



SERVIÇO NACIONAL DE
APRENDIZAGEM INDUSTRIAL

Resumo Relatório de testes.

Juliana Barroso – Curso Desenvolvedor Full Stack – Turma 2

UC 15 – Testes de Back – End.

Sumário

1	OBJETIVO.....	3
2	ESCOPO	3
3	REQUISITOS.....	3
4	CASOS DE TESTE.....	3
4.1	CALCULAR IMC.....	3
4.2	CLASSIFICAR IMC	4
5	FERRAMENTAS	4
6	DESENHO DE TESTE.....	5
6.1	PROJETO IMC.....	5
6.2	TESTE XUNIT IMC	6
7	TESTE DE INTEGRAÇÃO CHAPTER.....	8
8	RECURSOS	11
8.1	EQUIPE.....	11
8.2	SISTEMA	11
9	CRONOGRAMA.....	11

1 Objetivo

Elaboração de um plano de teste para o sistema de cálculo de IMC solicitado pela empresa Nutrivitta, desenvolvido pela empresa Testful, contemplando:

- Executar testes de unidade;
- Verificação da funcionalidade do sistema de cálculo e classificação do IMC;
- Identificação de possíveis erros;
- Apresentação de melhorias para o sistema.

2 Escopo

Realização de testes unitários na calculadora de IMC nos requisitos de cálculo do IMC (utilizados dados hipotéticos) e na classificação do IMC com base na tabela Abeso.

3 Requisitos

Testes dos casos de cálculo e classificação de IMC, para verificação da funcionalidade do sistema, certificando que estão apresentando os resultados de forma correta e coerente.

4 Casos de Teste

4.1 Calcular IMC

Calcular o IMC do usuário da seguinte forma:

VEJA A FÓRMULA DO IMC

$$\text{IMC} = \frac{\text{PESO}}{\text{ALTURA} \times \text{ALTURA}}$$

Exemplo: Usuário 1 possui 68 Kg e 1,70 m de altura.

Onde $1,70 \times 1,70 = 2,89$.

Logo $68 / 2,89 = 23,53$. O resultado do IMC é: 23,53.

4.2 Classificar IMC

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

De acordo com a tabela acima, o Usuário 1 exemplificado possui um IMC referente a Peso Normal, enquadrando dentro do valor de 25,0 a 29,9, **portanto será exibida uma mensagem de: “Peso Normal.”**

5 Ferramentas

Foram utilizados os seguintes softwares para a realização deste *Plano de Teste*:

- S.O Windows 10 Pro;
- Microsoft Office Word 2021;
- Microsoft Visual Studio 2022;
- Xunit;
- Linguagem de programação C#;

6 Desenho de teste

No desenho de teste há a demonstração de um teste unitário, conforme o exemplo descrito no item anterior e o teste com outros exemplos de usuário, demonstrando o cálculo do IMC e a classificação de cada um.

6.1 Projeto IMC

```
namespace NutrivittaIMC
{
    public static class CalcIMC
    {
        public static double CalculoIMC(double peso, double altura)
        {

```

```

        return peso / (altura * altura);
    }

    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;
    }
}

```

6.2 Teste Xunit IMC

```

using NutrivittaIMC;
using System;
using Xunit;

namespace TesteIMC
{
    public class UnitTest1
    {
        [Fact]
        public void CalculoIMC()
        {
            //Arrange

            double peso = 68;
            double altura = 1.70;
            double imc = 23.53;

            //Act
            var resultado = CalcIMC.CalculoIMC(peso, altura);

```

```

        //Assert
        Assert.Equal(imc, Math.Round(resultado, 2)); // math round e o 2 após o
resultado é para usar até 2 casas depois do ponto (vírgula)
    }

    [Theory]

    [InlineData(45, 1.57, 18.26)]
    [InlineData(65, 1.70, 22.49)]
    [InlineData(74, 1.70, 25.61)]
    [InlineData(95, 1.67, 34.06)]
    [InlineData(110, 1.75, 35.92)]
    [InlineData(125, 1.70, 43.25)]

    public void CalcIMCLista(double peso, double altura, double imc)
    {
        var resultado = CalcIMC.CalculoIMC(peso, altura);
        Assert.Equal(imc, Math.Round(resultado, 2));
    }

    [Fact]
    public void ClassIMC()
    {
        //Arrange
        double imc = 23.52;
        var classificar = "Peso Normal";

        //Act
        var result = CalcIMC.ClassificarIMC(imc);

        //Assert
        Assert.Equal(classificar, result);
    }

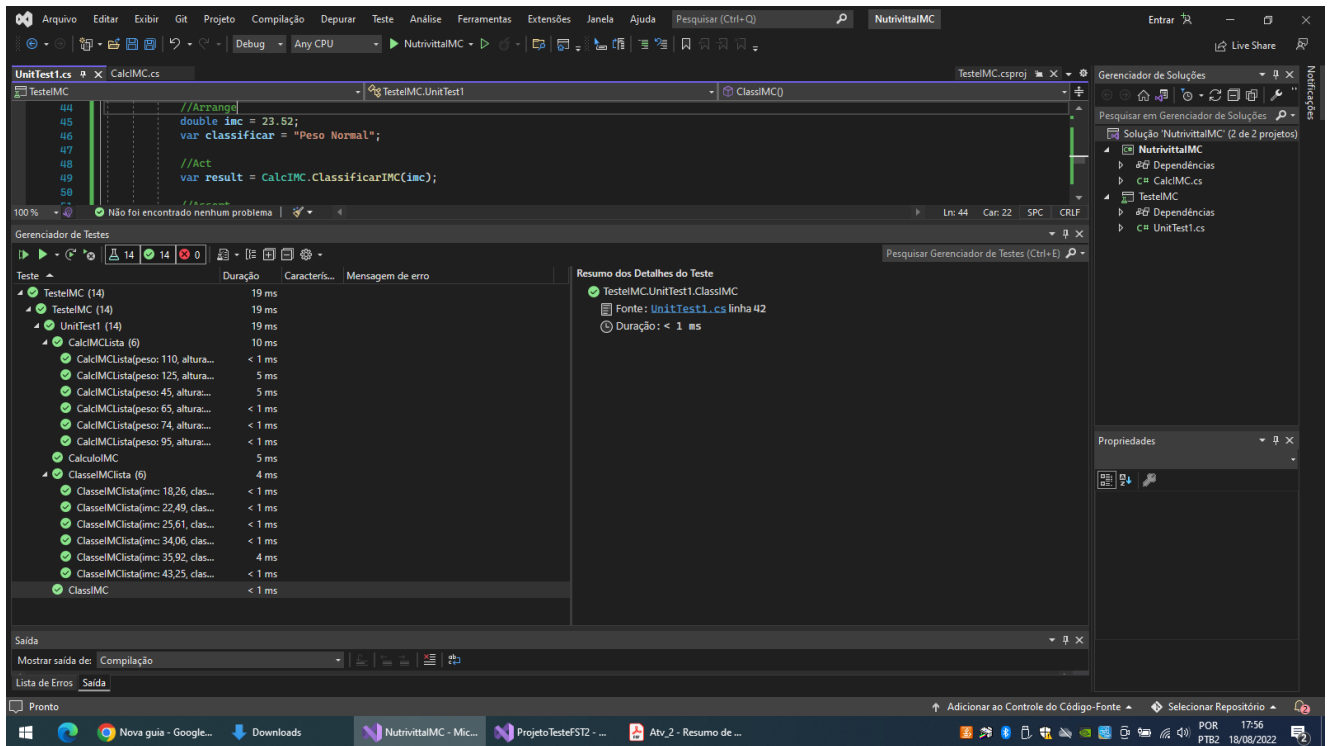
    [Theory]

    [InlineData(18.26, "Abaixo do peso")]
    [InlineData(22.49, "Peso Normal")]
    [InlineData(25.61, "Sobrepeso")]
    [InlineData(34.06, "Obesidade Grau I")]
    [InlineData(35.92, "Obesidade Grau II")]
    [InlineData(43.25, "Obesidade Grau III")]

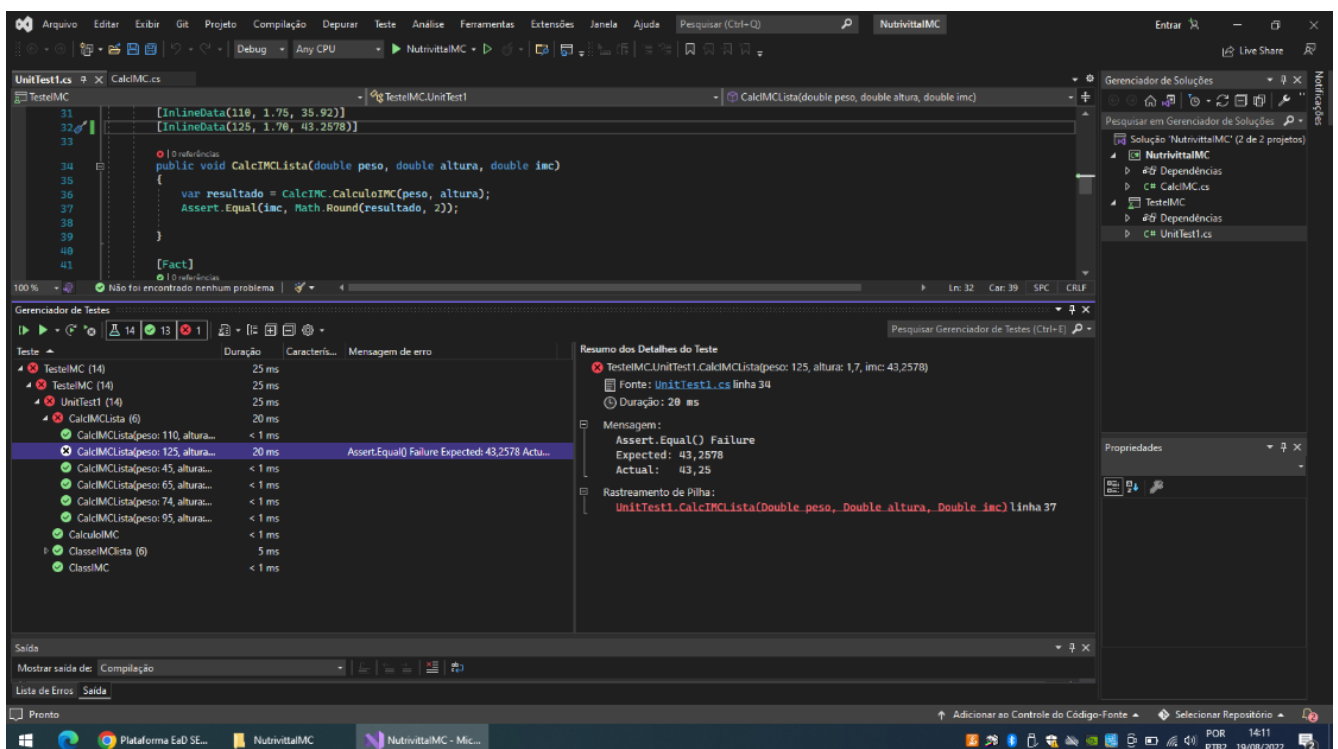
    public void ClasseIMCLista(double imc, string classificacao)
    {
        var resultado = CalcIMC.ClassificarIMC(imc);
        Assert.Equal(classificacao, resultado);
    }
}

```

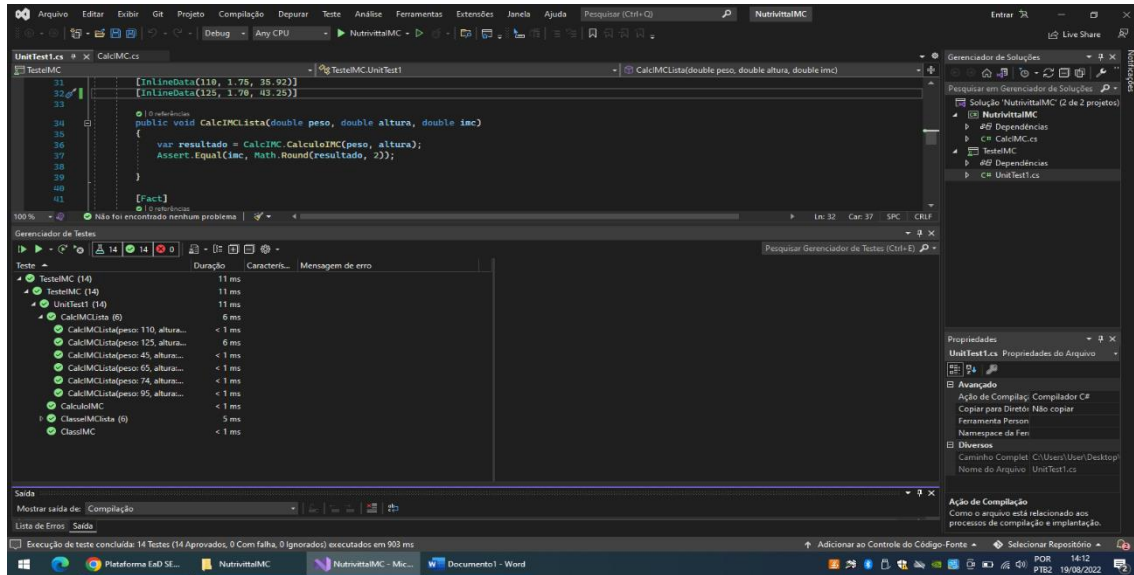
Imagem Resultado do Teste.



Simulação de um erro de cálculo no teste de IMC:



Erro corrigido:



7 Teste de Integração Chapter

Foi realizado o teste de integração de api no arquivo CHAPTER usado em aula do encontro remoto simulando o retorno de um usuário inválido.

```
using ChapterFST1.Controllers;
using ChapterFST1.Interfaces;
using ChapterFST1.Models;
using ChapterFST1.ViewModels;
using Microsoft.AspNetCore.Mvc;
using Moq;
using System.IdentityModel.Tokens.Jwt;
using Xunit;

namespace TesteIntegracao.Controllers
{
    public class LoginControllerTeste
    {
        [Fact]
        public void LoginController_Retornar_Usuario_Invalido()
        {
            // Arrange - Preparação

            var repositorioEspelhado = new Mock<IUsuarioRepository>();

            repositorioEspelhado.Setup(x => x.Login(It.IsAny<string>()),
            It.IsAny<string>())) .Returns((Usuario)null);
```



```

LoginViewModel dados = new LoginViewModel();
dados.Email = "email@email.com";
dados.Senha = "123456";

var controller = new LoginController(repositorioEspelhado.Object);

// Act - Ação

var resultado = controller.Login(dados);

// Assert - Verificação

Assert.IsType<UnauthorizedObjectResult>(resultado);
}

[Fact]
public void LoginController_Retornar_Token()
{
    // Arrange - Preparação

    Usuario usuarioRetorno = new Usuario();
    usuarioRetorno.Email = "email@email.com";
    usuarioRetorno.Senha = "123456";
    usuarioRetorno.Tipo = "1";
    usuarioRetorno.Id = 1;

    var repositorioEspelhado = new Mock<IUsuarioRepository>();

    repositorioEspelhado.Setup(x => x.Login(It.IsAny<string>()),
It.IsAny<string>()))).Returns(usuarioRetorno);

    string issuerValido = "chapter.webapi";

    LoginViewModel dados = new LoginViewModel();
    dados.Email = "email@email.com";
    dados.Senha = "1234";

    var controller = new LoginController(repositorioEspelhado.Object);

    // Act - Ação

    OkObjectResult resultado = (OkObjectResult)controller.Login(dados);

    string tokenString = resultado.Value.ToString().Split(' ')[3];

    var jwtHandler = new JwtSecurityTokenHandler();
    var tokenJwt = jwtHandler.ReadJwtToken(tokenString);

```

```
// Assert - Verificação
```

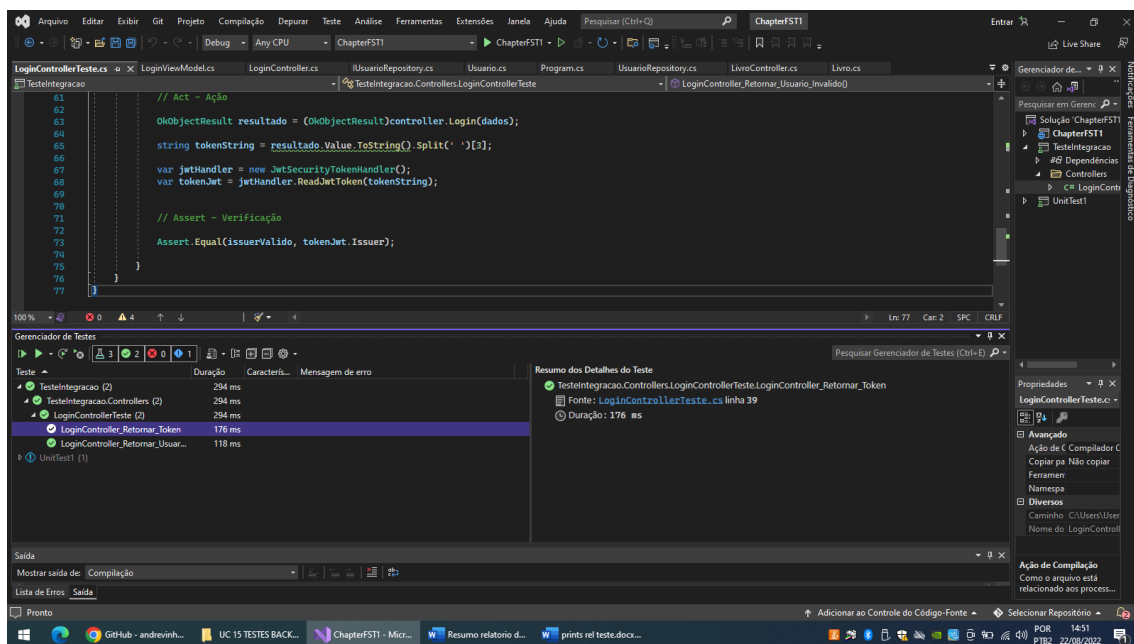
```
Assert.Equal(issuerValido, tokenJwt.Issuer);
```

```
}
```

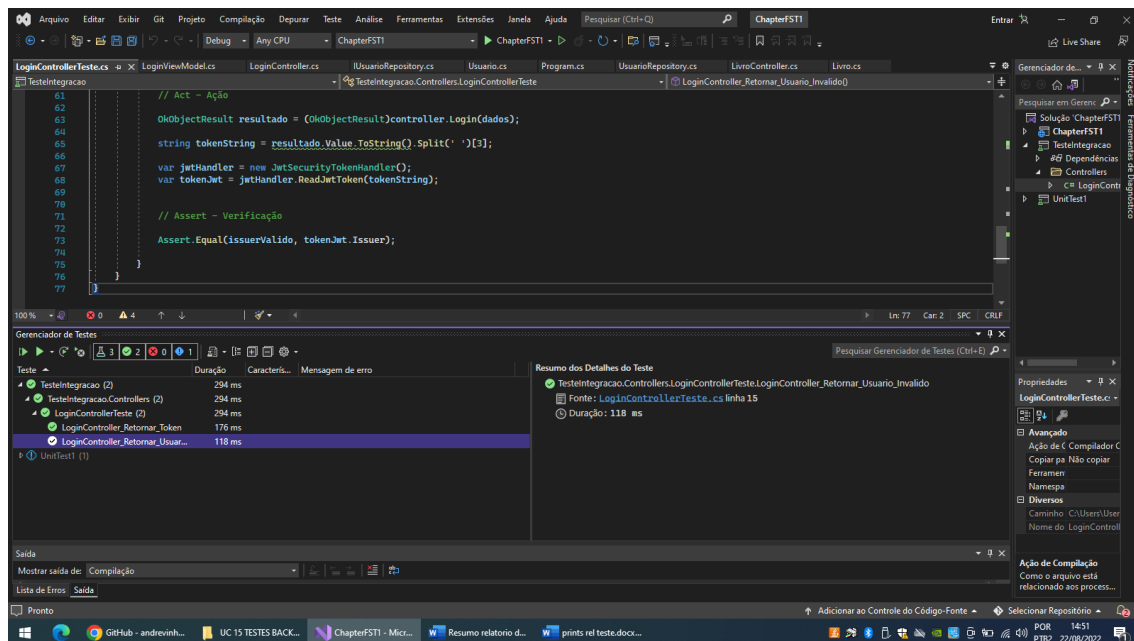
```
}
```

```
}
```

Retorno do token concluído com sucesso.



Retorno do token concluído com sucesso.



8 Recursos

8.1 Equipe

Desenvolvedor(a) executante do plano e desenho de teste: Juliana Barroso.
(Aluna curso Desenvolvedor Full Stack Senai)

Professor responsável pela disciplina: Odirlei Sabella de Assis

8.2 Sistema

Implementado pela linguagem de programação C# e utilizando Xunit como programa de teste e executado pelo sistema operacional Windows 10 Pro.

9 Cronograma

Tempo de execução do teste IMC: 7 minutos.

Tempo de execução do teste de integração: 20 minutos.