



GOVERNO DO ESTADO DO AMAZONAS
UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA

TÓPICOS ESPECIAIS
TRABALHO PRÁTICO I

MAX BARROS DE SALES

MANAUS-AM

Para este trabalho foi utilizada uma função que calcula o Índice de massa corporal(IMC), tendo como entrada o peso(em kg) e a altura(em metros).

As possíveis entradas e saídas, bem como casos de testes estão especificadas na tabela abaixo:

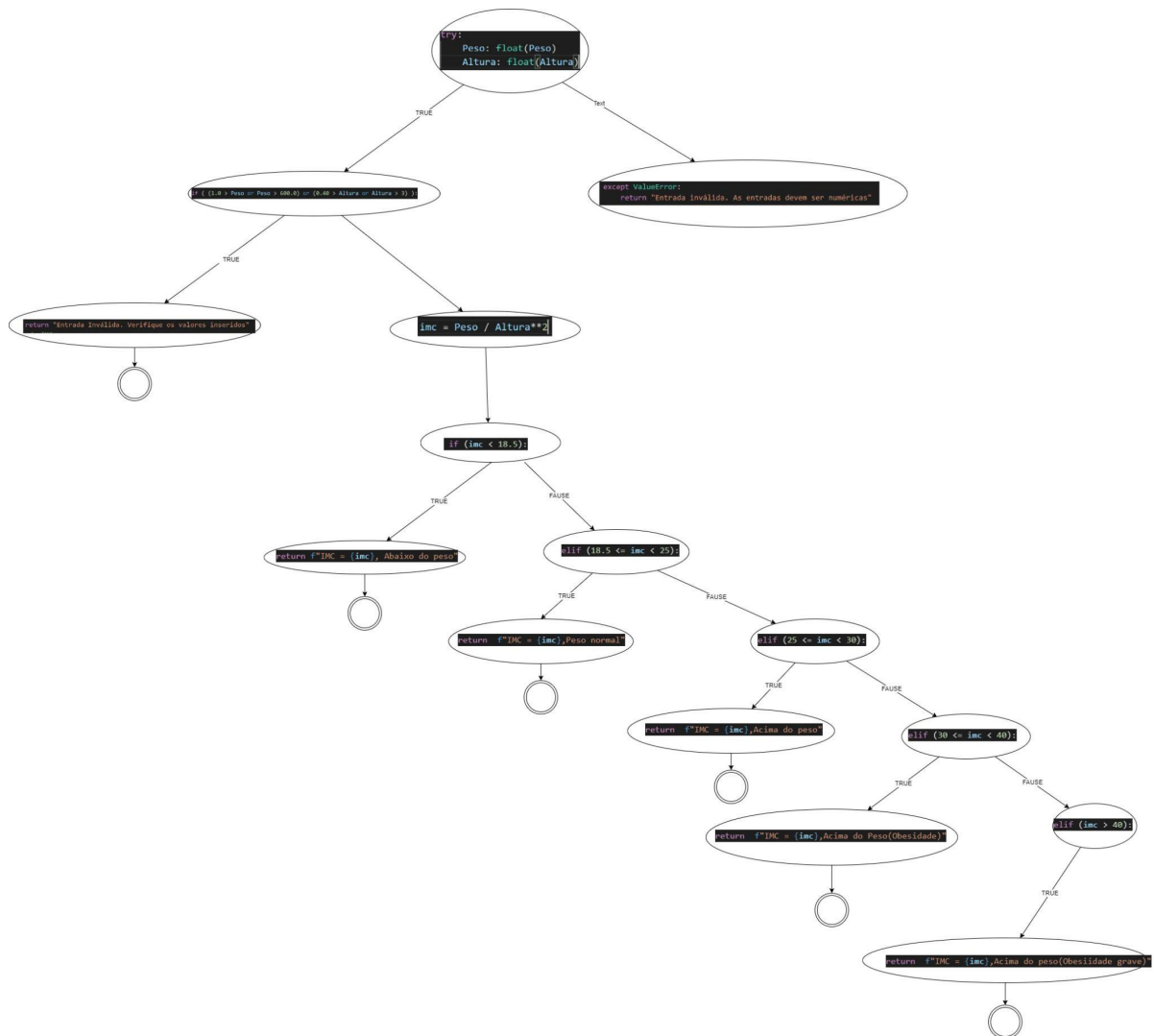
	A	B	C	D	E	F	G	H
1								
2	Entrada	peso(em kg) e altura (em metros)						
3	Saída	Abaixo do peso, peso Normal, acima do peso, muito acima do peso(obesidade), muito acima do peso(obesidade grave)						
4								
5	Entradas	Válidas	Inválidas					
6	peso	$\geq 1 \leq 600$	$<1, >600$					
7	altura	$\geq 0.40 \leq 3$	$<0.40, > 3$					
8								
9	Caso de Teste	peso	altura	Saída esperada				
10	CT01	50	1.82	Abaixo do peso				
11	CT02	50	1.60	Peso Normal				
12	CT03	60	1.43	Acima do peso				
13	CT04	100	1.81	obesidade				
14	CT05	110	1.64	obesidade grave				
15	CT06	600,1	1.72	Entrada inválida				
16	CT07	0.5	1.80	Entrada inválida				
17	CT08	50	0.2	Entrada inválida				
18	CT09	68	1.69	Peso Normal				
19	CT10	78	1.77	Peso Normal				
20	CT11	55	1.80	Abaixo do peso				
21	CT12	90	3.1	Entrada Inválida				
22	CT13	1	0.40	Abaixo do peso				
23	CT14	600	3	Obesidade grave				
24	CT15	adffd	abcd	Entrada Inválida				

Nessa tabela é possível observar que há testes para cada uma das saídas possíveis, para valores de entrada válidos e inválidos. Também há casos de testes para valores limites.

Código da função:

```
C: > Users > MaxBa > Videos > AED2 > Trab_Prattl > myIMC.py > ...  
1  #PROGRAMA PARA MEDIR O ÍNDICE DE MASSA CORPORAL(IMC)  
2  peso = input("Insira o Peso: ")  
3  altura = input("Insira a Altura: ")  
4  def imc(Peso, Altura):  
5      try:  
6          Peso = float(Peso)  
7          Altura = float(Altura)  
8      except ValueError:  
9          return "Entrada inválida. As entradas devem ser numéricas"  
10  
11     if ( (1.0 > Peso or Peso > 600.0) or (0.40 > Altura or Altura > 3) ):  
12         return "Entrada Inválida. Verifique os valores inseridos"  
13     #Calculo do IMC  
14     imc = Peso / Altura**2  
15  
16     if (imc < 18.5):  
17         return f"IMC = {imc}, Abaixo do peso"  
18     elif (18.5 <= imc < 25):  
19         return f"IMC = {imc}, Peso normal"  
20     elif (25 <= imc < 30):  
21         return f"IMC = {imc}, Acima do peso"  
22     elif (30 <= imc < 40):  
23         return f"IMC = {imc}, Acima do Peso (Obesidade)"  
24     elif (imc > 40):  
25         return f"IMC = {imc}, Acima do peso (Obesidade grave)"  
26  
27     print(imc(peso, altura))
```

Usando o critério de fluxo de controle, da técnica estrutural, foi possível obter o seguinte grafo:



Obs: O grafo pode ser visualizado com mais clareza no arquivo original do link <https://drive.google.com/file/d/1i5FkA7e8WSSf1Qnlv9ewbgLRwfg4LP9m/view?usp=sharing>

O arquivo escolhido para teste foi “myIMC.py” e o arquivo com a implementação dos testes é “teste_IMC.py”, que ficou com o seguinte código:

```
src > teste_IMC.py > RoteiroTesteFuncional > test_ct04
1  import unittest
2  import myIMC
3
4  class RoteiroTesteFuncional(unittest.TestCase):
5
6      def test_ct01(self):
7          res = myIMC.imc(50, 1.82)
8          self.assertEqual(res, "Abaixo do peso")
9
10     def test_ct02(self):
11         res = myIMC.imc(50, 1.60)
12         self.assertEqual(res, "Peso normal")
13
14     def test_ct03(self):
15         res = myIMC.imc(60, 1.43)
16         self.assertEqual(res, "Acima do peso")
17
18     def test_ct04(self):
19         res = myIMC.imc(100, 1.81)
20         self.assertEqual(res, "Acima do Peso(Obesidade)")
21
22     def test_ct05(self):
23         res = myIMC.imc(110, 1.64)
24         self.assertEqual(res, "Acima do peso(Obesiidade grave)")
25
26     def test_ct06(self):
27         res = myIMC.imc(600.1, 1.72)
28         self.assertEqual(res, "Entrada Inválida. Verifique os valores inseridos")
29
30     def test_ct07(self):
31         res = myIMC.imc(600.1, 1.72)
32         self.assertEqual(res, "Entrada Inválida. Verifique os valores inseridos")
```

Restante do código:

```
src > teste_IMC.py > RoteiroTesteFuncional > test_ct04
32     self.assertEqual(res, "Entrada Inválida. Verifique os valores inseridos")
33
34     def test_ct08(self):
35         res = myIMC.imc(0.5, 1.80)
36         self.assertEqual(res, "Entrada Inválida. Verifique os valores inseridos")
37
38     def test_ct09(self):
39         res = myIMC.imc(68, 1.69)
40         self.assertEqual(res, "Peso normal")
41
42     def test_ct010(self):
43         res = myIMC.imc(78, 1.77)
44         self.assertEqual(res, "Peso normal")
45
46     def test_ct011(self):
47         res = myIMC.imc(55, 1.80)
48         self.assertEqual(res, "Abaixo do peso")
49
50     def test_ct012(self):
51         res = myIMC.imc(90, 3.1)
52         self.assertEqual(res, "Entrada Inválida. Verifique os valores inseridos")
53
54     def test_ct013(self):
55         res = myIMC.imc(1, 0.40)
56         self.assertEqual(res, "Abaixo do peso")
57
58     def test_ct014(self):
59         res = myIMC.imc(600, 3)
60         self.assertEqual(res, "Acima do peso(Obesiidade grave)")
61     def test_ct15(self):
62         res = myIMC.imc("adfffd", "abcd")
63         self.assertEqual(res, "Entrada inválida. As entradas devem ser numéricas")
```



GOVERNO DO ESTADO DO AMAZONAS
UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA

Quanto à execução, Todos os testes foram realizados com sucesso e com a cobertura de 100% da função testada, assim como mostrado abaixo:

```
===== 15 passed in 0.09s =====

MaxBa@DESKTOP-DQP5ET5 MINGW64 ~/Videos/Topicos_especiais/AP1/Topicos_Especial_AP1/src (main)
$ pytest teste_IMC.py
===== test session starts =====
platform win32 -- Python 3.10.2, pytest-7.1.1, pluggy-1.0.0
rootdir: C:\Users\MaxBa\Videos\Topicos_especiais\AP1\Topicos_Especial_AP1\src
plugins: cov-3.0.0
collected 15 items

teste_IMC.py ..... [100%]

===== 15 passed in 0.05s =====

MaxBa@DESKTOP-DQP5ET5 MINGW64 ~/Videos/Topicos_especiais/AP1/Topicos_Especial_AP1/src (main)
$
```

teste de cobertura:

```
MaxBa@DESKTOP-DQP5ET5 MINGW64 ~/Videos/Topicos_especiais/AP1/Topicos_Especial_AP1
$ pytest -vv --cov=. --cov-report=term-missing teste_IMC.py
===== test session starts =====
platform win32 -- Python 3.10.2, pytest-7.1.1, pluggy-1.0.0 -- C:\Users\MaxBa\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.10.2.0_qbz9nmbk2glc%
cachedir: .pytest_cache
rootdir: C:\Users\MaxBa\Videos\Topicos_especiais\AP1\Topicos_Especial_AP1\src
plugins: cov-3.0.0
collected 15 items

teste_IMC.py::RoteiroTesteFuncional::test_ct01 PASSED
teste_IMC.py::RoteiroTesteFuncional::test_ct010 PASSED
teste_IMC.py::RoteiroTesteFuncional::test_ct011 PASSED
teste_IMC.py::RoteiroTesteFuncional::test_ct012 PASSED
teste_IMC.py::RoteiroTesteFuncional::test_ct013 PASSED
teste_IMC.py::RoteiroTesteFuncional::test_ct014 PASSED
teste_IMC.py::RoteiroTesteFuncional::test_ct02 PASSED
teste_IMC.py::RoteiroTesteFuncional::test_ct03 PASSED
teste_IMC.py::RoteiroTesteFuncional::test_ct04 PASSED
teste_IMC.py::RoteiroTesteFuncional::test_ct05 PASSED
teste_IMC.py::RoteiroTesteFuncional::test_ct06 PASSED
teste_IMC.py::RoteiroTesteFuncional::test_ct07 PASSED
teste_IMC.py::RoteiroTesteFuncional::test_ct08 PASSED
teste_IMC.py::RoteiroTesteFuncional::test_ct09 PASSED
teste_IMC.py::RoteiroTesteFuncional::test_ct15 PASSED

----- coverage: platform win32, python 3.10.2-final-0 -----
Name                Stmts   Miss  Cover   Missing
-----
myIMC.py              19      0   100%
teste_IMC.py         48      0   100%
-----
TOTAL                 67      0   100%
```