



Disciplina	Prof. João Choma	
PROJETO IMPLEMENTAÇÃO E TESTE DE SOFTWARE	Valor	+01 ATV
ATIVIDADE : TESTE ESTRUTURAL	Aluno: Pedro Emanuel	Aluno:
ESOFT - 6 - N A	Aluno:	Aluno: Roberto Yanez

Atividade prática de teste Estrutural Passos:

1. Projetar **casos de teste Estruturais** para avaliar os quatro algoritmos dos itens listados abaixo. Conforme o exemplo abaixo, e o excerto do Livro Didático. 2. Preencher os ARTEFATOS de teste abaixo para os testes projetados.

b. Um algoritmo que lê dois números e calcula o máximo divisor comum pelo método de Euclides:

```
a = int(input('Digite o primeiro número: '))
b = int(input('Digite o segundo número: '))
while b != 0:
    r = a % b
    a = b
    b = r
print(f'O MDC é {a}')
```

Passo 1: Desenhe o grafo de fluxo correspondente

Passo 2: $V(G) = 3$

Passo 3: Determine um conjunto base de caminhos independentes.

Início

1. Início
2. Leitura do primeiro número
3. Leitura do segundo número
4. Verificação de erro (entrada inválida)

5. Teste $b \neq 0$
6. Cálculo resto
7. Troca valores
8. Fim do laço
9. Impressão do resultado
10. Mensagem de erro
11. Fim

Caminho 1: 1-2-3-5-6-7-5...-8-9-10

Passo 4: Prepare os casos de teste:
PLANOS DE TESTE A SER DESCRITO :

ITENS A TESTAR / ABORDAGEM:

N.	Item	Especificação ABORDAGEM:
1	Leitura do número	Programa deve aceitar dois inteiros, positivos ou negativos ----- Deve calcular corretamente o MDC usando algoritmo de Euclides ----- Se entrada não-numérica ou ambos zero, informar erro ou encerrar
2	Cálculo do MDC	
3	Tratamento de exceções	

CRONOGRAMA DE TESTES

ID	Tarefa	Início	Fim	Esforço Pré Pessoa	Obs
01	Testar entrada válida	21/09/25	21/09/25	5min	Positivo e negativo
02	Testar cálculo do MDC	21/09/25	21/09/25	2min	Conferir resultados
03	Testar saída	21/09/25	21/09/25	2min	Verifique ordem e ausência de duplicatas
04	Testar zero e exceção	21/09/25	21/09/25	1min	b = 0, ambos zeros ou texto

AMBIENTE DE TESTE

Ambiente	Descrição
Hardware	Computador padrão, sem requisitos especiais
Software	Python 3.x instalado
Ferramental	VSCode

IDENTIFICAÇÃO DE CASO DE TESTE / IDENTIFICAÇÃO DE PROCEDIMENTO DE TESTE

Nº	Caso de Teste	Identificação do Caso de Teste	Procedimento	Identificação do Procedimento de Teste
1	MDC de (12, 8)	MDC simples	Digitar 12, depois 8	MDCT-01
2	MDC de (15, 0)	Um zero	Digitar 15, depois 0	MDCT-02
3	MDC de (0, 0)	Ambos zeros	Digitar 0, depois 0	MDCT-03
4	MDC de (-36, 12)	Negativo	Digitar -36, depois 12	MDCT-04
5	MDC de texto	Inválido	Digitar 'a', depois 5	MDCT-05

CASO DE TESTE

Identificação	MDCT-01, 02, 03, 04, 05	
Itens a Testar	Entrada, cálculo, saída	
Entradas	Campo	Valor
	a, b	12, 8
	a, b	15, 0
	a, b	0, 0
	a, b	-36, 12
	a, b	a, 5
Saídas Esperadas	Campo	Valor
	MDC -----	MDC é 4 -----

	MDC	O MDC é 15
	MDC	O MDC é 0
	MDC ----- MDC	O MDC é 12 ----- Traceback (most recent call last): a = int(input('Digite o primeiro número: ')) ~~~~~ ValueError: invalid literal for int() with base 10: 'a'
Ambiente	VSCode	
Procedimento	Executar programa com valores 12,8 ; 15,0 ; 0,0 ; -36,12 ; a,5 e conferir a listagem dos divisores conforme esperado.	
Dependência	Código funcionando, ambiente Python pronto.	

PROCEDIMENTO DE TESTE

Identificação	MDCT -01, 02, 03, 04
Objetivo	Verificar o cálculo e print do MDC correto.
Requisitos	Python instalado para rodar arquivos .py
Fluxo	<ol style="list-style-type: none"> 1. Abrir/rodar programa em Python 2. Digitar 12, 8 3. Verificar saída: 4 ----- <ol style="list-style-type: none"> 1. Abrir/rodar programa em Python 2. Digitar 15, 0 3. Verificar saída: 15 ----- <ol style="list-style-type: none"> 4. Abrir/rodar programa em Python 5. Digitar 0, 0 6. Verificar saída: 0 -----

7. Abrir/rodar programa em Python

8. Digitar -36, 12

9. Verificar saída:12

10. Abrir/rodar programa em Python

11. Digitar a, 5

12. Verificar saída:

Traceback (most recent call last):

a = int(input('Digite o primeiro número: '))

~~~~~

ValueError: invalid literal for int() with base 10: 'a'