Relatório de Resumo de Testes de Unidade

Calculdora IMC - NutriVitta

Autor: Pedro Ingro K S P Santos

Objetivo

Verificar se a classe que calcula o IMC da calculadora está retornando o resultado correto de acordo com os valores de entrada.

Equipe

O presente teste foi realizado pela desenvolvedora Aline Pereira.

Casos de teste

Ao todo foram executados oito casos de teste. Todos os casos de teste obtiveram êxito ao serem executados como pode ser verificado nas tabelas que seguem abaixo.

Caso de Teste	**Teste IMC Peso Normal**
Descrição do Teste	Teste de unidade realizado para averiguar o comportamento da Classe CalculdoraIMC ao receber valores válidos de entrada. O teste foi desenvolvido em linguagem de programação C# com o auxílio do Framework UnitTesting.
Pré-Condições	O usuário deve inserir peso de altura válidos para cálculo de IMC
Pós-Condições	A classe deve retornar ao usuário o seu IMC e sua classificação de acordo com a tabela da Abeso.
Notas:	Valores utilizados: peso 70kg e altura 1.75m
Resultados (Passou/Falhou/Incompleto)	Passou

	Passo do Teste	Resultados Esperados do Teste	P	F
1.	Classe recebe os valores 70 e 1.75 para peso e altura respectivamente	Nenhum comportamento esperado	x	
2.	Classe deve calcular o valor do IMC para os valores recebidos	IMC calculado deve retornar o valor 22.86	x	
3.	Classe deve classificar o IMC de acordo com seu valor	IMC deve receber a classificação “Peso Normal”	x	

Caso de Teste	Teste IMC abaixo do peso
Descrição do Teste	Teste de unidade realizado para averiguar o comportamento da Classe CalculdoraIMC ao receber valores válidos de entrada. O teste foi desenvolvido em linguagem de programação C# com o auxílio do Framework UnitTesting.
Pré-Condições	O usuário deve inserir peso de altura válidos para cálculo de IMC
Pós-Condições	A classe deve retornar ao usuário o seu IMC e sua classificação de acordo com a tabela da Abeso.
Notas:	Valores utilizados: peso 50kg e altura 1.65m
Resultados (Passou/Falhou/Incompleto)	Passou

	Passo do Teste	Resultados Esperados do Teste	P	F
1.	Classe recebe os valores 50 e 1.65 para peso e altura respectivamente	Nenhum comportamento esperado	x	
2.	Classe deve calcular o valor do IMC para os valores recebidos	IMC calculado deve retornar o valor 18.37	x	
3.	Classe deve classificar o IMC de acordo com seu valor	IMC deve receber a classificação “Abaixo do Peso”	x	

Caso de Teste	Teste IMC Sobrepeso
Descrição do Teste	Teste de unidade realizado para averiguar o comportamento da Classe CalculdoraIMC ao receber valores válidos de entrada. O teste foi desenvolvido em linguagem de programação C# com o auxílio do Framework UnitTesting.
Pré-Condições	O usuário deve inserir peso de altura válidos para cálculo de IMC
Pós-Condições	A classe deve retornar ao usuário o seu IMC e sua classificação de acordo com a tabela da Abeso.
Notas:	Valores utilizados: peso 85kg e altura 1.75m
Resultados (Passou/Falhou/Incompleto)	Passou

	Passo do Teste	Resultados Esperados do Teste	P	F
1.	Classe recebe os valores 85 e 1.75 para peso e altura respectivamente	Nenhum comportamento esperado	x	
2.	Classe deve calcular o valor do IMC para os valores recebidos	IMC calculado deve retornar o valor 27.76	x	
3.	Classe deve classificar o IMC de acordo com seu valor	IMC deve receber a classificação “Sobrepeso”	x	

Caso de Teste	Teste IMC Obesidade Grau I
Descrição do Teste	Teste de unidade realizado para averiguar o comportamento da Classe CalculdoraIMC ao receber valores válidos de entrada. O teste foi desenvolvido em linguagem de programação C# com o auxílio do Framework UnitTesting.
Pré-Condições	O usuário deve inserir peso de altura válidos para cálculo de IMC
Pós-Condições	A classe deve retornar ao usuário o seu IMC e sua classificação de acordo com a tabela da Abeso.
Notas:	Valores utilizados: peso 100kg e altura 1.75m
Resultados (Passou/Falhou/Incompleto)	Passou

	Passo do Teste	Resultados Esperados do Teste	P	F
1.	Classe recebe os valores 100 e 1.75 para peso e altura respectivamente	Nenhum comportamento esperado	x	
2.	Classe deve calcular o valor do IMC para os valores recebidos	IMC calculado deve retornar o valor 32.65	x	
3.	Classe deve classificar o IMC de acordo com seu valor	IMC deve receber a classificação “Obesidade Grau I”	x	

Caso de Teste	Teste IMC Obesidade Grau II
Descrição do Teste	Teste de unidade realizado para averiguar o comportamento da Classe CalculdoraIMC ao receber valores válidos de entrada. O teste foi desenvolvido em linguagem de programação C# com o auxílio do Framework UnitTesting.
Pré-Condições	O usuário deve inserir peso de altura válidos para cálculo de IMC
Pós-Condições	A classe deve retornar ao usuário o seu IMC e sua classificação de acordo com a tabela da Abeso.
Notas:	Valores utilizados: peso 120kg e altura 1.75m
Resultados (Passou/Falhou/Incompleto)	Passou

	Passo do Teste	Resultados Esperados do Teste	P	F
1.	Classe recebe os valores 120 e 1.75 para peso e altura respectivamente	Nenhum comportamento esperado	x	
2.	Classe deve calcular o valor do IMC para os valores recebidos	IMC calculado deve retornar o valor 39.18	x	
3.	Classe deve classificar o IMC de acordo com seu valor	IMC deve receber a classificação “Obesidade Grau II”	x	

Caso de Teste	Teste IMC Obesidade Grau III
Descrição do Teste	Teste de unidade realizado para averiguar o comportamento da Classe CalculdoraIMC ao receber valores válidos de entrada. O teste foi desenvolvido em linguagem de programação C# com o auxílio do Framework UnitTesting.
Pré-Condições	O usuário deve inserir peso de altura válidos para cálculo de IMC
Pós-Condições	A classe deve retornar ao usuário o seu IMC e sua classificação de acordo com a tabela da Abeso.
Notas:	Valores utilizados: peso 150kg e altura 1.75m
Resultados (Passou/Falhou/Incompleto)	Passou

	Passo do Teste	Resultados Esperados do Teste	P	F
1.	Classe recebe os valores 150 e 1.75 para peso e altura respectivamente	Nenhum comportamento esperado	x	
2.	Classe deve calcular o valor do IMC para os valores recebidos	IMC calculado deve retornar o valor 48.98	x	
3.	Classe deve classificar o IMC de acordo com seu valor	IMC deve receber a classificação “Obesidade Grau III”	x	

Caso de Teste	**Teste IMC para Altura Zero**
Descrição do Teste	Teste de unidade realizado para averiguar o comportamento da Classe CalculdoraIMC ao receber valores válidos de entrada. O teste foi desenvolvido em linguagem de programação C# com o auxílio do Framework UnitTesting.
Pré-Condições	O usuário deve inserir peso de altura válidos para cálculo de IMC
Pós-Condições	A classe deve retornar ao usuário o seu IMC e sua classificação de acordo com a tabela da Abeso.
Notas:	Valores utilizados: peso 70kg e altura 0.0
Resultados (Passou/Falhou/Incompleto)	Passou

	Passo do Teste	**Resultados Esperados do Teste**	**P**	**F**
1.	Classe recebe os valores 70 e 0.0 para peso e altura respectivamente	Nenhum comportamento esperado	x	
2.	Classe deve calcular o valor do IMC para os valores recebidos	Classe deve retornar exceção do tipo *DivideByZeroException*	x	

Caso de Teste		**Teste IMC para peso inválido (negativo)**		
Descrição do Teste		Teste de unidade realizado para averiguar o comportamento da Classe CalculdoraIMC ao receber valores válidos de entrada. O teste foi desenvolvido em linguagem de programação C# com o auxílio do Framework UnitTesting.		
Pré-Condições		O usuário deve inserir peso de altura válidos para cálculo de IMC		
Pós-Condições		A classe deve retornar ao usuário o seu IMC e sua classificação de acordo com a tabela da Abeso.		
Notas:		Valores utilizados: peso -70kg e altura 1.75m		
Resultados (Passou/Falhou/Incompleto)		Passou		
	Passo do Teste	**Resultados Esperados do Teste**	**P**	**F**
1.	Classe recebe os valores -70 e 1.75 para peso e altura respectivamente	Nenhum comportamento esperado	x	
2.	Classe deve calcular o valor do IMC para os valores recebidos	Classe deve retornar exceção do tipo *System.Exception*	x	

Relatório de Resumo de Testes de Integração

Chapter API

Autor: Pedro Ingro K S P Santos

Objetivo

Validar o comportamento da API Chapter em relação a autenticação de usuários.

Equipe

O presente teste foi realizado pela desenvolvedora Aline Pereira.

Casos de teste

Ao todo foram executados dois casos de teste. Todos os casos de teste obtiveram êxito ao serem executados como pode ser verificado nas tabelas que seguem abaixo.

Caso de Teste	**Retornar Usuário Inválido**
Descrição do Teste	Teste de integração realizado para averiguar o comportamento do controlador *LoginController* ao receber credenciais inválidas de um usuário. O teste foi desenvolvido em linguagem de programação C# com o auxílio do Framework Xunit e da biblioteca Moq.
Pré-Condições	N/A
Pós-Condições	O usuário não pode ser logado
Notas:	
Resultados (Passou/Falhou/Incompleto)	Passou

	Passo do Teste	**Resultados Esperados do Teste**	**P**	**F**
1.	Simular repositório de usuários através da biblioteca Moq	Nenhum comportamento esperado	x	
2.	Configurar método *Login* para retornar *null* quando chamado por quaisquer parâmetros.	Nenhum comportamento esperado	x	
3.	Criar objeto *LoginViewModel* com credenciais inválidas	Nenhum comportamento esperado	x	
4.	Chamar o método *Login* do controlador	O resultado deve ser do tipo *UnauthorizedObjectResult*	x	

Caso de Teste	**Teste Retornar Token**
Descrição do Teste	Teste de integração realizado para averiguar se o controlador *LoginController* retorna um token JWT válido quando as credenciais de um usuário são válidas. O teste foi desenvolvido em linguagem de programação C# com o auxílio do Framework Xunit e da biblioteca Moq.
Pré-Condições	N/A
Pós-Condições	O token JWT deve ser válido
Notas:	
Resultados (Passou/Falhou/Incompleto)	Passou

	Passo do Teste	**Resultados Esperados do Teste**	**P**	**F**
1.	Simular repositório de usuários através da biblioteca Moq	Nenhum comportamento esperado	x	
2.	Configurar o método *Login* para retornar um objeto *Usuario* quando chamado com quaisquer parâmetros	Nenhum comportamento esperado	x	
3.	Criar Objeto *LoginViewModel* com credenciais válidas	Nenhum comportamento esperado	x	
4.	Chamar método *Login* do controlador	Método deve retornar um objeto *OkObjectResult*	x	
5.	Extrair token JWT do objeto *OkObjectResult*	Nenhum comportamento esperado	x	
6.	Decodificar token JWT	Verificar se o emissor *Issuer* do token é v	x	