

# Teste de Cobertura

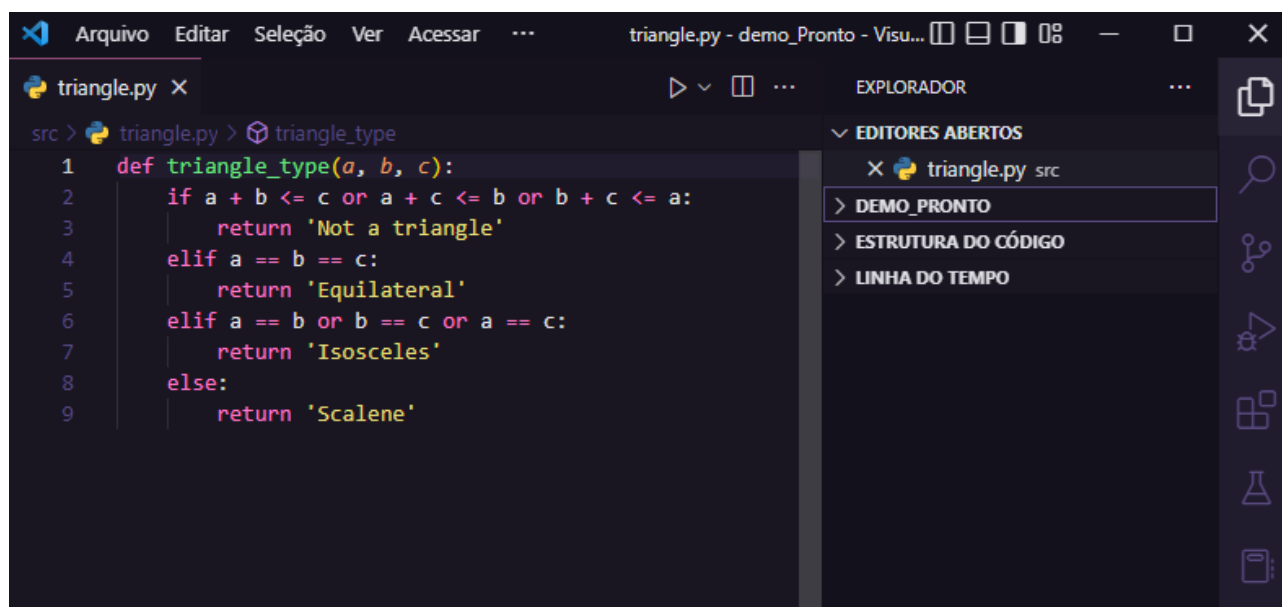
Centro de Informática  
Universidade Federal de Pernambuco

Residente: Rosane Caldeira

07/10/2022

Escreva uma suite de testes que alcance 100% de cobertura de instruções (statements) e 100% de cobertura de decisões (branches) para o código em anexo (triangle.py).

Utilize o plugin pytest-cov para verificar a cobertura da sua suíte de testes.

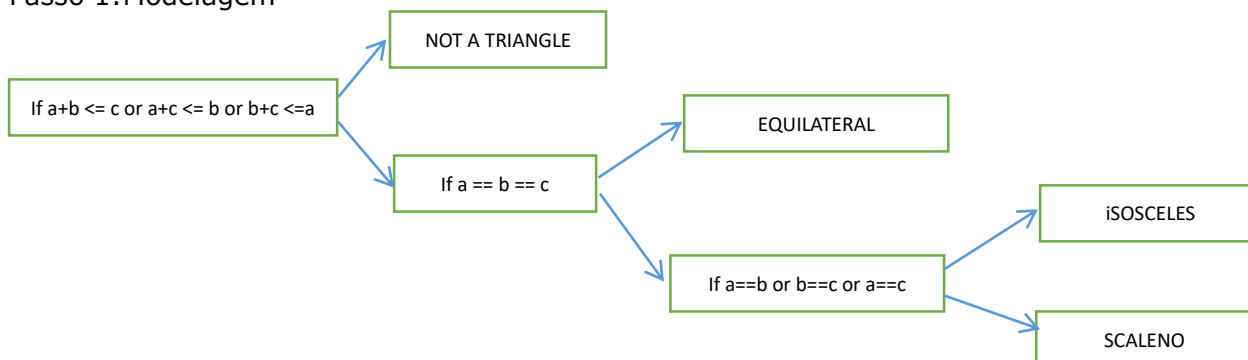


```
1 def triangle_type(a, b, c):
2     if a + b <= c or a + c <= b or b + c <= a:
3         return 'Not a triangle'
4     elif a == b == c:
5         return 'Equilateral'
6     elif a == b or b == c or a == c:
7         return 'Isosceles'
8     else:
9         return 'Scalene'
```

Visualização vscode

Passos para a confecção do trabalho\_

Passo 1: Modelagem



Lembrando que para cobrir cada decisão deve-ser verificar com TRUE or FALSE.

Passo 2: Construção da tabela de casos de testes possíveis para o código de um triangle:

entradas											
a	b	c	$a+b \leq c$	$a+c \leq b$	$b+c \leq a$		$a == b == c$		$a==b$	$b==c$	$a==c$
3	2	1	F	F	V		F		F	F	F
3	3	3	F	F	F		V		V	V	V
3	3	4	F	F	F		F		V	F	F
3	4	5	F	F	F		F		F	F	F
1	2	3	V	V	F		F		F	F	F
1	1	1	F	F	F		V		V	V	V
1	1	2	V	V	F		F		V	F	F

Do conjunto de testes possíveis foram escolhidas as combinações: {3,2,1}, {3,3,3}, {3,3,4}, {3,4,5} por retornarem 100% de cobertura.

Passo 3: pytest -cov retorna os percentuais cobertos pelos casos de teste que atende ao código triangle em 100% cobertura.

Passo 4: para maiores detalhes acesso via github -> <https://github.com/rcaldeira22/SuiteTestes>

“A imaginação é mais importante que o conhecimento.  
O conhecimento é limitado, enquanto a imaginação  
abraça o mundo inteiro, estimulando o progresso,  
e dando origem à evolução.”  
Albert Einstein