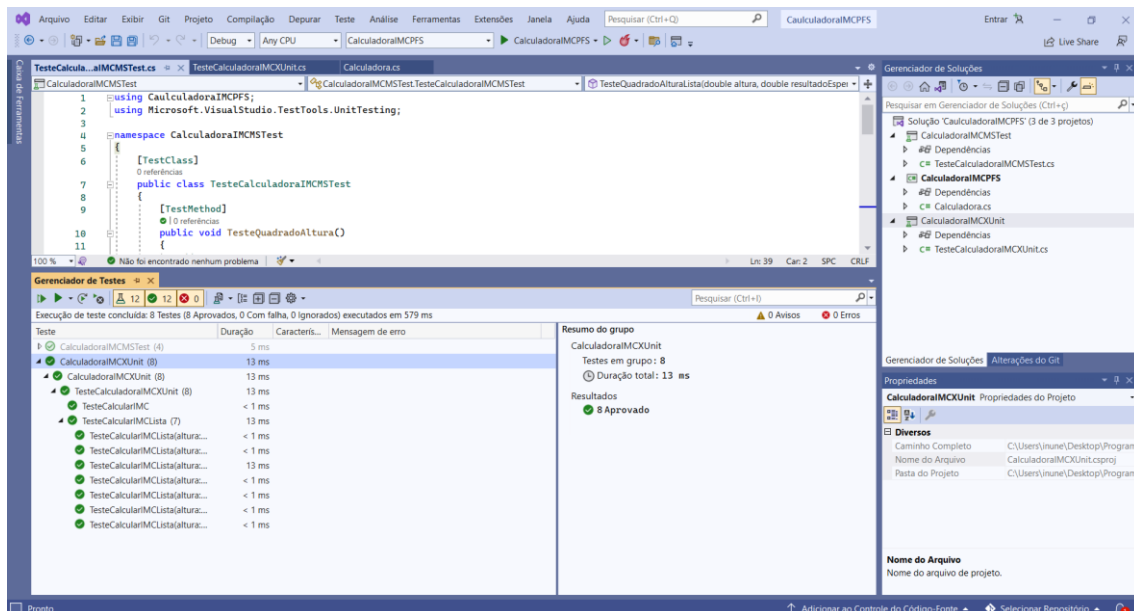
Testes em grupo: 8;

Duração da execução do teste: 13ms;

Duração total: 5 min;

Resultados: 8 Aprovados, 0 Com Falhas, 0 Ignorados;

Descrição: Os testes foram realizados com sucesso, sem apresentar qualquer tipo de erro. Não foram necessários testes adicionais.



7.2 Testes de Integração

Testes do projeto Xunit:

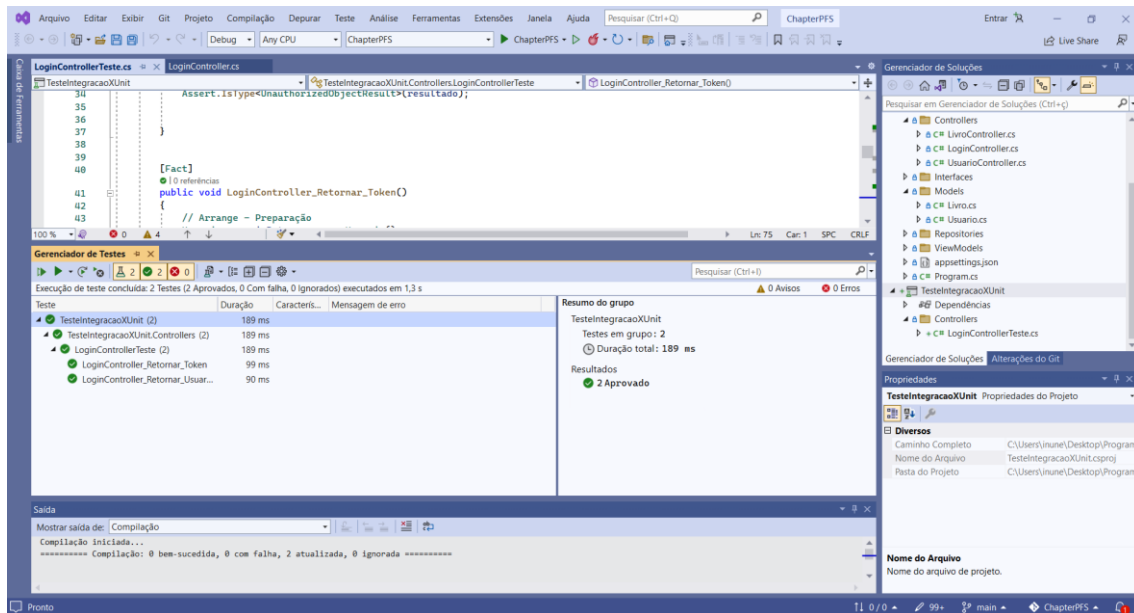
Testes em grupo: 2;

Duração da execução do teste: 189ms;

Duração total: 2 min;

Resultados: 2 Aprovados, 0 Com Falhas, 0 Ignorados;

Descrição: Os testes foram realizados com sucesso, sem apresentar qualquer tipo de erro. Não foram necessários testes adicionais.



7.3 Responsáveis

Os responsáveis pelo planejamento, execução e avaliação do teste, além de demais possíveis alterações e adaptações será a equipe:

Representante	Igor dos Santos Nunes
Descrição	Responsável pelo planejamento, execução e avaliação do teste
Tipo de Formação	Graduando em Engenharia da Computação e Programador Full-Stack.
Responsabilidade	Testar a funcionalidade do componente escolhido, além de garantir integridade do teste, deste documento e dos outros relatórios advindos do projeto.
Envolvimento	Dedicado integralmente ao projeto.

8. Normas e referências utilizadas

A equipe responsável por esse projeto utilizou como base a norma IEEE 829 (2008) para a confecção do planejamento do mesmo e deste documento e seguiu o formato de modelo referência de plano de teste da IBM (disponível no seguinte link: <https://www.ibm.com/docs/pt-br/elm/6.0?topic=sections-test-plan-template-reference>), sendo possível a existência de alterações e adaptações para a montagem do projeto.