

SENAI-SP

# Plano de Testes

NutriVitta

Aluno: Marcelo Marques

Data: Junho/2022.

# SUMÁRIO

<b>1. INTRODUÇÃO</b>	<b>3</b>
<b>1.2 OBJETIVOS</b>	<b>3</b>
1.3 ESCOPO GERAL	3
<b>2. REQUISITOS DE TESTE</b>	<b>3</b>
2.1 TESTE UNITÁRIO	3
<b>3. ESTRATÉGIAS DE TESTE</b>	<b>4</b>
3.1 Casos de Teste	4
3.1.1 Lançamento de Exceção ( Parâmetros inválidos para cálculo do IMC)	4
3.1.2 Cálculo do IMC	5
3.1.3 Classificação do IMC	6
<b>4. RECURSOS</b>	<b>7</b>
4.1 FERRAMENTAS	7
4.2 RECURSOS HUMANOS/EQUIPE	7
<b>5. DESENHO DE TESTE</b>	<b>7</b>
5.1 Teste para Lançamento de exceção no cálculo IMC (peso e/ou altura inválidos) .	8
5.2 Teste para cálculo do IMC.	8
5.3 Teste para classificação do IMC.	9
<b>6. CRONOGRAMA</b>	<b>9</b>

# 1. INTRODUÇÃO

Este documento elaborado pela empresa *Testful*, destina-se à especificação de um planejamento de atividades de teste para um sistema de cálculo e classificação do IMC , que representa um padrão internacional para avaliar o grau de obesidade, servindo de parâmetro para prestar atendimento clínico aos clientes da Empresa *NutriVitta*.

Outrossim, este documento terá como premissa o seu acompanhamento durante todas as fases de teste, de modo a poder sofrer revisões , verificações e validações dos resultados obtidos.

## 1.2 OBJETIVOS

Esta seção contém os objetivos descritos que orientam as atividades de teste do sistema a ser desenvolvido, conforme os itens abaixo:

- Identificar os requisitos ou componentes de software que serão testados.
- Definir as estratégias de teste que serão empregadas.
- Estabelecer os recursos necessários para obter uma previsão do empenho que será exercido durante as atividades de teste.
- Identificar os artefatos de saída ou produtos decorrentes da atividade do teste.
- Estabelecer um cronograma para contemplar as atividades do teste.

## 1.3 ESCOPO GERAL

Serão elaborados testes unitários em relação aos códigos implementados que definem os requisitos descritos como : cálculo de IMC e classificação do IMC.

# 2. REQUISITOS DE TESTE

## 2.1 TESTE UNITÁRIO

Permite avaliar trechos específicos de código relacionados às funcionalidades descritas no escopo, observando os parâmetros de entrada e as saídas durante a execução, para que possa ser feita uma análise comparativa entre o resultado real e o esperado.

- a) Para a funcionalidade **Cálculo do IMC**, verificar se os parâmetros de entrada, como peso e/ou altura são positivos e diferentes de zero, caso contrário, será lançada uma exceção.

- b) Para a funcionalidade **Cálculo do IMC**, considerar os parâmetros de entrada (peso e altura) de acordo com o esperado conforme item (a), e verificar o retorno do valor do IMC calculado.
- c) Para a funcionalidade **Classificação do IMC**, certificar-se que o IMC calculado na operação anterior, retorna como resultado um dos valores descritos na faixa de domínio chamada categorias de IMC.

## 3. ESTRATÉGIAS DE TESTE

### 3.1 Casos de Teste

#### 3.1.1 Lançamento de Exceção ( Parâmetros inválidos para cálculo do IMC)

<b>Objetivo do teste</b>	Verificar o lançamento de exceção para os testes do cálculo IMC, de acordo com a fórmula <b><i>peso/(altura)<sup>2</sup></i></b> .
<b>Técnica</b>	Utilizando a linguagem C# e a ferramenta de testes chamada xUnit, aplicar a realização dos casos de testes para cálculo IMC, utilizando peso e/ou altura com valores inválidos.
<b>Critério de Finalização</b>	Lançamento de exceção do tipo <b>ArgumentOutOfRangeException</b> para os casos de teste contendo valores inválidos, confirmando o erro.

Caso	Peso(Kg)	Altura(m)	Resultado esperado
1	0.0	1.60	ArgumentOutOfRangeException
2	-60.0	1.80	ArgumentOutOfRangeException
3	70.0	0.00	ArgumentOutOfRangeException
4	80.0	-1.70	ArgumentOutOfRangeException
5	-80.0	-1.70	ArgumentOutOfRangeException
6	0.00	0.00	ArgumentOutOfRangeException

### 3.1.2 Cálculo do IMC

<b>Objetivo do teste</b>	Considerando as variáveis do peso(Kg) e a altura (m), através da fórmula para cálculo do IMC : $\text{peso}/(\text{altura})^2$ , verificar por meio das entradas correspondentes se o resultado atende ao esperado.
<b>Técnica</b>	Utilizando a linguagem C# e a ferramenta de testes chamada xUnit, aplicar a realização dos testes para o cálculo do IMC , cujo valor deve ser retornado com uma casa decimal.
<b>Critério de Finalização</b>	Análise comparativa entre o resultado real e esperado, que devem ser iguais.

Caso	Peso(Kg)	Altura(m)	Resultado esperado
1	40.0	1.60	15.6
2	70.0	1.80	21.6
3	70.0	1.58	28.0
4	85.0	1.65	31.2
5	80.0	1.50	35.6
6	90.0	1.50	40.0

### 3.1.3 Classificação do IMC

<b>Objetivo do teste</b>	Considerando o resultado do cálculo do IMC para cada caso de teste anterior, verificar a classificação de acordo com a tabela Abeso.
<b>Técnica</b>	Usando a linguagem C# e a ferramenta de testes chamada xUnit, aplicar o desenho de teste voltado para classificação do IMC, com base no IMC calculado.
<b>Critério de Finalização</b>	Análise comparativa entre o resultado real da classificação do IMC e o resultado esperado, que devem ser iguais.

<b>Caso</b>	<b>IMC</b>	<b>Resultado esperado</b>
1	15.6	Abaixo do peso
2	21.6	Peso normal
3	28.0	Sobrepeso
4	31.2	Obesidade Grau I
5	35.6	Obesidade Grau II
6	40.0	Obesidade Grau III

## 4. RECURSOS

Esta seção identifica os principais recursos utilizados para a realização dos testes.

### 4.1 FERRAMENTAS

	Ferramenta/Software	<i>Criado por</i>
Sistema Operacional	Windows 10.0, versão 20H2	Microsoft
Plataforma de desenvolvimento	.NET 6.0	Microsoft
Framework para teste	xUnit	NUnit
Ambiente de desenvolvimento(IDE)	Visual Studio 2022, Versão 17.1.6	Microsoft
Software para documentação	Google Docs	Google

### 4.2 RECURSOS HUMANOS/EQUIPE

Aluno do Curso de Qualificação FullStack do SENAI-SP - Módulo Back end , Marcelo Marques, sob orientação do Sr. Professor Odirlei Sabella.

## 5. DESENHO DE TESTE

Consiste na implementação do código usado para realização do teste unitário do sistema de cálculo de IMC , tendo como base o framework xUnit e os casos de teste que deverão ser avaliados conforme os resultados obtidos.

## 5.1 Teste para Lançamento de exceção no cálculo IMC (peso e/ou altura inválidos) .

```
[Theory]
[InlineData(0.0, 1.60)]
[InlineData(-60.0, 1.80)]
[InlineData(70.0, 0.00)]
[InlineData(80.0, -1.70)]
[InlineData(-80.0, -1.70)]
[InlineData(0.0, 0.00)]
public void Peso_Ou_Altura_ComValorInvalido_RetornaErro(double peso,
double altura)
{
    Assert.Throws<ArgumentOutOfRangeException>(() =>
Operacao.CalcularIMC(peso, altura));
}
```

## 5.2 Teste para cálculo do IMC.

```
[Theory]
[InlineData(40.0, 1.60, 15.6)]
[InlineData(70.0, 1.80, 21.6)]
[InlineData(70.0, 1.58, 28.0)]
[InlineData(85.0, 1.65, 31.2)]
[InlineData(80.0, 1.50, 35.6)]
[InlineData(90.0, 1.50, 40.0)]
public void CalcularIMCLista(double peso, double altura, double imc)
{
    var resultado = Operacao.CalcularIMC(peso, altura);

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



### 5.3 Teste para classificação do IMC.

```
[Theory]
[InlineData(15.6, Categoria.AbaixoDoPeso)]
[InlineData(21.6, Categoria.PesoNormal)]
[InlineData(28.0, Categoria.Sobrepeso)]
[InlineData(31.2, Categoria.ObesidadeGrauI)]
[InlineData(35.6, Categoria.ObesidadeGrauII)]
[InlineData(40.0, Categoria.ObesidadeGrauIII)]
public void ClassificarIMCLista(double imc, Categoria categoria)
{
    var resultado = Operacao.ExibirClassificacao(imc);

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

## 6. CRONOGRAMA

	Data de Início	Data do término
Planejamentos dos testes	9 de jun. de 2022	9 de jun. de 2022
Execução dos testes	9 de jun. de 2022	9 de jun. de 2022
Avaliação dos testes	10 de jun. de 2022	10 de jun. de 2022