**Disciplina**: Engenharia de Software

**Curso**: Bacharelado em Sistemas de Informação e Bacharelado em Ciência da Computação

**Aplicar técnicas para verificação e validação de produtos de software**

**Descrição da atividade**

Esta atividade trata do **planejamento** e da **execução de testes** de um programa que deve funcionar da seguinte maneira:

* Recebe 3 valores **numéricos** em 3 variáveis: A, B e C
* Troca os valores entre as variáveis de forma que os valores entre as variáveis sejam ordenados de forma ascendente (menor para o maior). Não utilizar nenhuma função de ordenação da linguagem selecionada (**implementação raiz**).
* Imprime os valores de A, B e C

**Exemplo**

**Entrada:**

**A = 2**

**B = 5**

**C = 1**

**Saída:**

**1 2 5**

1. Descreva todos os **Casos de Teste** necessários para garantir que todas as saídas correspondam aos valores esperados.

|  |  |  |
| --- | --- | --- |
| ID caso de teste | Entrada | Saída esperada |
| *1* | *A = 1; B = 2; C = 3* | *1 2 3* |
| *2* | *A = 3; B = 2; C = 1* | *1 2 3* |
| *3* | *A = -1; B = -2; C = -3* | *-1 -2 -3* |
| *4* | *A = 1; B = -1; C = -11* | *-1 1 11* |
| *5* | *A = 456; B = 123; C = -90* | *-90 123 456* |

1. Considerando os casos de testes identificados, procure implementar (abordagem TDD) a automação dos testes (em qualquer linguagem).

import unittest

def ordena\_ascending(a, b, c):

if a > b:

a, b = b, a

if b > c:

b, c = c, b

if a > b:

a, b = b, a

return a, b, c

class TestOrdenaAscending(unittest.TestCase):

def test\_valores\_em\_ordem\_crescente(self):

resultado = ordena\_ascending(-1, -2, -3)

self.assertEqual(resultado, (-3, -2, -1))

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

1. Implemente um programa (em qualquer linguagem) que realize as funções descritas no enunciado da atividade.

def ordena\_ascending(a, b, c):

    if a > b:

        a, b = b, a

    if b > c:

        b, c = c, b

    if a > b:

        a, b = b, a

    return a, b, c

# Leitura dos valores de entrada

a = int(input("Digite o valor de A: "))

b = int(input("Digite o valor de B: "))

c = int(input("Digite o valor de C: "))

# Chama a função para ordenar os valores

a, b, c = ordena\_ascending(a, b, c)

# Exibe os valores ordenados

print(a, b, c)

1. Execute os **Casos de Teste** planejados no Passo 1, utilizando o programa desenvolvido no Passo 2 e registre se os casos foram executados com SUCESSO, ERRO ou FALHA. Caso encontre ERRO ou FALHA identifique o DEFEITO.

|  |  |  |
| --- | --- | --- |
| ID caso de teste | Sucesso/Erro/Falha | Motivo do Defeito |
| *1* | *SUCESSO* | *-* |
| *2* | *ERRO* | *Saída esperada 1 2 3, mas retornou 1 3 2* |
| *3* | *SUCESSO* | *Saída esperada -3 -2 -1 e retornou -3 -2 -1* |
| *4* | *SUCESSO* | *Saída esperada -1 1 11 e retornou -1 1 11* |
| *5* | *SUCESSO* | *Saída esperada -90 123 456 e retornou -90 123 456* |

Obs: o objetivo da atividade não é criar o programa perfeito, mas entender o processo de planejamento dos testes (Passo 1) e posteriormente o registro dos resultados dos testes e os defeitos identificados (Passo 4). Em vermelho exemplos de casos de testes.