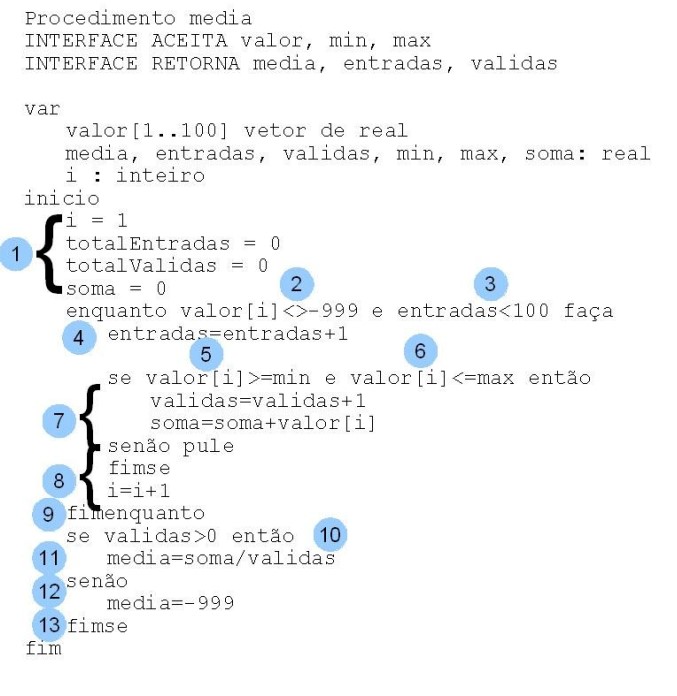
|  |  |  |
| --- | --- | --- |
| **Disciplina** | **Prof. Dacio Machado** |  |
| PROJETO IMPLEMENTAÇÃO E TESTE DE  SOFTWARE | Valor | +01 ATV |
| **ATIVIDADE : TESTE ESTRUTURAL** | Aluno: Antonio Ferreira de lima | Aluno: Karen Tanaka |
| **ESOFT - 6 - N** | Aluno: Gabrielle Morente Perna | Aluno: Leonardo Leitão Souza |

**Atividade prática de teste Estrutural Passos:**

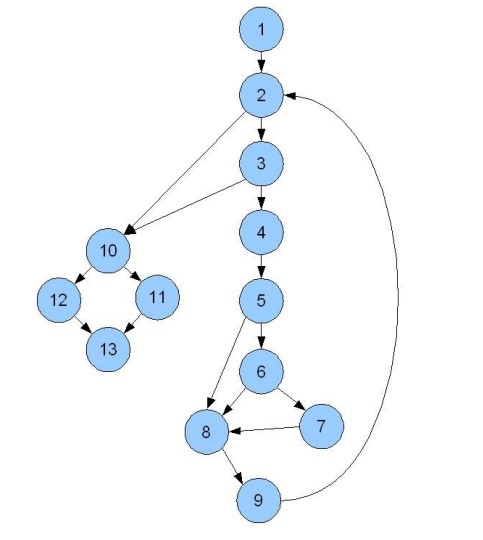
1. Projetar **casos de teste Estruturais** para avaliar os quatro algoritmos dos itens listados abaixo. Conforme o exemplo abaixo, e o excerto do Livro Didático.
2. Preencher os ARTEFATOS de teste abaixo para os testes projetados.
3. Construa, em sua linguagem de preferência os seguintes algoritmos:
   1. Um algoritmo que lê um número e imprime a lista dos seus divisores
   2. Um algoritmo que lê dois números e calcula o máximo divisor comum pelo método de Euclides.
   3. Um algoritmo que lê as 4 notas de um aluno e diga se ele passou por média, está em final ou reprovou
   4. Um algoritmo em que dado dois números *n* e *k*  (n< k ), calcule e apresente a combinatória de n elementos tomados k a k

**Exemplo de Desenvolvimento**: Derivar os casos de teste para um programa que calcula a

média das entradas válidas, usando o método do caminho básico.



**Passo 1:** Desenhe o grafo de fluxo correspondente

**Passo 2:** Calcule a complexidade ciclomática. V(G) = 6 regiões V(G) = 17 arestas – 13 nós + 2 = 6 V(G) = 5 nós predicados + 1 = 6

**Passo 3:** Determine um conjunto base de caminhos independentes.

Caminho 1: 1-2-10-11-13

Caminho 2: 1-2-10-12-13

Caminho 3: 1-2-3-10-11-13 Caminho 4: 1-2-3-4-5-8-9-2...

Caminho 5: 1-2-3-4-5-6-8-9-2...

Caminho 6: 1-2-3-4-5-6-7-8-9-2…

**Passo 4:** Prepare os casos de teste que vão forçar a execução de cada caminho: O caminho 1 só pode ser testado como parte dos caminhos 4, 5 e 6

Caminho 2: valor (i) = -999; resultados esperados: média = -999 e os outros valores com os valores iniciais.

Caminho 6: valor (i) = entrada válida; resultados esperados: média correta baseada em n valores e totais apropriados.

**PLANOS DE TESTE A SER DESCRITO :**

ITENS A TESTAR / ABORDAGEM:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N**॰ | **Item** | **Especificação** |  | **ABORDAGEM:** Teste estrutural (método do caminho básico). Objetivo: cobrir decisões e iterações do algoritmo (laço de leitura, verificação de faixa e cálculo final). Gerar casos que forcem: parada por sentinela (-999), parada por limite 100, valores válidos/ inválidos, bordas (valor == min, value == max) e caso validas = 0. Registrar entradas, validas, soma e media. |
| 1 | Entrada de valores | O algoritmo deve aceitar até 100 valores ou até encontrar o sentinela -999. |
| 2 | |  | | --- | | Verificação de limites |  |  | | --- | |  | | Apenas valores dentro do intervalo [min, max] devem ser considerados válidos. |
| 3 | |  | | --- | | Cálculo da média |  |  | | --- | |  | | e houver entradas válidas, calcular corretamente a média; caso contrário, retornar -999. |

# CRONOGRAMA DE TESTES

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Tarefa** | **Início** | **Fim** | **Esforço** | **Pré** | **Pessoa** | **Obs** |
| 01 | Preparar casos de teste (grafo + caminhos) | 11/09 | 11/09 | 2h |  | Gabrielle |  |
| 02 | Implementar algoritmo em código | 11/09 | 12/09 | 3h | 01 | Antonio |  |
| 03 | Executar testes com entradas simuladas | 12/09 | 11/09 | 2h | 02 | Karen |  |
| 04 | |  | | --- | | Documentar resultados obtidos |  |  | | --- | |  | | 12/09 | 11/09 | 1h | 03 | Leonardo |  |

# AMBIENTE DE TESTE

|  |  |
| --- | --- |
| **Ambiente** | **Descrição** |
| Hardware | Notebook com 8 GB RAM, Intel i5 |
| Software | Windows 10, VS Code |
| Ferramental | Java , editor de texto, compilador/intérprete |

## IDENTIFICAÇÃO DE CASO DE TESTE / IDENTIFICAÇÃO DE PROCEDIMENTO DE TESTE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **N**॰ | **Caso de Teste** | **Identificação do Caso de Teste** |  | **Procedimento** | **Identificação do Procedimento de Teste** |
| 1 | |  | | --- | | Nenhum valor válido |  |  | | --- | |  | | |  | | --- | | CT-01 |  |  | | --- | |  | | |  | | --- | | Inserir apenas -999 como primeira entrada |  |  | | --- | |  | | PT-01 |
| 2 | |  | | --- | | Apenas um valor válido |  |  | | --- | |  | | CT-02 | Inserir um número dentro do intervalo e depois -999 | PT-02 |
| 3 | |  | | --- | | Valores fora do intervalo |  |  | | --- | |  | | CT-03 | Inserir valores menores que min ou maiores que max, depois -999 | PT-03 |
| 4 | |  | | --- | | Vários valores válidos |  |  | | --- | |  | | CT-04 | Inserir sequência de valores dentro do intervalo e depois -999 | PT-04 |
| 5 | |  | | --- | | Limite máximo |  |  | | --- | |  | | CT-05 | Inserir 100 valores válidos (sem -999) | PT-05 |

CASO DE TESTE

|  |  |  |
| --- | --- | --- |
| **Identificação** | CT-01 |  |
| **Itens a Testar** | Entrada de valores, Cálculo da média |  |
| **Entradas** | **Campo**   |  | | --- | | **valor[1]** |  |  | | --- | |  | | **Valor**  -999 |
|  |  |
|  |  |
|  |  |
| **Saídas**  **Esperadas** | **Campo** | **Valor** |
| media | |  | | --- | | -999 |  |  | | --- | |  | |
|  |  |
| **Ambiente** | VS Code |  |
| **Procedimento** | Executar o algoritmo e inserir -999 como primeira entrada. |  |
| **Dependência** | NA |  |

PROCEDIMENTO DE TESTE

|  |  |
| --- | --- |
| **Identificação** | **PT-01** |
| **Objetivo** | Validar se o algoritmo encerra a leitura imediatamente ao receber -999 e retorna media = -999. |
| **Requisitos** | Programa implementado e executável.  Definição de min e max (exemplo: min=0, max=10). |
| **Fluxo** | Iniciar a execução do programa.  Informar a primeira entrada: -999.  Encerrar entrada (se necessário).  Observar a saída.  Conferir se a saída corresponde ao esperado: media = -999, entradas = 0, validas = 0. |