



N704-Programação funcional - Grupo 4

AKASSIO CORREIA DA SILVA, FRANCISCO – 2222819
MENDONÇA COELHO GALVÃO, MARIANA – 2222923
VICTOR DUARTE FERNANDES DA SILVA, JOÃO – 2222816
LUCAS PINTO MARTINS, ANTÔNIO – 2222842
BARBOZA, LUNNA – 2222791
SANTIAGO TORRES BANDEIRA, DIEGO – 2222779

2024



1. INTRODUÇÃO

- a. **Objetivos – Praticar e testar o conhecimento e programação funcional com o desenvolvimento de uma aplicação.**

2. REQUISITOS

Requisitos Funcionais:

Cálculo de Imposto: O sistema deve calcular o imposto sobre produtos com base em um valor mínimo e uma taxa de imposto predefinida.

Aplicação de Desconto: O sistema deve permitir a aplicação de um desconto sobre o valor total do imposto.

Interface com o Usuário: O sistema deve solicitar ao usuário a quantidade de produtos e o valor de cada um.

Saída: O sistema deve apresentar o valor total do imposto após a aplicação do desconto.

Requisitos Não Funcionais:

Desempenho: O sistema deve em menos de 1 segundo ao usuário responder rapidamente ao usuário.

Flexibilidade: O sistema deve permitir a alteração da taxa de imposto e do valor mínimo sem grandes modificações no código.

3. PAPEIS

- ☐ **Mariana Mendonça, Diego Santiago e Lunna Barboza – Requisitos;**
- ☐ **Antônio Lucas, Akassio Correia, João Victor Duarte – Implementação do código;**
- ☐ **Antônio Lucas, Akassio Correia, João Victor Duarte – Testes;**
- ☐ **Mariana Mendonça, Diego Santiago e Lunna Barboza - Elaboração do relatório;**

4. TESTES

```
#testando aplicações
class TestCalculoImposto(unittest.TestCase):

    def test_valor_maior_1000_com_desconto(self):
        desconto_final = aplicar_desconto(0.1)
        imposto_com_desconto = desconto_final(calcular_imposto(1500))
        self.assertEqual(imposto_com_desconto, 225)
        # Para um produto de 1500, o imposto seria 250 e com 10% de desconto, o imposto é 225

    def test_valor_menor_1000(self):
        self.assertEqual(calcular_imposto(800), 0)
        # Produto com valor menor que 1000 não paga imposto

    def test_calculo_total_impostos(self):
        valores = [800, 1500, 2000]
        total_impostos = calcular_impostos_total(valores) # Aplicando 10% de desconto
        desconto_final2 = aplicar_desconto(0.1)
        total_com_desconto2 = desconto_final2(total_impostos)
        self.assertEqual(total_com_desconto2, 675)
        # Impostos:
        # Produto 1 (800): 0
        # Produto 2 (1500): 225 (desconto aplicado)
        # Produto 3 (2000): 450 (desconto aplicado)
        # Total = 0 + 225 + 450 = 675
```

```
Quantos produtos você deseja calcular? 1
Digite o valor do produto: 1500
O valor total do imposto a ser pago, após desconto, é de R$ 225.00
...
-----
Ran 3 tests in 0.000s
```

```
AKIO/OneDrive/Camily-area de trabalho/Área de Trabalho/Nova pasta/imposto.py
Quantos produtos você deseja calcular? 1
Digite o valor do produto: 800
O valor total do imposto a ser pago, após desconto, é de R$ 0.00
...
-----
Ran 3 tests in 0.000s

OK
PS C:\Users\USUARIO\OneDrive\Camily-area de trabalho\Área de Trabalho\Nova pasta>
```

```
Quantos produtos você deseja calcular? 1
Digite o valor do produto: 1500
O valor total do imposto a ser pago, após desconto, é de R$ 225.00
...
-----
Ran 3 tests in 0.000s

OK
```

```
AKIO/OneDrive/Camily-area de trabalho/Área de Trabalho/Nova pasta/imposto.py
Quantos produtos você deseja calcular? 3
Digite o valor do produto: 800
Digite o valor do produto: 1500
Digite o valor do produto: 2000
O valor total do imposto a ser pago, após desconto, é de R$ 675.00
...
-----
Ran 3 tests in 0.000s

OK
PS C:\Users\USUARIO\OneDrive\Camily-area de trabalho\Área de Trabalho\Nova pasta>
```

