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

**Disciplina: Tópicos Especiais de Eng. Software**
**Aluno: Natanael da Mota Figueira**

**Programa**

O programa foi desenvolvido com o intuito de calcular a média das notas de AP1 e AP2 de um aluno da UEA em uma matéria, para verificar se com as notas ele passa direto, se precisa fazer uma PF para passar ou se reprova sem a possibilidade de fazer uma PF. Abaixo a função principal que vai ser usada para os casos de testes.

```python
def calculaMedia(ap1, ap2):
    #Tratando casos de entradas inválidas
    try:
        ap1 = float(ap1)
        ap2 = float(ap2)
    except ValueError:
        return("Entrada inválida, a "+
            "entrada não é um número.")

    if(ap1 > 10):
        return("Entrada inválida, a "+
                "AP1 tem valor maior que 10.")
    elif(ap1 < 0):
        return("Entrada inválida, a "+
                "AP1 tem valor negativo.")
    elif(ap2 > 10):
        return("Entrada inválida, a "+
                "AP2 tem valor maior que 10.")
    elif(ap2 < 0):
        return("Entrada inválida, a "+
                "AP2 tem valor negativo.")

    #fazendo cálculo da média
    media = (ap1 + ap2) / 2
    if (media >= 8):
        return("Aluno aprovado.")
    elif (media < 4):
        return("Aluno reprovado.")
    else:
        return("Aluno precisa fazer PF.")
```

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

**Roteiro de Testes - Técnica Funcional**

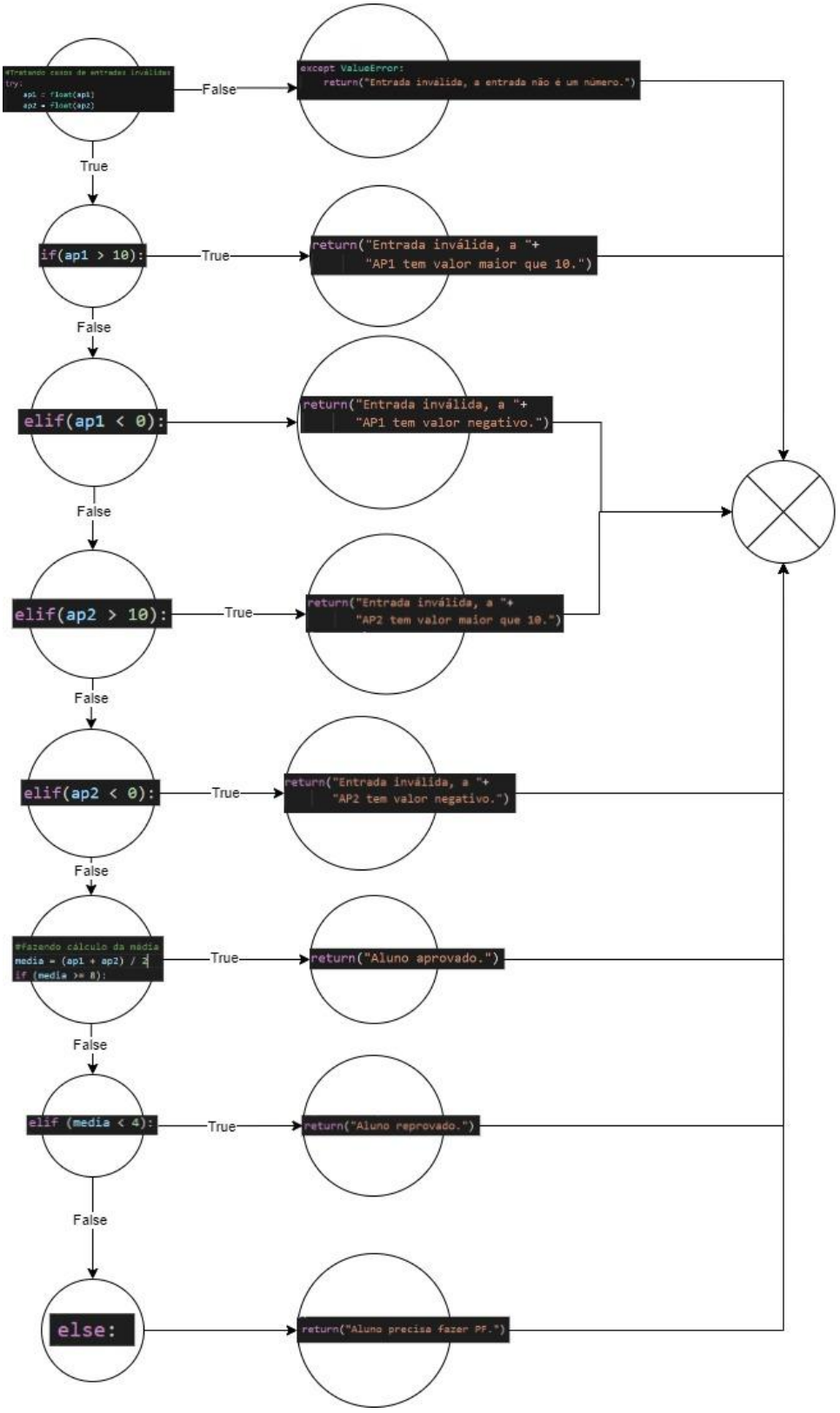
Para os testes foi considerado que:

- AP1 e AP2 devem ser números float;
- AP1 e AP2 devem ser de 0 até 10;
- Para média maior ou igual a 8, será retornado “Aluno aprovado”;
- Para média menor que 8 e maior ou igual a 4, será retornado “Aluno precisará fazer PF”;
- Para média menor que 4, será retornado “Aluno reprovado”.

| **Roteiro: R1** | Verificar aprovação de aluno com base nas notas de AP1 e AP2 | | |
|---|---|---|---|
| | **Entradas** | | **Saídas** |
| **ID_CT** | **Ap1** | **Ap2** | |
| CT01 | -0.01 | 5 | Entrada inválida, a AP1 tem valor negativo. |
| CT02 | 10.01 | 5 | Entrada inválida, a AP1 tem valor maior que 10. |
| CT03 | 5 | -0.01 | Entrada inválida, a AP2 tem valor negativo. |
| CT04 | 5 | 10.01 | Entrada inválida, a AP2 tem valor maior que 10. |
| CT05 | 5 | a | Entrada inválida, a entrada não é um número. |
| CT06 | a | 5 | Entrada inválida, a entrada não é um número. |
| CT07 | 3.9 | 3.9 | Aluno reprovado. |
| CT08 | 0.01 | 0.01 | Aluno reprovado. |
| CT09 | 7.9 | 7.9 | Aluno precisa fazer PF. |
| CT10 | 0 | 0 | Aluno reprovado. |
| CT11 | 10 | 10 | Aluno aprovado. |
| CT12 | 4 | 4 | Aluno precisa fazer PF. |
| CT13 | 8 | 8 | Aluno aprovado. |
| CT14 | 8.01 | 8.01 | Aluno aprovado. |
| CT15 | 4.01 | 4.01 | Aluno precisa fazer PF. |

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

# Roteiro de Testes - Técnica Estrutural



## Implementação dos Casos de Teste

### Código do arquivo responsável pelo teste dos casos de teste:

```
import unittest
import calculaMedia

class testCalculaMedia (unittest.TestCase):
    def test_ct01(self):
        ap1 = -0.01
        ap2 = 5
        self.assertEqual(calculaMedia.calculaMedia(ap1, ap2), "Entrada
inválida, a AP1 tem valor negativo.")

    def test_ct02(self):
        ap1 = 10.01
        ap2 = 5
        self.assertEqual(calculaMedia.calculaMedia(ap1, ap2), "Entrada
inválida, a AP1 tem valor maior que 10.")

    def test_ct03(self):
        ap1 = 5
        ap2 = -0.01
        self.assertEqual(calculaMedia.calculaMedia(ap1, ap2), "Entrada
inválida, a AP2 tem valor negativo.")

    def test_ct04(self):
        ap1 = 5
        ap2 = 10.01
        self.assertEqual(calculaMedia.calculaMedia(ap1, ap2), "Entrada
inválida, a AP2 tem valor maior que 10.")

    def test_ct05(self):
        ap1 = 5
        ap2 = "a"
        self.assertEqual(calculaMedia.calculaMedia(ap1, ap2), "Entrada
inválida, a entrada não é um número.")

    def test_ct06(self):
```

```
        ap1 = "a"
        ap2 = 5
        self.assertEqual(calculaMedia.calculaMedia(ap1, ap2), "Entrada
inválida, a entrada não é um número.")

    def test_ct07(self):
        ap1 = 3.9
        ap2 = 3.9
        self.assertEqual(calculaMedia.calculaMedia(ap1, ap2), "Aluno
reprovado.")

    def test_ct08(self):
        ap1 = 0.01
        ap2 = 0.01
        self.assertEqual(calculaMedia.calculaMedia(ap1, ap2), "Aluno
reprovado.")

    def test_ct09(self):
        ap1 = 7.9
        ap2 = 7.9
        self.assertEqual(calculaMedia.calculaMedia(ap1, ap2), "Aluno
precisa fazer PF.")

    def test_ct10(self):
        ap1 = 0
        ap2 = 0
        self.assertEqual(calculaMedia.calculaMedia(ap1, ap2), "Aluno
reprovado.")

    def test_ct11(self):
        ap1 = 10
        ap2 = 10
        self.assertEqual(calculaMedia.calculaMedia(ap1, ap2), "Aluno
aprovado.")

    def test_ct12(self):
        ap1 = 4
        ap2 = 4
```

```
        self.assertEqual(calculaMedia.calculaMedia(ap1, ap2), "Aluno
precisa fazer PF.")

    def test_ct13(self):
        ap1 = 8
        ap2 = 8
        self.assertEqual(calculaMedia.calculaMedia(ap1, ap2), "Aluno
aprovado.")

    def test_ct14(self):
        ap1 = 8.01
        ap2 = 8.01
        self.assertEqual(calculaMedia.calculaMedia(ap1, ap2), "Aluno
aprovado.")

    def test_ct15(self):
        ap1 = 4.01
        ap2 = 4.01
        self.assertEqual(calculaMedia.calculaMedia(ap1, ap2), "Aluno
precisa fazer PF.")

if __name__ == '__main__':
    unittest.main()
```

**Resultado do Teste:**

Todos os casos de testes passaram com sucesso, com cobertura de 100%.

```
natanaelmota1@DESKTOP-GD2CTNJ:/mnt/c/Users/natan/Downloads/Backup-codigos-de-estudo-master/Codigos_Python/Uri$ pytest testCalculaMedia.py
============================= test session starts ==============================
platform linux -- Python 3.8.10, pytest-7.1.1, pluggy-1.0.0
rootdir: /mnt/c/Users/natan/Downloads/Backup-codigos-de-estudo-master/Codigos_Python/Uri
plugins: cov-3.0.0
collected 15 items

testCalculaMedia.py ...............                                      [100%]

============================== 15 passed in 0.19s ==============================
```