

Plano de teste:

Calculadora IMC para empresa NutriVittA

Preparado por:

EMILY MUELLER

ANA FULANA

BEATRIZ CICLANA

(05/10/2022)

SUMÁRIO

1 INTRODUÇÃO

2 OBJETIVOS E MEMBROS DA EQUIPE

2.1 Objetivos

2.2 Membros da Equipe

3 ESCOPO GERAL

4 REQUISITO DE TESTES

4.1 Teste unitário

5 ESTRATÉGIA DE TESTE

5.1 Casos de teste

6 DESENHO DOS TESTES

6.1 Testes MSTest

6.2 Testes XUnit

7 REQUISITOS DE HARDWARE

8 FERRAMENTAS

9 PROCEDIMENTOS DE CONTROLE

10 RISCOS

11 CRONOGRAMA

12 HISTÓRICO DE REVISÕES

1.0 INTRODUÇÃO

O Plano de Teste foi criado para comunicar aos membros da equipe afim de elaborar um sistema de cálculo de IMC e conseqüentemente sua classificação para a empresa NutriVitta. Incluímos a seguir, os objetivos, escopo, cronograma e riscos. Este documento demonstrará quais serão os resultados do teste.

2.0 OBJETIVOS E MEMBROS DE EQUIPE

2.1 Objetivos

TESTE IMC é um sistema de cálculo do IMC, podendo armazenar os resultados para futuras avaliações. Esta ferramenta é um novo sistema que utiliza linguagem C# e um banco de dados SQL Server. A equipe de teste é tanto cliente quanto testador do produto e deverá garantir que ele atenda às suas necessidades.

2.2 Membros de Equipe

EQUIPE	PAPEL
Emily Mueller	Analista de testes
Ana Fulana	Gerente de testes
Beatriz Ciclana	Diretora de QA

3.0 ESCOPO GERAL

A fase inicial incluirá todos os requisitos necessárias. Ao final, o testador deve ser capaz de:

- Criar um teste manual com o máximo de etapas necessárias
- Salvar
- Recuperar e ter a capacidade de visualizá-lo ao executar o teste
- Digitar resultados e comentários apropriados
- Ver resultados

4.0 REQUISITOS DE TESTES

4.1 Teste unitário

Definição:

O sistema será testado isoladamente, afim de identificar erros com rapidez e precisão. Os testes exploratórios desempenharão uma grande parte dos testes, sendo assim a equipe aprenderá à medida que forem executados.

Através de casos de teste, iremos implementar as devidas correções, caso necessárias.

Testes automatizados de unidade fazem parte do processo de desenvolvimento, afim de prever acessos simultâneos ao sistema da calcula, mas nenhum teste funcional automatizado está planejado no momento.

5.0 ESTRATÉGIA DE TESTE

O sistema será testado isoladamente, afim de identificar erros com rapidez e precisão. Iremos utilizar o cálculo de IMC abaixo:

$$\frac{\text{Peso(kg)}}{(\text{Altura(m)} \times \text{Altura(m)})}$$

5.1 Casos de teste

Serão utilizados testes de velocidade para avaliar o retorno de resposta do sistema da calculadora IMC. Assim como testes de volume de dados que podem ser salvos no banco de dados e consequentemente seu impacto no sistema.

A seguir apresentamos uma tabela de casos de testes com as expectativas de resultados:

CASO	CÁLCULO	RESULTADO	CATEGORIA
1	40kg/(1,49mX1,49m)	18,01 – Faixa: Abaixo de 18,5	ABAIXO DO PESO
2	50kg/(1,45mX1,45m)	23,78 – Faixa: 18,5 – 24,9	PESO NORMAL
3	69kg/(1.64mX1.64m)	25,65 – Faixa: 25,0 – 29,9	SOBREPESO
4	89kg/(1,72mX1,72m)	30,08 – Faixa: 30,0 – 34,9	OBESIDADE GRAU 1
5	99kg/(1,68mX1,68m)	35,07 – Faixa: 35,0 – 39,9	OBESIDADE GRAU 2
6	122kg/(1,74mX1,74m)	40,29 – Faixa: 40,0 e acima	OBESIDADE GRAU 3

6.0 DESENHOS DOS TESTES

Para termos os testes tipo MSTest e Xunit, devemos ter um arquivo, nomeado Calculadora com os possíveis retornos de comparação. Segue abaixo a Figura 1:

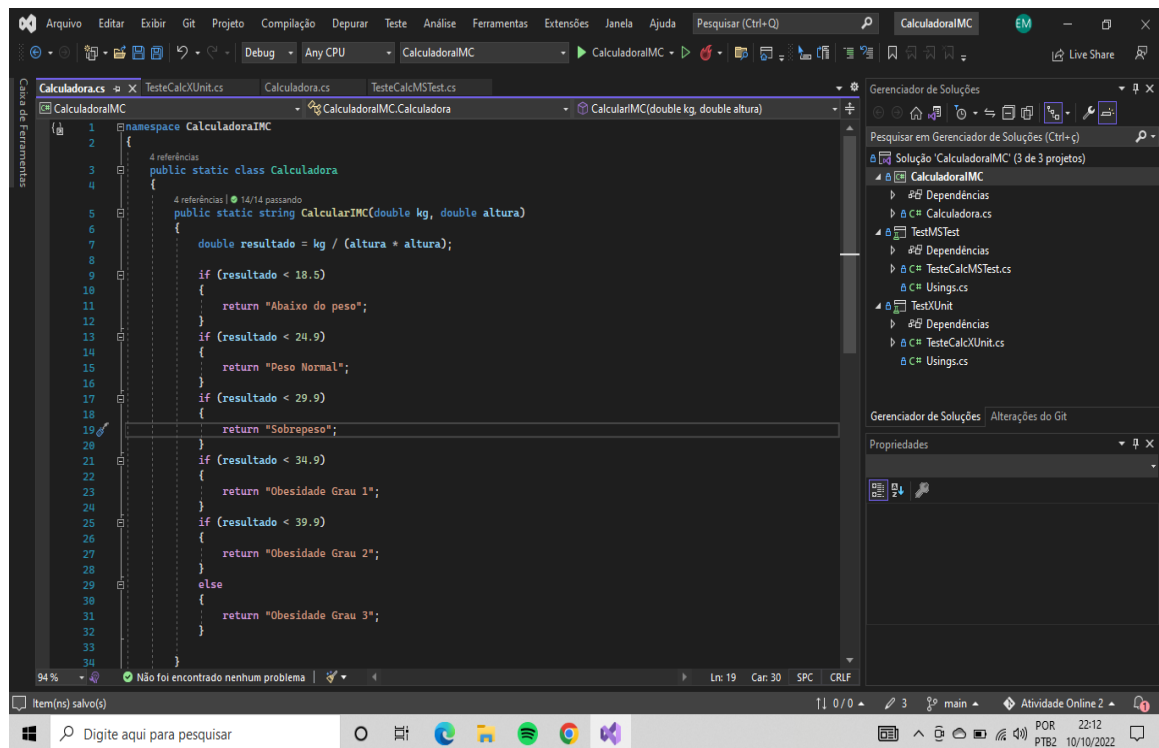


Figura 1: Arquivo “Calculadora” com os resultados esperados de comparação dos casos de testes, onde teremos kg e altura denominadas variáveis do tipo double e o resultado como variável do tipo string.

6.1 TESTES MSTEST

Para compararmos os casos de testes, utilizamos os testes do tipo MSTEST como representado na figura 2:

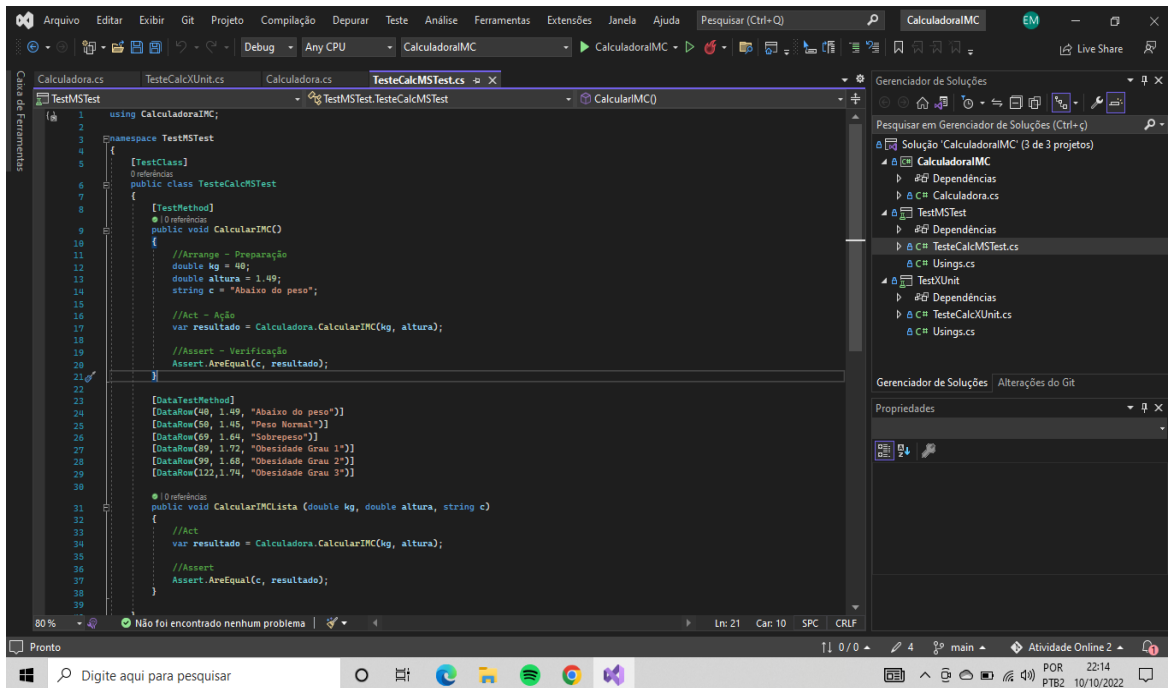


Figura 2: Arquivo com os casos de testes do tipo MSTest onde teremos kg e altura como variáveis do tipo double e o “c” e o resultado como variáveis do tipo string.

6.2 TESTES XUnit

Para compararmos os casos de testes, utilizamos os testes do tipo XUnit como representado na figura 3:

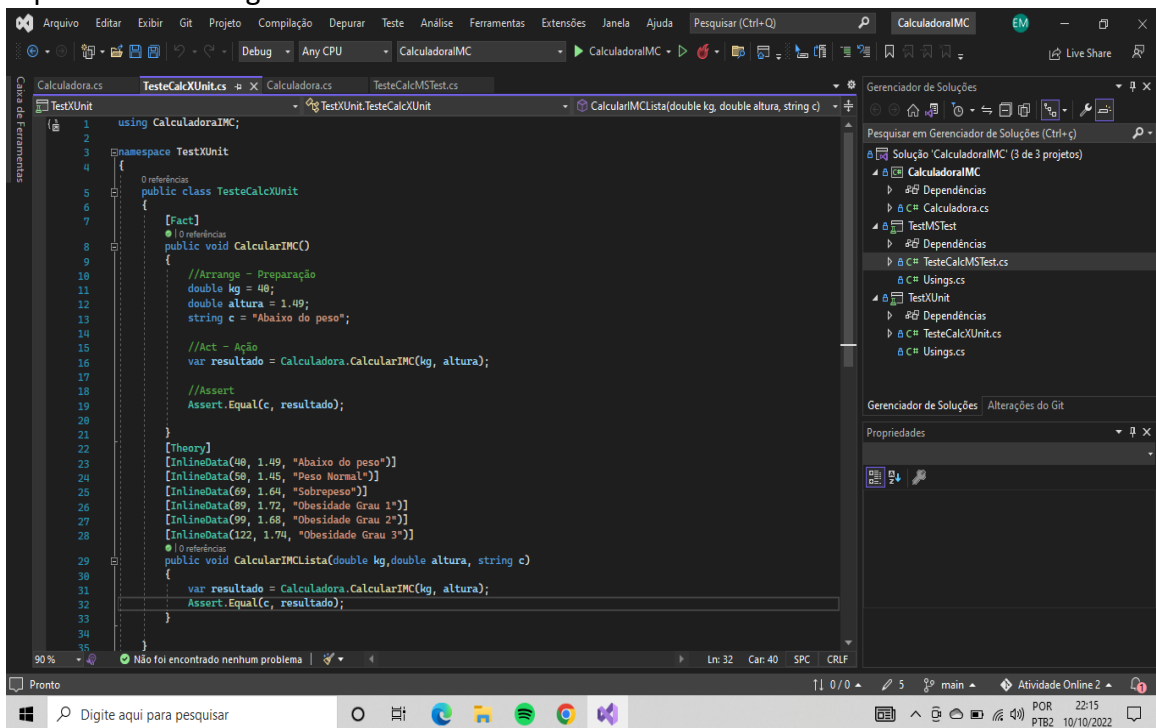


Figura 3: Arquivo com os casos de testes do tipo XUnit onde teremos kg e altura como variáveis do tipo double e o “c” e o resultado como variáveis do tipo string.

7.0 REQUISITOS DE HARDWARE

Computadores com processador 6ª geração, memória acima de 8GB, armazenamento mínimo de 256GB e sistema Windows 7.

8.0 FERRAMENTAS

Instalação do Microsoft Visual Studio versão 2022;
Instalação do Sql SERVER versão 2022

9.0 PROCEDIMENTOS DE CONTROLE

Relatórios de problemas

Quaisquer incidentes encontrados durante o processo de teste, deverão ser reportados na reunião semanal para que possam ser anexados ao plano de teste.

Solicitações de alteração

Em processos de modificações no software, haverá identificação de quem assinará as alterações e quais seriam os critérios para incluir as alterações no produto atual. Se as alterações afetarem os programas existentes, esses módulos precisam ser identificados.

10.0 RISCOS

De início, identificamos os seguintes riscos e as ações apropriadas para mitigar seu impacto no projeto. O impacto (ou gravidade) do risco baseia-se em como o projeto seria afetado se o risco fosse acionado. O gatilho é que marco ou evento faria com que o risco se tornasse um problema a ser tratado.

#	Risco	Impacto	Gatilho	Plano de Mitigação
1	À medida que os testadores se familiarizam mais com a ferramenta, eles vão querer mais funcionalidade	Alto	Atrasos na data de implementação	Cada iteração, a funcionalidade será monitorada de perto. As prioridades serão definidas e discutidas pelas partes interessadas.
2	Alterações na funcionalidade podem anular os testes já escritos	Alta	Perda de todos os casos de teste	Exportar dados antes de qualquer atualização conforme necessário e reimportar após a atualização.
3	A entrega semanal não é possível porque o desenvolvedor trabalha fora do site	Média	Produto não foi entregue dentro do prazo	Estabelecer plataforma de acesso mútuo para não ocorrer riscos de não entregar.

11.0 CRONOGRAMA

Nome da tarefa	Começar	Acabar	Esforço	Comentários
Planejamento de testes	05/10	05/10	1d	
Documentos de requisitos de revisão	05/10	05/10	1d	
Criar estimativas iniciais de teste	06/10	07/10	2 d	
Equipe e treinar novos recursos de teste	06/10	07/10	2 d	
Primeiro implantar no ambiente de teste de QA	06/10	07/10	2 d	
Teste funcional – Iteração 1	06/10	07/10	2 d	
Iteração 2 implantar para ambiente de teste de QA	10/10	11/10	2 d	
Teste funcional – Iteração 2	10/10	11/10	2 d	
Resolução de defeitos finais e testes finais de construção	13/10	15/10	3 d	
Implantar para o ambiente de preparação	16/10	17/10	2 d	
Teste de desempenho	17/10	18/10	2 d	

12.0 HISTÓRICO DE REVISÕES

VERSÃO	DATA	RESPONSÁVEL	OBSERVAÇÕES